

Dual-Prompt-Driven Textual Knowledge Graph Generation With Heterogeneous Information Embedding for CTR Prediction

Xiling Feng¹, Han Liu^{1,2*}, Yinghui Pan³, Zhiwu Xu¹

¹ College of Computer Science and Software Engineering, Shenzhen University

² Guangdong Provincial Key Laboratory of Intelligent Information Processing, Shenzhen University

³ School of Artificial Intelligence, Shenzhen University
{han.liu, panyinghui, xuzhiwu}@szu.edu.cn
{2410105004}@mails.szu.edu.cn

Abstract

In click-through rate prediction tasks, learning high-quality feature embeddings from multi-field categorical data is an essential task. Early approaches typically need to transform categorical features into one-hot encoded vectors for producing embeddings. This transformation often results in loss of semantic information. In the era of large language models, it has become necessary to transform multi-field categorical data into text to achieve embedding of semantic information. Moreover, semantic information of each categorical feature can be useful to model the relationships among categories for enhancing the embedding quality. To address this point, in this paper, we propose an approach of dual-prompt-driven textual knowledge graph generation with heterogeneous information embedding for enhancing the representation of multi-field categorical data. Specifically, we design two prompt templates to transform multi-field categorical data into a textual knowledge graph for problem reformulation, and propose a heterogeneous information embedding approach that works by jointly representing the generated text and knowledge graph in the setting of cross-attention gated modality fusion with feature alignment. The experimental results indicate that our approach outperforms those baselines of click-through rate prediction. The results also indicate that each module of our approach is useful for improving the performance¹.

1 Introduction

Click-Through Rate (CTR) prediction is vital for online advertising and e-commerce recommendations, directly affecting platform revenue and user experience (Yang and Zhai, 2022). It aims to estimate user click probability by learning feature embeddings from multi-field categorical data. CTR models have evolved through three stages.

Traditional methods, such as Logistic Regression (LR) (Richardson et al., 2007) and Factorization Machines (FMs) (Rendle, 2010; Pan et al., 2018), model linear and low-order feature interactions. Deep learning models (Qu et al., 2016; Wang et al., 2017; Guo et al., 2017; Lian et al., 2018; Zhou et al., 2019; Song et al., 2019) capture high-order interactions and dynamic user interests. Advances in Natural Language Processing (NLP) and Knowledge Graph (KG) further boost CTR tasks. Specifically, text-enhanced models (Muhamed et al., 2021; Fu et al., 2025; Lin et al., 2024) extract textual semantics, while KG-enhanced models (Wang et al., 2018, 2020) mitigate data sparsity via structured knowledge. Despite advances, critical challenges persist: **(1) Semantic deficiency in traditional categorical feature embedding:** One-hot and index encoding cannot capture semantic correlations between categorical features, leading to high dimensionality, heavy manual dependence and poor semantic expression. **(2) Insufficient collaborative learning from heterogeneous information:** Most models adopt single-modal representation and cannot integrate textual semantics and structural knowledge, failing to exploit cross-modal complementary information. **(3) Inefficient multimodal fusion:** Existing multimodal schemes adopt simple fusion strategies lacking precise feature alignment, which limits fine-grained dependency modeling and adaptive weight allocation across modalities.

To address these challenges, we propose **Text-KG Embedding for CTR Prediction (TKE4CTR)**, a **dual-prompt-driven multimodal framework** centered on Text-KG dual-modal fusion. TKE4CTR converts multi-field categorical data into textual and KG representations via designed prompts, integrates embeddings using a cross-attention gated modality fusion module, and adopts a feature-gated MLP for CTR inference. Our main contributions are as follows:

(1) We propose a dual-prompt-driven modality

*Corresponding author

¹<https://github.com/fx19/TKE4CTR>

transformation paradigm with two tailored prompts to convert multi-field categorical data into unified textual forms and task-adaptive KGs, reducing dependence on manual schemas and external knowledge, and alleviating semantic loss and dimensionality curse in discrete feature encoding.

(2) We design a collaborative heterogeneous embedding scheme with modality-specialized encoders to jointly learn textual and structural representations, enabling cross-modal complementarity and mitigating single-modal limitations.

(3) We propose a cross-attention gated fusion mechanism to align heterogeneous embeddings, capture fine-grained cross-modal correlations, and adaptively adjust modality weights, reducing the redundancy of simple fusion schemes.

(4) Extensive experiments demonstrate that our TKE4CTR outperforms various competitive baseline methods in AUC. Ablation studies validate the effectiveness of different modalities, the role of each core module, and their mutual synergy.

2 Related Work

2.1 Discrete Feature-Based CTR Prediction

Discrete ID and categorical features prevail in CTR tasks and are generally one-hot encoded. Classic methods include linear LR (Richardson et al., 2007) and various types of FMs (Rendle, 2010; Pan et al., 2018) for low-order feature interaction modeling.

Deep models tackle such flaws and fall into four categories: (1) Interaction-focused models: PNN (Qu et al., 2016), DCN (Wang et al., 2017), DeepFM (Guo et al., 2017) and xDeepFM (Lian et al., 2018); (2) Sequence-based interest models: DIN (Zhou et al., 2018), DIEN (Zhou et al., 2019) and SIM (Pi et al., 2020); (3) Sparse learning optimized models: AutoInt (Song et al., 2019), FiBiNET (Huang et al., 2019) and AFN (Cheng et al., 2020); (4) Deployment-efficient models: IntTower (Li et al., 2022) and FinalMLP (Mao et al., 2023).

2.2 Text Embedding-Based CTR Prediction

Traditional CTR models fail to exploit the semantics of categorical features, limiting user-item modeling and generalization. Driven by NLP advances, Pre-trained Language Model (PLM) and Large Language Model (LLM) have been widely applied to CTR tasks (Qiu et al., 2020). Classic PLMs, including BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019), extract semantic information from categorical data, while advanced LLMs cap-

ture latent user preferences. Representative PLM-based works, such as CTR-BERT (Muhammed et al., 2021), DCAF-BERT (Muhammed et al., 2022) and BERT4CTR (Wang et al., 2023), integrate textual and numerical features. In contrast, LLM-based methods including LLMRank (Hou et al., 2024), FLIP (Wang et al., 2024) and ClickPrompt (Lin et al., 2024) mitigate the inherent limitations of one-hot encoding via external semantic knowledge. Uni-CTR (Fu et al., 2025) works by learning universal semantic representations for multi-domain CTR prediction, and iFusion (Ni et al., 2026) utilizes diffusion models to fuse dynamic user interests.

2.3 KG-Based CTR Prediction

Rich entity and relational connections in KG mitigate data sparsity of CTR tasks (Jin and Yang, 2025). Current KG-based CTR prediction mainly involves three branches. (1) KGE techniques map entities and relations into low-dimensional embeddings. Typical models like DKN (Wang et al., 2018), DMCM (Wang et al., 2019), KGFER (Fan et al., 2022) and KGBPR (Ma et al., 2023) generally take TransE (Bordes et al., 2013) and TransR (Lin et al., 2015) as basic structures. (2) GNN-based methods, such as CKAN (Wang et al., 2020), MCCLK (Zou et al., 2022) and KNI (Qu et al., 2019), enhance user and item embeddings via multi-hop relational aggregation. (3) Hybrid architectures tightly integrate KGE and CTR modules, such as KGCAN (Elahi et al., 2024) and RAKCR (Cui et al., 2024). Recent works like DH-GCN (Jin et al., 2026), ProGraph (Wei et al., 2026) and GenCI (Ou et al., 2026) improve graph modeling and user mining to lift model performance.

3 Proposed Problem Formulation

For the CTR prediction task, we define a user set U , an item set P , a text set T , and a knowledge graph $G = (V, E)$. Here, T denotes unified prompt-generated narratives from multi-field data, and G denotes prompt-constructed task-adaptive heterogeneous graphs. Each sample is represented as a 4-tuple $(u, p, t_{u,p}, g_{u,p})$, where $u \in U$, $p \in P$, $t_{u,p} \in T$, and $g_{u,p} \subseteq G$. The CTR task aims to learn a mapping function f to predict user-item click probability $\hat{y}_{u,p}$. The framework consists of four core functional modules: text encoder E_T , graph encoder E_G , cross-attention (CA) gated fusion module F , and prediction module f_{pred} . We utilize encoders E_T and E_G to generate text embed-

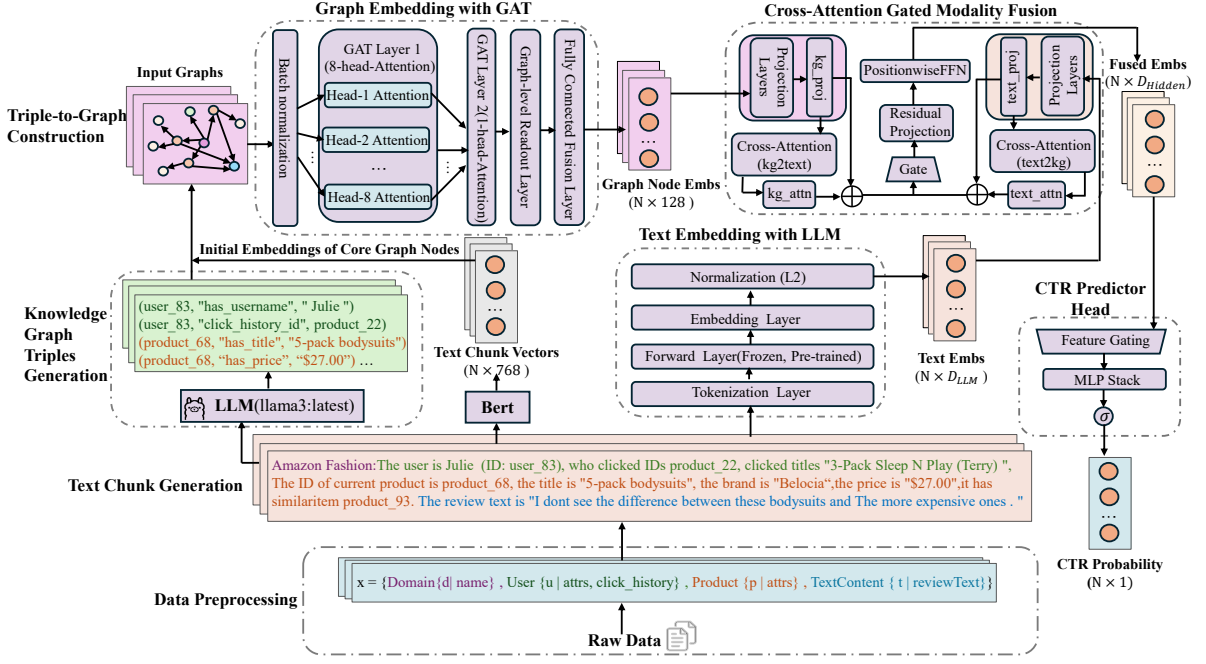


Figure 1: Overall framework of TKE4CTR consists of four core modules, namely, dual-prompt modality conversion, dual-modality encoder, cross-attention gated fusion and CTR prediction.

ding h_T and graph embedding h_G , respectively:

$$h_T = E_T(t_{u,p}), \quad (1)$$

$$h_G = E_G(g_{u,p}). \quad (2)$$

The cross-attention gated module aligns and fuses h_T and h_G to obtain fused embedding z_f :

$$\begin{aligned} z_f &= F(h_T, h_G) \\ &= CA(h_T, h_G) + CA(h_G, h_T). \end{aligned} \quad (3)$$

The prediction head outputs a click probability:

$$\hat{y}_{u,p} = f_{\text{pred}}(z_f). \quad (4)$$

4 Text-KG Embedding for CTR

4.1 Overview

Fig. 1 shows the pipeline of TKE4CTR, which performs end-to-end embedding, alignment and fusion of KG structure and text semantics for CTR prediction. The framework includes four modules: (1) **Dual-Prompt-Driven Modality Transformation**: Transform tabular features into structured text and KG subgraphs via prompts and Llama-3 (Grattafiori et al., 2024). (2) **Dual-Modality Feature Encoder**: Generate graph embedding h_G using GAT (Veličković et al., 2018) and semantic embedding h_T via frozen LLM. (3) **Cross-Attention Gated Modality Fusion**: Fuse h_T and h_G with bidirectional cross-attention and gate mechanism. (4) **CTR Prediction Head**: Adopt gated MLP to predict click probability from fused embedding.

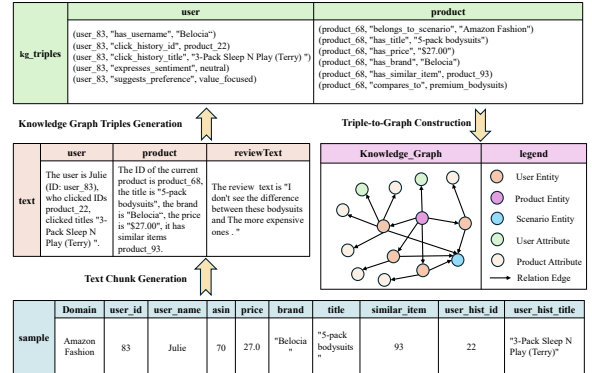


Figure 2: Case 1: Illustration of Dual-Prompt-Driven Modality Transformation Pipeline.

4.2 Dual-Prompt-Driven Modality Transformation

As shown in Fig. 2, our module converts categorical data into structured KG in three stages:

Text Chunk Generation. To capture categorical semantics in cleaned merged tabular data, we design a **prompt-template-based text chunk generation strategy**. The generated text chunks consist of four components: domain (d), user information (u), product information (p), and review text (t). We adopt a unified prompt template to aggregate multi-field categorical sequences x_{text} . The detailed prompt template for text chunk generation

is formulated as follows:

$x_{\text{text}} = [\text{Domain Name}]: \text{The user is } [\text{Username}],$
 who clicked IDs [Click ID 1,2]
 and clicked titles [Click Title 1,2] recently.
 The ID of the current product is [Product ID],
 the title is [Product Title],
 the brand is [Brand],
 the price is [Price],
 it has similar items [Similar Product ID].
 The review text is [Review].

(5)

In the template, all placeholders (e.g., [Domain Name], [Username], [Product ID]) are populated with discrete field values derived from the text chunks. This standardized prompt template ensures structural consistency of the generated text chunks, which in turn facilitates effective semantic extraction and stable model training in subsequent procedures.

KG Triples Generation. Compared with conventional rule-based and pipeline generation approaches, LLMs achieve better semantic comprehension and open-domain reasoning, supporting high-quality triple extraction and implicit sentiment mining to enrich KG semantics. However, LLM-generated triples inevitably contain hallucinations. To address this issue, we propose a two-stage strategy for hallucination suppression. During generation, structured constrained few-shot prompts regularize outputs using fixed relation categories, unified entity standards and standard samples, avoiding unfounded content and schema conflicts. Candidate triples are assessed from entity validity, relational logic and text consistency. Unqualified results will be regenerated up to three times. Text chunks failing repeated verification are filtered by rules to retain valid triples. The strategy effectively reduces hallucinations and annotation costs.

Triple-to-Graph Construction. We build heterogeneous knowledge subgraphs for each text chunk to generate valid inputs for GAT. Entities are defined as nodes equipped with multi-dimensional features, and extracted triples correspond to directed edges carrying relational types and weight information. This local subgraph construction method satisfies two practical requirements: (1) It reduces resource overhead compared with a single global knowledge graph and improves model

scalability. (2) It maintains semantic coherence between subgraphs and original texts and eliminates illogical semantic connections.

4.3 Dual-Modality Feature Encoder

4.3.1 Graph Embedding with GAT

We adopt GAT to embed subgraphs $g_{u,p}$ via dynamic neighbor weighting. BERT is used to extract features $\mathbf{x}_v^{\text{bert}}$ from core nodes to reduce noise. L2-normalized feature concatenation is taken for obtaining node embeddings:

$$\mathbf{x}_v^{(0)} = \text{L2Norm} \left(\mathbf{x}_v^{\text{type}} \parallel \mathbf{x}_v^{\text{id}} \parallel \mathbf{x}_v^{\text{bert}} \cdot \mathbb{I}[v \in \mathcal{V}_{\text{core}}] \parallel \mathbf{x}_v^{\text{struct}} \right). \quad (6)$$

We employ a two-layer GAT to learn graph representations from initialized node features. The first layer involves 8 attention heads to calculate normalized attention coefficients, aggregate neighbor features and concatenate multi-head outputs:

$$\alpha_{vu}^k = \frac{\exp(\sigma_L(\mathbf{a}_k^\top [\mathbf{z}_{v,k} \parallel \mathbf{z}_{u,k}]))}{\sum_{w \in \mathcal{N}(v)} \exp(\sigma_L(\mathbf{a}_k^\top [\mathbf{z}_{v,k} \parallel \mathbf{z}_{w,k}]))}. \quad (7)$$

The second layer involves single-head attention and mean aggregation to refine embeddings. We extract final heterogeneous graph embedding $\mathbf{h}_G$ from core nodes. The GAT is optimized via CTR labels, and trained parameters are used for validation and test sets. More details are in Appendix A.2.

4.3.2 Text Embedding with Frozen PLM

We utilize a frozen PLM to extract comprehensive contextual text embeddings. Given a text chunk $t_{u,p} = \{t_1, \dots, t_n\}$, we first tokenize it:

$$\mathbf{X}_{t_{u,p}} = \text{Tokenize}(t_{u,p}), \quad \mathbf{X}_{t_{u,p}} \in \mathbb{N}^L. \quad (8)$$

Then we feed these tokens into the model to obtain dense hidden semantic features:

$$\mathbf{H}_{t_{u,p}} = \text{PLM}(\mathbf{X}_{t_{u,p}}), \quad \mathbf{H}_{t_{u,p}} \in \mathbb{R}^{L \times d}. \quad (9)$$

Mean pooling yields L2-normalized $\mathbf{h}_T$:

$$\mathbf{h}_T = E_T(t_{u,p}) = \frac{\bar{\mathbf{H}}_{t_{u,p}}}{\|\bar{\mathbf{H}}_{t_{u,p}}\|_2} \in \mathbb{R}^d. \quad (10)$$

Powered by self-attention and pre-trained knowledge, $\mathbf{h}_T$ effectively captures global textual semantics with steady representation ability.

4.4 Cross-Attention Gated Modality Fusion

Heterogeneous knowledge from KG and text shows inherent modal complementarity, with structured relational facts and rich semantic content, respectively. We design a cross-attention gated fusion module for cross-modal alignment and adaptive feature aggregation, containing four functional units:

Modality Projection to Shared Space. To alleviate cross-modal distribution discrepancy, we map KG embedding $\mathbf{h}_G \in \mathbb{R}^{d_G}$ and text embedding $\mathbf{h}_T \in \mathbb{R}^{d_T}$ into a unified latent space:

$$\tilde{\mathbf{h}}_G = \text{LN}(\text{Swish}(\mathbf{W}_G \mathbf{h}_G + \mathbf{b}_G)), \quad (11)$$

$$\tilde{\mathbf{h}}_T = \text{LN}(\text{Swish}(\mathbf{W}_T \mathbf{h}_T + \mathbf{b}_T)). \quad (12)$$

Bidirectional Cross-Attention Interaction. We adopt 8-head bidirectional cross-attention to capture fine-grained cross-modal dependencies. Here, d_h denotes the dimension per attention head. $\sigma_s(\cdot)$ indicates row-wise softmax, which operates along the last dimension of the attention-score matrix. Suppose $\tilde{\mathbf{h}}_G, \tilde{\mathbf{h}}_T \in \mathbb{R}^{N \times d_h}$, then $\mathbf{A}_{GT}, \mathbf{A}_{TG} \in \mathbb{R}^{N \times N}$.

$$\mathbf{A}_{GT} = \sigma_s \left(\frac{\tilde{\mathbf{h}}_G \tilde{\mathbf{h}}_T^\top}{\sqrt{d_h}} \right), \quad \mathbf{H}'_G = \mathbf{A}_{GT} \tilde{\mathbf{h}}_T, \quad (13)$$

$$\mathbf{A}_{TG} = \sigma_s \left(\frac{\tilde{\mathbf{h}}_T \tilde{\mathbf{h}}_G^\top}{\sqrt{d_h}} \right), \quad \mathbf{H}'_T = \mathbf{A}_{TG} \tilde{\mathbf{h}}_G. \quad (14)$$

Adaptive Gated Feature Fusion. A lightweight adaptive gating unit is introduced to dynamically adjust the weight ratio between original modal features and cross-modal interactive features:

$$\mathbf{g} = \sigma \left(\mathbf{W}_g [\tilde{\mathbf{h}}_G \parallel \mathbf{H}'_G \parallel \tilde{\mathbf{h}}_T \parallel \mathbf{H}'_T] + \mathbf{b}_g \right), \quad (15)$$

$$\begin{aligned} \mathbf{h}_G^f &= \mathbf{g} \odot \mathbf{H}'_G + (1 - \mathbf{g}) \odot \tilde{\mathbf{h}}_G, \\ \mathbf{h}_T^f &= \mathbf{g} \odot \mathbf{H}'_T + (1 - \mathbf{g}) \odot \tilde{\mathbf{h}}_T. \end{aligned} \quad (16)$$

Unified Cross-Modal Representation. We further combine residual projection and feed-forward network to aggregate dual-modal gated features and obtain the final comprehensive representation:

$$\begin{aligned} \mathbf{h}_f &= \frac{1}{2} (\mathbf{h}_G^f + \mathbf{h}_T^f), \\ \mathbf{z}_f &= \text{FFN} \left(\mathbf{h}_f + \text{LN} \left(\mathbf{W}_r [\mathbf{h}_G^f \parallel \mathbf{h}_T^f] \right) \right). \end{aligned} \quad (17)$$

$\mathbf{z}_f$ is obtained by simultaneously fusing structural prior information from KG and contextual semantic information from plain text.

4.5 CTR Prediction Head

The fused representation $\mathbf{z}_f \in \mathbb{R}^{d_f}$ is fed into the CTR prediction head to estimate the click probability of user-item pair (u, p) . A feature-gated MLP is used for adaptive feature modulation:

$$\tilde{\mathbf{z}}_{u,p} = \mathbf{g}_f \odot \mathbf{z}_f, \quad \mathbf{g}_f = \sigma(\mathbf{W}_p \mathbf{z}_f + \mathbf{b}_p). \quad (18)$$

The gated $\tilde{\mathbf{z}}_{u,p}$ is processed by fully connected layers, and the predicted click probability is:

$$\hat{y}_{u,p} = \sigma(\mathbf{w}_o^\top \tilde{\mathbf{z}}_{u,p} + b_o). \quad (19)$$

5 Experiments

5.1 Experimental Settings

Table 1: Statistics of the Amazon Review Data.

Domains	Users	Products	Samples
All Beauty	319,335	32,486	371,345
Amazon Fashion	749,233	186,189	883,636
Digital Music	127,174	66,010	1,584,082
Gift Cards	128,873	1,547	147,194
Mus. Instruments	903,060	112,132	1,512,530

Datasets Classical CTR benchmarks such as Criteo and Avazu (Zhu et al., 2021) convert raw features into anonymous identifiers in preprocessing. Such operation leads to hiding inherent semantic correlations, making these datasets incompatible with TKE4CTR. As shown in Table 1, we adopt the 2018 Amazon Review Dataset (Ni et al., 2019) and select five product domains covering twelve years of real user-item interactions. To avoid data leakage, we split the dataset temporally into training/validation/test sets at 80%/10%/10%, in which samples with ratings above 3 are treated as positive and the rest as negative.

The training set covers interactions from earlier time sequences. We adopt static basic attributes (e.g., Domain, user_id, asin, price, brand and title), the target user’s click history features (e.g., user_hist_id and user_hist_title), and review text to construct LLM-derived semantic triples for representation enhancement. This enhancement method, used only during training, injects fine-grained user preferences and item semantic knowledge extracted from reviews and user click history, helping the model learn richer semantic patterns, without introducing post-click interaction signals during evaluation. In the validation and test sets, only the basic static attributes are kept, while interaction

Table 2: CTR prediction performance (AUC). Best baseline: underlined, best overall: **bold**.

Models	All Beauty		Amazon Fashion		Digital Music		Gift Cards		Mus. Instruments	
	AUC	RI	AUC	RI	AUC	RI	AUC	RI	AUC	RI
PNN	0.600	161.6%	0.6979	56.7%	0.5904	266.0%	0.5959	240.0%	0.6859	55.3%
DCN	0.620	118.0%	0.6985	56.3%	0.5919	260.0%	0.6126	190.0%	0.6893	52.5%
DeepFM	0.605	150.0%	0.6982	56.5%	0.5917	261.0%	0.5937	248.0%	0.6880	53.5%
xDeepFM	0.622	114.4%	0.7031	52.7%	0.5957	246.0%	0.6121	191.0%	0.6892	52.6%
DIEN	0.618	121.7%	0.6995	55.5%	0.5900	267.7%	0.6105	195.0%	0.6881	53.4%
AutoInt	0.608	142.2%	0.7003	54.9%	0.5913	262.4%	0.5976	234.0%	0.6888	52.9%
FiBiNET	0.620	118.0%	0.6770	75.3%	0.5832	297.7%	0.6120	191.0%	0.6878	53.7%
IntTower	0.618	121.7%	0.6988	56.0%	0.5920	259.7%	0.6100	196.0%	0.6888	52.9%
FinalMLP	0.6130	131.5%	0.6907	62.7%	0.5947	249.4%	0.6082	201.0%	0.6884	53.2%
Uni-CTR	<u>0.725</u>	16.27%	<u>0.7523</u>	23.0%	<u>0.6140</u>	190.3%	<u>0.7246</u>	45.2%	<u>0.7569</u>	12.3%
TKE4CTR	0.7616	-	0.8102	-	0.8309	-	0.8261	-	0.7886	-

features on review text and user click history are completely removed. Knowledge-graph-based text trunks are fully built from user and product-related basic information, avoiding leakage of post-click interaction information during evaluation.

Evaluation Metrics Following prior works, we adopt AUC and RelImpr (RI) for evaluation, with RI defined as:

$$RI = \left(\frac{AUC_{\text{target}} - 0.5}{AUC_{\text{baseline}} - 0.5} - 1 \right) \times 100\%. \quad (20)$$

Note that a slight increase in AUC at the 0.001 level is known to be a significant improvement for CTR prediction (Cheng et al., 2020).

Baselines We compare our approach with representative CTR prediction baselines: **PNN** (Qu et al., 2016) employs a product layer to model complex feature interactions via inner/outer product operations; **DCN** (Wang et al., 2017) employs cross-network modules to capture bounded-degree feature interactions with high efficiency. **DeepFM** (Guo et al., 2017) combines FM (Rendle, 2010) for low-order and deep networks for high-order feature interactions; **xDeepFM** (Lian et al., 2018) augments DeepFM with compressed interaction network (CIN) to model high-order feature interactions; **DIEN** (Zhou et al., 2019) captures dynamic user preference trajectories via interest extraction and evolution layers; **AutoInt** (Song et al., 2019) uses multi-head self-attention to learn hierarchical high-order feature interactions from raw features; **FiBiNET** (Huang et al., 2019) employs bilinear modules and SE-attention for dynamic

feature weighting and complex interaction modeling; **IntTower** (Li et al., 2022) is a pre-ranking two-tower model that strengthens user-item interactions for accuracy and efficiency; **FinalMLP** (Mao et al., 2023) adopts dual-stream MLP to learn CTR feature interactions with parallel representations; **Uni-CTR** (Fu et al., 2025) is an LLM-powered multi-domain unified CTR framework with domain-adaptive components.

Implementation Details All experiments are implemented in PyTorch on NVIDIA A100-40GB GPUs. For the KG modality, triples are generated from text chunks using LLaMA-3 (Grattafiori et al., 2024) via Ollama, and local subgraphs are encoded using a two-layer GAT. The first GAT layer involves 8 attention heads and the second layer has a single head, producing 128-dimensional node embeddings. For the text modality, we use Qwen3-4B (Yang et al., 2025) to extract contextualized embeddings. Each text trunk is truncated or padded to 256 tokens. The embedding of the last non-padding token is produced as the text representation, yielding a 2560-dimensional vector that is subsequently L2-normalized. Graph and text embeddings are fused through a cross-attention gated module with 8 attention heads and a hidden dimension of 512, resulting in a unified 512-dimensional representation. Dropout with a rate of 0.3 is applied in both the fusion module and the CTR prediction head that consists of a single hidden layer with 256 units, ReLU activation and layer normalization.

The model is trained for a maximum of 50 epochs using the AdamW optimizer (Kingma and

Table 3: Input modality ablation results on two datasets

Modality	All Beauty	Amazon Fashion
Text-KG	0.7616	0.8102
w/o KG (only Text)	0.7411	0.7305
w/o Text (only KG)	0.6624	0.7470
only ID	0.5985	0.6518

Table 4: Component ablation results on two datasets

Models	All Beauty	Amazon Fashion
TKE4CTR	0.7616	0.8102
w/o Cross-Attn Gate	0.7593	0.8021
w/o Feat-Gated Head	0.7585	0.8082

Ba, 2015) with a learning rate of 1×10^{-5} and a weight decay of 1×10^{-4} . A cosine learning rate schedule with warm-up is employed, and binary cross-entropy loss is adopted as the training objective. Early stopping is triggered if the validation AUC is not improved for 25 consecutive epochs. All experiments are conducted with a fixed random seed of 42 to ensure reproducibility.

5.2 Performance Comparison

As shown in Table 2, existing CTR models have complementary advantages and limitations. **PNN** and **DCN** capture low-order feature interactions, yielding AUCs below 0.70. High-order interaction models (**DeepFM**, **xDeepFM**) achieve slight improvements, with **xDeepFM** performing best on the second dataset. Attention and sequential models (**DIEN**, **AutoInt**) enhance performance via adaptive feature weighting, while structure-enhanced models (**FiBiNET**, **IntTower**) yield stable results. **Uni-CTR** achieves the strongest baseline performance (0.6140-0.7569) through unified multi-domain representation learning.

Furthermore, TKE4CTR substantially outperforms all baselines, achieving state-of-the-art AUCs from **0.7616 to 0.8309** across all datasets. It improves over Uni-CTR by **12.3% to 190.3%**, with the smallest gain on the fifth dataset and the largest on the third one. Notably, TKE4CTR shows significant advantages on datasets where baselines perform poorly, demonstrating its ability to capture complex feature interactions and cross-domain dependencies for accurate CTR prediction.

5.3 Ablation Studies

We showcase results on the first two datasets. All others are in Tables 12 and 13 in Appendix A.4.

Impact of Input Modalities. We explore the effects of only Text, only KG and combined Text-KG modalities on CTR prediction through modal ablation, taking ID-only input as the baseline. Table 3 presents AUC results on All Beauty and Amazon Fashion datasets. Text-KG dual-modal fusion outperforms the ID-only scheme, enriching feature information via textual semantics and relational knowledge. Moreover, the contribution degrees of the two modalities differ by datasets. Text information takes a dominant position on All Beauty, while relational knowledge contributes more significantly to prediction on Amazon Fashion.

Impact of Model Components. Beyond input modalities, we further examine two architectural choices in TKE4CTR, namely, the cross-attention gated fusion module and the feature-gated MLP prediction head. We remove each component independently and report AUC in Table 4. TKE4CTR achieves the best AUC, verifying the complementarity of the two components. Removing the cross-attention gate leads to a decrease of AUC by 0.0023 and 0.0081, while removing the feature-gated head makes AUC reduced by 0.0031 and 0.0020, respectively. The consistent drops demonstrate that both modules enhance ranking quality.

5.4 Impact of LLM on KG Triples Generation

The selection of LLMs is critical to KG triples generation quality. To identify the optimal model for downstream tasks, we fix prompt templates and input samples, evaluating three representative 8B-scale LLMs: Llama3 (Grattafiori et al., 2024), Qwen3 (Yang et al., 2025), and Deepseek-R1 (DeepSeek-AI et al., 2025). We adopt an intrinsic evaluation from textual grounding and schema compliance perspectives, using five metrics: *Evidence Support* (ES), *Relation Validity* (RV), *Information Coverage* (IC), *Noise Control* (NC), and *Relation Consistency* (RC) (Salman et al., 2024) (detailed definitions in Appendix A.5). The overall score is computed as the arithmetic mean of the five individual metrics:

$$S_{\text{avg}} = \frac{1}{5} (s_{\text{ES}} + s_{\text{RV}} + s_{\text{IC}} + s_{\text{NC}} + s_{\text{RC}}). \quad (21)$$

As shown in Table 5 and Figure 3, Llama3 achieves the highest average score ($S_{\text{avg}} = 0.886$), outperforming Qwen3 (0.859) and Deepseek-r1 (0.790). It ensures valid structure and standard

Table 5: KG triples quality resulting from LLMs

LLM	S_{avg}	ES	RV	IC	NC	RC
Llama3	0.886	0.898	0.984	0.679	0.941	0.929
Qwen3	0.859	0.800	0.920	0.745	0.831	1.000
Deepseek-r1	0.790	0.903	0.710	0.783	0.986	0.567

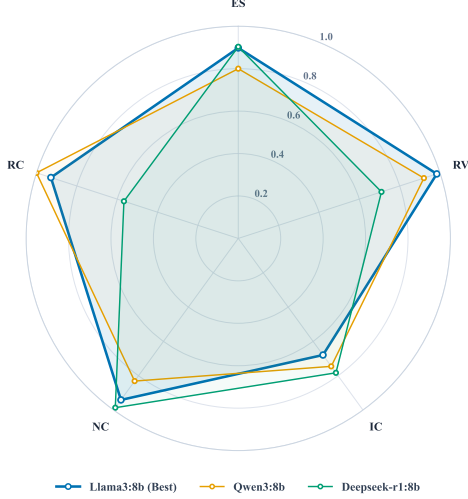


Figure 3: Radar chart comparing five evaluation metrics of KG triples.

schema with balanced performance. Qwen3 gains perfect RC and competitive IC but falls behind in ES and NC. Deepseek-R1 performs well on ES, IC and NC yet has poor RV and RC, leading to structural errors and irregular relation naming. Considering generation quality, schema conformity and output stability, we adopt **Llama3** as a core model for subsequent KG triples generation.

5.5 Hallucination Mitigation Analysis

We adopt three evaluation metrics: Natural Language Inference Factual Fidelity (NLI-F), Rule-based Fidelity (R-F), and Hallucination Rate (HR) (Lavrinovics et al., 2025). NLI-F measures semantic consistency between generated triples and original text. R-F evaluates triple compliance with predefined KG constraints. HR reflects model hallucination by counting logically invalid triples. Corresponding formulas are given in Appendix A.6.

Comparison of Three Prompt Variants. Table 6 reports the generation performance under three prompt designs, with full prompt templates provided in Appendix A.1. The Baseline Prompt (BP) yields low NLI-F and R-F but an extremely high HR of 84.92%, where NLI-F is 15.41% and R-F is 14.76%. This suggests that unconstrained baseline prompts fail to regulate LLM outputs, lead-

Table 6: Performance of different prompt strategies

Prompt	Samples	Triples	NLI-F(%)	R-F(%)	HR(%)
BP	100	1128	15.41	14.76	84.92
SCZP	100	896	58.37	61.42	59.16
SCFP	100	835	72.15	80.34	36.28

Table 7: Ablation Results of LLM secondary validation

Condition	Samples	Triples	NLI-F(%)	R-F(%)	HR(%)
w/o Valid.	100	835	72.15	80.34	36.28
w/ Valid.	100	779	77.53	85.67	33.42

ing to severe structural and semantic hallucinations. The Structured Constraints Zero-shot Prompt (SCZP) substantially boosts NLI-F to 58.37% and R-F to 61.42%, while lowering HR to 59.16%. Incorporating few-shot examples, the Structured Constrained Few-shot Prompt (SCFP) achieves the best NLI-F of 72.15% and R-F of 80.34%, together with the lowest HR of 36.28%.

Ablation Analysis of LLM Secondary Validation.

We conduct ablation experiments to verify the effectiveness of the secondary validation pipeline. As shown in Table 7, introducing validation mechanism and iterative regeneration raises NLI-F from 72.15% to 77.53% and pushes R-F from 80.34% up to 85.67%, while HR is reduced from 36.28% to 33.42%. The slight reduction in extracted triples results from screening low-quality outputs, demonstrating that the proposed pipeline can effectively alleviate implicit hallucinations and improve overall extraction reliability.

5.6 PLM Performance on Text Embedding

We investigate PLM impacts on text embedding and CTR prediction with six pre-trained models, namely, BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), DeBERTa-v3-large (He et al., 2023), Sheared-LLaMA-1.3B (Xia et al., 2023), Hunyuan-4B-Instruct and Qwen3-4B (Yang et al., 2025). Partial results are displayed in Figure 4, and complete data are provided in Appendix A.7. Qwen3-4B delivers the best performance, with Hunyuan-4B-Instruct and Sheared-LLaMA-1.3B achieving similar outcomes. Divergent semantic modeling abilities cause the performance discrepancy. Table 8 shows a clear trade-off between model performance and efficiency. Lightweight models run quickly but lack strong representa-

Table 8: Model Inference Performance.

Model	Params	Throughput (samples/s)	Latency (s/sample)
BERT	110M	265.51	0.0038
GPT-2	124M	250.46	0.0040
DeBERTa-v3-large	400M	17.22	0.058
Sheared-LLaMA-1.3B	1.3B	56.99	0.018
Hunyuan-4B-Instruct	4.0B	3.42	0.292
Qwen3-4B	4.0B	23.52	0.043

Table 9: Model Parameters and AUC Performance

Model	Params	All Beauty	Amazon Fashion
Uni-CTR (Sheared-LLaMA)	1.3B + 9.71M	0.7250	0.7523
TKE4CTR (Sheared-LLaMA)	1.3B + 7.57M	0.7530	0.7580
TKE4CTR (Qwen3)	4.0B + 9.01M	0.7616	0.8102

tion capacity. Having comparable parameter size, Hunyuan-4B-Instruct consumes more computing resources. Qwen3-4B attains satisfying prediction accuracy at lower cost. We finally adopt Qwen3-4B as the text encoder in our framework.

5.7 Parameter Sensitivity Analysis

Effect of Attention Head Number in Cross-Attention Gated Modality Fusion This section explores the effect of the cross-attention head count on CTR prediction, revealing a nonlinear performance correlation. Increasing the number of heads from 2 to 8 improves AUC and LogLoss, with latency rising moderately to ~ 0.15 ms/sample, as shown in Figure 5 and Appendix A.8. More heads capture fine-grained cross-modal correlations. Further expanding to 16 brings marginal gains below 0.001, yet latency jumps to ~ 0.32 ms/sample, showing performance saturation and higher computational cost. The 8-head configuration achieves optimal stable performance and balances prediction accuracy and inference efficiency.

5.8 Parameter Effectiveness Analysis

Table 9 presents parameter scale and AUC results of Uni-CTR and our TKE4CTR framework. Using Sheared-LLaMA-1.3B for fair comparison, TKE4CTR attains higher AUC with fewer train-

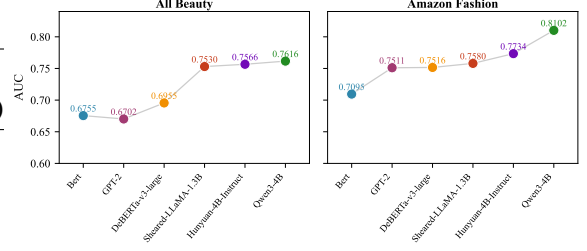


Figure 4: Performance comparison of different PLMs on two datasets (All Beauty and Amazon Fashion).

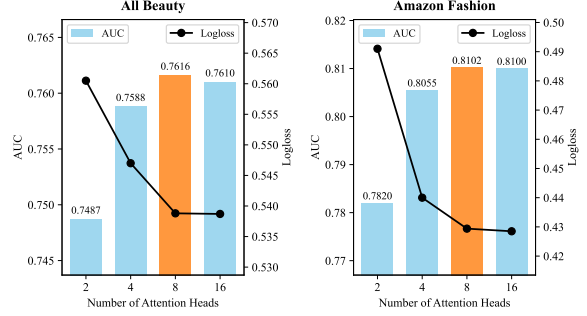


Figure 5: Effect of Attention Head Number in Cross-Attention Gated Modality Fusion on CTR Performance.

able parameters, validating its efficiency and modeling ability. Switching to Qwen3-4B gains steady improvement, and the corresponding TKE4CTR variant yields the optimal AUC overall.

6 Conclusion

In this paper, we have proposed an approach of dual-prompt-driven textual knowledge graph generation with heterogeneous information embedding for CTR. Specifically, we have designed two prompt templates to transform multi-field categorical data into text and knowledge graphs for problem reformulation and accordingly propose a heterogeneous information embedding approach that works by jointly representing the two generated modalities in the setting of cross-attention gated modality fusion with feature alignment. The experimental results indicate that our approach outperforms those state-of-the-art methods. The results of the ablation experiments show that each component of the proposed approach is effective and necessary in improving the CTR prediction performance. In the future, we will explore how our approach can be transferred to other similar tasks, such as user behavior prediction.

Limitations

Our current work focuses on generating textual KGs for CTR prediction, which uses LLMs and prompt templates, and may encounter inherent hallucination issues. To address this problem, future work will explore retrieval-augmented strategies to mitigate LLM hallucinations and further improve the factual accuracy and stability of KG construction. Moreover, KG generation via LLMs involves heavy computational overhead and slow inference speed, restricting its deployment in high-throughput and real-time scenarios. We will tackle this issue by investigating lightweight model optimization for efficient LLM-based KG generation. Beyond these two optimizations, we will also extend our method to cross-task user behavior prediction and build dynamic session-level knowledge graphs, so as to better capture dynamic user interests in real-world recommendation systems.

Ethics Statement

Our work focuses on CTR prediction for recommendation systems, where all datasets used in our experiments are publicly available. We have properly cited these datasets and used them strictly in accordance with the terms of their respective licenses and intended use cases. This research is based solely on the pre-collected and publicly released data provided by the dataset owners, and we do not collect or process any additional raw user data or personally identifiable information. Moreover, this work does not involve human subjects or animal experiments. Therefore, we do not anticipate any ethical issues arising from the work presented in this paper.

Acknowledgments

This paper is supported by Natural Science Foundation of Shandong Province (Grant ZR2025QC1591), National Natural Science Foundation of China (Grant 62276168), Natural Science Foundation of Guangdong Province (Grant 2023A1515010869), Guangdong Provincial Key Laboratory (Grant 2023B1212060076) and Shenzhen Science and Technology Program (Grant ZDSYS20220527171400002).

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko.

2013. [Translating embeddings for modeling multi-relational data](#). In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, volume 2, pages 2787–2795.

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*, volume 34, pages 3609–3616.

Yachao Cui, Hongli Yu, Xiaoxu Guo, Han Cao, and Lei Wang. 2024. [RAKCR: Reviews sentiment-aware based knowledge graph convolutional networks for personalized recommendation](#). *Expert Systems with Applications*, 248:123403.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z.F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 83 others. 2025. [DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ehsan Elahi, Sajid Anwar, Babar Shah, Zahid Halim, Abrar Ullah, Imad Rida, and Muhammad Waqas. 2024. [Knowledge graph enhanced contextualized attention-based network for responsible user-specific recommendation](#). *ACM Transactions on Intelligent Systems and Technology*, 15(4):83.

Huilian Fan, Yuanchang Zhong, Guangpu Zeng, and Chenhao Ge. 2022. [Improving recommender system via knowledge graph based exploring user preference](#). *Applied Intelligence*, 52(9):10032–10044.

Zichuan Fu, Xiangyang Li, Chuhan Wu, Yichao Wang, Kuicai Dong, Xiangyu Zhao, Mengchen Zhao, Huifeng Guo, and Ruiming Tang. 2025. [A unified framework for multi-domain ctr prediction via large language models](#). *ACM Transactions on Information Systems*, 43:117.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 81 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. [DeepFM: A factorization-machine based neural network for ctr prediction](#).

- In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 1725–1731.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [DeBERTaV3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023)*.
- Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. [Large language models are zero-shot rankers for recommender systems](#). In *Advances in Information Retrieval*, pages 364–381.
- 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*, pages 169–177.
- Ying Jin and Yanwu Yang. 2025. [A survey on knowledge graph-based click-through rate prediction](#). *Expert Systems with Applications*, 281:127501.
- Ying Jin, Yanwu Yang, and Baojun Ma. 2026. [Dynamic heterogeneous graph convolutional networks for click-through rate prediction in recommender systems](#). *Electronic Commerce Research*, 26:2211–2240.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.
- Ernests Lavrinovics, Russa Biswas, Johannes Bjerva, and Katja Hose. 2025. [Knowledge graphs, large language models, and hallucinations: An nlp perspective](#). *Journal of Web Semantics*, 85:100844.
- Xiangyang Li, Bo Chen, Huifeng Guo, Jingjie Li, Chenxu Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, Jinxing Liu, Zhenhua Dong, and Ruiming Tang. 2022. [IntTower: The next generation of two-tower model for pre-ranking system](#). In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3292–3301.
- 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*, pages 1754–1763.
- 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 Web Conference 2024*, pages 3319–3330.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. [Learning entity and relation embeddings for knowledge graph completion](#). In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187.
- Ran Ma, Xiaotian Yang, Jiang Li, and Fei Gao. 2023. [Bayesian personalized ranking based on knowledge graph](#). In *Proceedings of the 2022 11th International Conference on Computing and Pattern Recognition*, pages 545–550.
- Kelong Mao, Jieming Zhu, Liangcai Su, Guohao Cai, Yuru Li, and Zhenhua Dong. 2023. [FinalMLP: An enhanced two-stream mlp model for ctr prediction](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4552–4560.
- Aashiq Muhamed, Iman Keivanloo, Sujana 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 *Proceedings of the NeurIPS Efficient Natural Language and Speech Processing (Models, Training, and Inference) Workshop*.
- Aashiq Muhamed, Jaspreet Singh, Shuai Zheng, Iman Keivanloo, Sujana Perera, James Mracek, Yi Xu, Qingjun Cui, Santosh Rajagopalan, Belinda Zeng, and Trishul Chilimbi. 2022. [DCAF-BERT: A distilled cachable adaptable factorized model for improved ads ctr prediction](#). In *Companion Proceedings of the Web Conference 2022*, pages 110–115.
- 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)*, pages 188–197.
- Ziheng Ni, Congcong Liu, Yuying Chen, Zhiwei Fang, Changping Peng, Zhongang Lin, Ching Law, and Jingping Shao. 2026. [iFusion: Integrating dynamic interest streams via diffusion model for click-through rate prediction](#). In *The Fourteenth International Conference on Learning Representations*.
- Kesha Ou, Zhen Tian, Wayne Xin Zhao, Hongyu Lu, and Ji-Rong Wen. 2026. [GenCI: Generative modeling of user interest shift via cohort-based intent learning for ctr prediction](#). In *Proceedings of the ACM Web Conference 2026*, pages 6709–6718.
- Junwei Pan, Jian Xu, Alfonso Lobos Ruiz, Wenliang Zhao, Shengjun Pan, Yu Sun, and Quan Lu. 2018. [Field-weighted factorization machines for click-through rate prediction in display advertising](#). In *Proceedings of the 2018 World Wide Web Conference*, pages 1349–1357.
- Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020.

- Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2685–2692.
- 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):1872–1897.
- Yanru Qu, Ting Bai, Weinan Zhang, Jianyun Nie, and Jian Tang. 2019. [An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation](#). In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*.
- 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)*, pages 1149–1154.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#).
- Steffen Rendle. 2010. [Factorization machines](#). In *2010 IEEE International Conference on Data Mining (ICDM)*, pages 995–1000.
- Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. [Predicting clicks: estimating the click-through rate for new ads](#). In *Proceedings of the 16th International Conference on World Wide Web*, pages 521–530.
- Muhammad Salman, Armin Haller, Sergio J. Rodriguez Mendez, and Usman Naseem. 2024. [Tiny but mighty: A crowdsourced benchmark dataset for triple extraction from unstructured text](#). In *Proceedings of the 20th Joint ACL - ISO Workshop on Interoperable Semantic Annotation @ LREC-COLING 2024*, pages 71–81.
- 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*, pages 1161–1170.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. [Graph attention networks](#). In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*.
- Dong Wang, Kavé Salamatian, Yunqing Xia, Weiwei Deng, and Qi Zhang. 2023. [BERT4CTR: An efficient framework to combine pre-trained language model with non-textual features for ctr prediction](#). In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5039–5050.
- 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 *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 94–104.
- Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. [DKN: Deep knowledge-aware network for news recommendation](#). In *Proceedings of the 2018 World Wide Web Conference*, pages 1835–1844.
- Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. [Deep & Cross network for ad click predictions](#). In *Proceedings of the ADKDD'17*.
- Xinyi Wang, Min Gao, Zhenni Lu, Zongwei Wang, Junwei Zhang, and Yi Zhang. 2019. [DMCM: A deep multi-channel model for dynamic movie recommendation](#). In *Neural Information Processing*, pages 425–432.
- Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 2020. [CKAN: Collaborative knowledge-aware attentive network for recommender systems](#). In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 219–228.
- Chuyuan Wei, Anning He, Shengda Zhuo, Changdong Wang, and Shuqiang Huang. 2026. [ProGraph: Graph prompt tuning with knowledge-aware contrastive learning for recommendation](#). *ACM Transactions on Multimedia Computing, Communications and Applications*, 22(4).
- Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. [Sheared llama: Accelerating language model pre-training via structured pruning](#). *Preprint*, arXiv:2310.06694.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Yanwu Yang and Panyu Zhai. 2022. [Click-through rate prediction in online advertising: A literature review](#). *Information Processing & Management*, 59(2):102853.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [XLNet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32.
- Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. [Deep interest evolution network for click-through rate prediction](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5941–5948.

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*, pages 1059–1068.

Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. [Open benchmarking for click-through rate prediction](#). In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2759–2769.

Ding Zou, Wei Wei, Xian-Ling Mao, Ziyang Wang, Minghui Qiu, Feida Zhu, and Xin Cao. 2022. [Multi-level cross-view contrastive learning for knowledge-aware recommender system](#). In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1358–1368.

A Appendix

A.1 Prompt Template Design

This section designs task-specific prompt templates covering two core modules: KG triples extraction, and KG triples validation.

Table 10: Baseline Prompt for KG Triples Generation

Prompt Content
Input Text: {text_chunk}
Task: Extract KG triples as a JSON array.
Format: [{"head", "relation", "tail"}]
Output: Only output the JSON array directly.

Prompt Template Design for KG Triples Generation. For KG triples extraction, two baseline templates, i.e., baseline prompt and structured constrained zero-shot prompt, are designed for comparison purposes, while the structured constrained few-shot prompt is the one that we actually adopt (Table 10, Table 11 and Table 14).

Prompt Template Design for KG Triples Validation. To ensure the factual accuracy and reliability of extracted KG triples, we adopt a dedicated validation prompt template to filter invalid triples from over-inference, semantic deviation and ungrounded generation. This mechanism checks textual grounding, relational consistency and entity formatting, and gives validation results with explicit reasons. The template is presented in Table 15.

Table 11: Structured Constrained Zero-Shot Prompt for KG Triples Generation

Prompt Content
Input Text: {text_chunk}
Task: Extract knowledge graph triples from the text as a JSON array.
Output Rules:
1. Output only JSON array format: [{"head": "...", "relation": "...", "tail": "..."}]
2. Empty array [] if no triples.
Structured Data Relations:
User: has_username, belongs_to_scenario.
Product: has_price, has_brand, has_title, belongs_to_category, has_style, has_rank.
Click: click_history_id
(train ONLY, valid/test FORBIDDEN).
Review Text Relations:
expresses_sentiment:
user→sentiment→product;
compares_to: product→other_product/category;
suggests_preference: user→preference_type.
Important Constraint: Review texts are only available in training set. Do NOT generate triples using these review-text-related relations for validation and test sets.
Entity Rules:
User ID: user_xxx (from text);
Product ID: product_xxx (from text);
Category: category_xxx (inferred from title);
Scenario: original name (e.g., Amazon Fashion);
Sentiment: positive/negative/neutral;
Preference: value_focused/quality_focused/etc.

Table 12: Input modality ablation results (AUC) across all datasets.

Modality	All Beauty	Amazon Fashion	Digital Music	Gift Cards	Musical Instruments
Text-KG	0.7769	0.8102	0.8309	0.8261	0.7886
w/o KG (only Text)	0.7411	0.7305	0.7878	0.7834	0.7240
w/o Text (only KG)	0.6624	0.7470	0.7955	0.7808	0.7545
only ID	0.5985	0.6518	0.6340	0.5837	0.6511

Table 13: Component ablation results (AUC) across all datasets.

Models	All Beauty	Amazon Fashion	Digital Music	Gift Cards	Musical Instruments
TKE4CTR	0.7769	0.8157	0.8309	0.8316	0.7886
w/o Cross-Attn Gate	0.7593	0.8021	0.8228	0.8161	0.7801
w/o Feat-Gated Head	0.7585	0.8082	0.8261	0.8212	0.7853

A.2 Graph Embedding with GAT

This module generates embeddings of knowledge subgraphs $g_{u,p}$ using a GAT that dynamically weights neighbor importance, with detailed implementation as follows:

Node Embedding Initialization. Core nodes $\mathcal{V}_{\text{core}}$ of a subgraph are prediction-relevant entities in $g_{u,p}$, grounded in generated text chunks and instantiated as product-type nodes for CTR prediction. Each node $v \in g_{u,p}$ is initially represented by concatenating heterogeneous features and applying L2 normalization, as referenced in Eq. (6) in the main text.

In detail, $\mathbf{x}_v^{\text{type}}$ is the node type embedding (e.g., user, product, attribute), $\mathbf{x}_v^{\text{id}}$ is a hash-based embedding ensuring entity uniqueness, and $\mathbf{x}_v^{\text{struct}}$ captures subgraph structure (e.g., degree, in/out connectivity and local clustering). $\mathbf{x}_v^{\text{bert}}$, which is from pretrained BERT (Devlin et al., 2019), applies only to core nodes via $\mathbb{I}[v \in \mathcal{V}_{\text{core}}]$, anchoring text semantics to prediction-relevant entities while avoiding auxiliary node noise.

GAT Encoding Details. The GAT consists of two layers, with detailed implementation supplementary to the main text: 1. First GAT Layer (8 parallel attention heads): For each node i , the k -th attention head computes the linear transformation $\mathbf{z}_{i,k}^{(\ell)} = \mathbf{W}_k \mathbf{h}_i^{(\ell)}$, where $\mathbf{W}_k \in \mathbb{R}^{d' \times d}$ is the learnable linear transformation matrix (input dimension d and output dimension d'). The attention coefficient α_{vu}^k shown in Eq. (7) quantifies the contribution of neighbor u to the representation update of node v , calculated via LeakyReLU activation and softmax normalization.

The output of the k -th head is a weighted sum of

transformed neighbor embeddings with nonlinear activation: $\mathbf{h}_v^{k,(\ell+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu}^k \mathbf{W}_k \mathbf{h}_u^{(\ell)}\right)$. The final output of the first layer is the concatenation of the results of all 8 attention heads, i.e., $\mathbf{h}_v^{(\ell+1)} = \parallel_{k=1}^8 \mathbf{h}_v^{k,(\ell+1)}$.

2. Second GAT Layer (single attention head):

This layer refines node embeddings using mean aggregation of neighbor features, with the learnable linear transformation matrix $\mathbf{W}$. The output is computed as $\mathbf{h}_v^{(\ell+2)} = \sigma\left(\frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} \mathbf{W} \mathbf{h}_u^{(\ell+1)}\right)$.

Embedding Finalization. The subgraph-level representation $\mathbf{h}_G$ serves as the final embedding of the core node, as defined in Eq. (2) in the main text. The GAT encoder is optimized on training subgraphs via CTR labels, and the learned parameters are directly transferred to the validation and test subgraphs for embedding generation and prediction.

A.3 Datasets

In classical CTR benchmark datasets (e.g., Criteo (Zhu et al., 2021) and Avazu (Zhu et al., 2021)), features are typically anonymized during preprocessing. Raw business-relevant attributes are transformed into opaque ID-based encodings, which obscure inherent semantic relationships. Since TKE4CTR critically relies on both semantic and structural information extracted from text and knowledge graphs, such ID-centric representations are not suitable for our approach.

We therefore adopt the Amazon Review Data (2018) (Ni et al., 2019) for our experiments. The dataset spans 12 years of Amazon user-item interactions (1996–2008) and contains non-anonymized features that make it feasible to produce the necessary semantic and structural information required

by TKE4CTR. We select five product categories as separate domains, as listed in Table 1.

To simulate real-world dynamic prediction scenarios and avoid time-related data leakage, the dataset is partitioned by timestamp, i.e., all interactions are sorted in a chronological order, with the first 80% as the training set, the middle 10% as the validation set, and the last 10% as the test set, in which samples with ratings above 3 are treated as positive and the rest as negative.

The training set covers interactions from earlier time sequences. We adopt static basic attributes (e.g., Domain, user_id, asin, price, brand and title), the target user’s click history features (e.g., user_hist_id and user_hist_title), and review text to construct LLM-derived semantic triples for representation enhancement. This enhancement method, used only during training, injects fine-grained user preferences and item semantic knowledge extracted from reviews and user click history, helping the model learn richer semantic patterns, without introducing post-click interaction signals during evaluation. In the validation and test sets, only the basic static attributes are kept, while interaction features on review text and user click history are completely removed. Knowledge-graph-based text trunks are fully built from user and product-related basic information, avoiding leakage of post-click interaction information during evaluation.

A.4 Ablation Results

This section supplements the ablation results of input modalities and model components across all datasets.

Input Modality Ablation Results. We further conduct input modality ablation across all five datasets, as summarized in Table 12. All domains share consistent performance trends, i.e., removing either the KG or text modality brings a noticeable performance decline, and excluding both modalities results in the lowest performance with only ID-based modeling. The results validate that both KG and text modalities are indispensable for CTR prediction.

Model Component Ablation Results. We also perform ablation on core model components over all five datasets, as listed in Table 13. The performance trends derived from the two representative datasets generalize well to all domains. Removing the cross-attention gated fusion module or the feature-gated head consistently decreases AUC

scores, which verifies the efficacy and generalizability of the two proposed components.

A.5 Metric Definitions for KG Triples Evaluation

This section presents the formal definitions of the intrinsic evaluation metrics adopted in the main text for KG triple extraction. We employ five comprehensive metrics to systematically assess the quality of extracted KGs.

To formalize the computation of these metrics, we first introduce the necessary notations. Let $\mathcal{T}$ denote the set of valid generated triples after filtering empty entries, with $N = |\mathcal{T}|$. For each sample i , let E_i be the set of expected relations and C_i be the set of relations covered by model outputs. Let $\mathcal{R}$ represent the set of canonical relation labels, and m_r denote the number of distinct surface forms corresponding to canonical relation r .

Evidence Support (ES) Evidence Support evaluates whether each generated triple is strictly grounded in the source text:

$$s_{\text{ES}} = \frac{1}{N} \sum_{t \in \mathcal{T}} a(t), \quad (22)$$

where $a(t) = 1$ if triple t obtains explicit textual support, and 0 otherwise.

Relation Validity (RV) Relation Validity quantifies the compliance of extracted triples with predefined relational schema and type constraints:

$$s_{\text{RV}} = \frac{1}{N} \sum_{t \in \mathcal{T}} v(t), \quad (23)$$

where $v(t) = 1$ if triple t is structurally and schematically valid, and 0 otherwise.

Information Coverage (IC) Information Coverage measures the proportion of key semantic relations successfully captured by the model:

$$s_{\text{IC}} = \frac{\sum_i |E_i \cap C_i|}{\sum_i |E_i|}. \quad (24)$$

Noise Control (NC) Noise Control assesses the ability to avoid meaningless placeholders and noisy tail entities:

$$s_{\text{NC}} = \frac{1}{N} \sum_{t \in \mathcal{T}} b(t), \quad (25)$$

where $b(t) = 1$ if the tail value is semantically valid and concise, and 0 otherwise.

Table 14: Structured Constrained Few-Shot Prompt for KG Triples Generation

Prompt Content
All rules, relations, and entity formats follow Table 11.
Few-shot Example (KG Construction Workflow):
<i>Input Text:</i> The user is Julie (ID: user_83), who clicked IDs product_22, clicked titles "3-Pack Sleep N Play (Terry)". The ID of the current product is product_68, the title is "5-pack bodysuits", the brand is "Belocia", the price is "\$27.00", it has similar items product_93. The review text is "I don't see the difference between these bodysuits and the more expensive ones."
<i>Expected Output:</i>
[
{"head": "user_83", "relation": "has_username", "tail": "Julie"},
{"head": "user_83", "relation": "click_history_id", "tail": "product_22"},
{"head": "user_83", "relation": "click_history_title", "tail": "3-Pack Sleep N Play (Terry)"},
{"head": "user_83", "relation": "expresses_sentiment", "tail": "neutral"},
{"head": "user_83", "relation": "suggests_preference", "tail": "value_focused"},
{"head": "product_68", "relation": "has_title", "tail": "5-pack bodysuits"},
{"head": "product_68", "relation": "has_price", "tail": "\$27.00"},
{"head": "product_68", "relation": "has_brand", "tail": "Belocia"},
{"head": "product_68", "relation": "has_similar_item", "tail": "product_93"},
{"head": "product_68", "relation": "compares_to", "tail": "premium_bodysuits"}
]

Relation Consistency (RC) Relation Consistency evaluates the naming stability of semantically equivalent relational expressions:

$$s_{RC} = 1 - \frac{\sum_{r \in \mathcal{R}} \max(0, m_r - 1)}{\max(|\mathcal{R}|, 1)}. \quad (26)$$

A.6 Definitions of Hallucination Evaluation Metrics

This section formally defines the metrics for evaluating hallucination in KG triple extraction, with clear calculation standards to ensure the rigor of the evaluation. Let $\mathcal{T}$ denote the set of all extracted KG triples; $\mathcal{T}_{NLI}$ represents the subset of KG triples that are semantically consistent with the source text; $\mathcal{T}_{Rule}$ denotes the subset of KG triples complying with predefined constraints; and $\mathcal{T}_{Hallu}$ stands for the subset of invalid, hallucinated KG triples that fail verification.

NLI Factual Fidelity (NLI-F) This metric measures the proportion of extracted KG triples that are semantically consistent with the source text, verified by Natural Language Inference (NLI) tech-

niques. Its calculation formula is:

$$NLI-F = \frac{|\mathcal{T}_{NLI}|}{|\mathcal{T}|}, \quad (27)$$

where a higher NLI-F value indicates that more extracted KG triples have reliable textual support and lower factual deviation.

Rule-based Fidelity (R-F) R-F evaluates the compliance of extracted KG triples with predefined schema and entity constraints. It is calculated as the ratio of schema-compliant KG triples to the total number of extracted triples:

$$R-F = \frac{|\mathcal{T}_{Rule}|}{|\mathcal{T}|}, \quad (28)$$

where a higher R-F value reflects that the extracted KG triples better conform to the preset construction norms of KG.

Hallucination Rate (HR) HR quantifies the overall hallucination degree of the model in KG triple extraction, calculated as the proportion of invalid and hallucinated KG triples:

$$HR = \frac{|\mathcal{T}_{Hallu}|}{|\mathcal{T}|}, \quad (29)$$

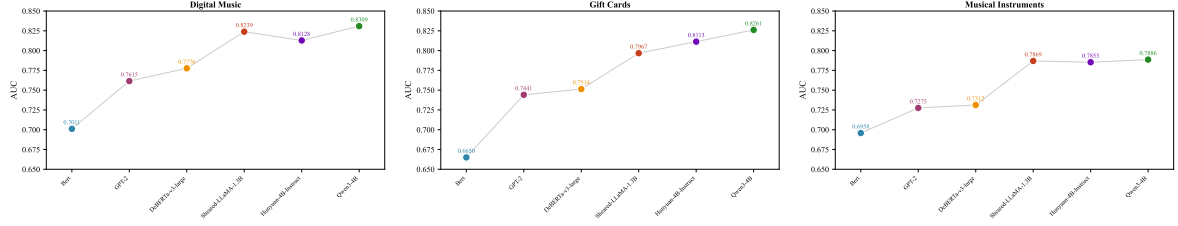


Figure 6: Performance comparison of baseline LLMs on three additional datasets. From left to right: Digital Music, Gift Cards, and Musical Instruments.

Table 15: Prompt for KG Triples Validation

Prompt Content
Task: Strictly validate the factual correctness of knowledge graph triples grounded only in the given source text. Reject over-inference, semantic deviation and ungrounded extra information. Input: Source Text: {source_text} Triple to Validate:(head,relation,tail) Validation Rules: 1. Check whether all abstract entities such as user_xxx and product_xxx exactly appear in the source text. 2. Verify whether the relation strictly belongs to the predefined allowed relation list. 3. The triple must be explicitly stated or strictly entailed by the source text. 4. Any triple with over-inference, unmentioned extra facts or semantic inconsistency should be marked invalid. Output: Only return JSON object in format {"valid": true/false, "reason": "explanation", "confidence": 0.0-1.0}.

where a lower HR value means that the model has better control over hallucination and generates more valid KG triples.

A.7 Performance Comparison of LLMs Across All Datasets

Table 16 lists the complete AUC of all six baseline models on all the datasets. Figure 6 displays the experimental trends on the remaining three datasets including Digital Music, Gift Cards and Musical Instruments. The ranking order of different models remains stable across diverse data domains, indicating the validity of our core experimental conclusions.

Experimental performance differs slightly in

various application scenarios. Taking actual results as examples, Qwen3-4B exceeds Sheared-LLaMA-1.3B by 0.0070 on Digital Music, and outperforms Hunyuan-4B-Instruct by 0.0148 on Gift Cards. These differences are mainly caused by distinct text styles and semantic difficulty in different fields, while the relative performance order of models does not change obviously. Combined with previous experimental observations, we can confirm that having more model parameters does not always bring better recommendation performance. Model structure design and training schemes have more obvious effects on practical prediction results.

A.8 Attention Head Number Analysis Across All Datasets

To further verify how attention head quantity affects cross-attention gated fusion in CTR prediction, we supplement experimental results on all five datasets in this section. We only present results on All Beauty and Amazon Fashion in the main text, and complete results on AUC and LogLoss under different head settings are listed in Table 17.

Experimental trends remain stable across all datasets. Model performance steadily rises when the number of attention heads is increased from 2 to 8, indicating that more attention heads help strengthen cross-modal interaction and feature fusion. The performance gets slightly declined when the head number is adjusted to 16. Extra attention branches bring redundant parameters and fail to boost prediction ability. The configuration with 8 attention heads maintains satisfactory prediction accuracy while controlling computational cost, supporting the hyperparameter selection in our main experiments.

A.9 Case Studies

We conduct concrete case studies to intuitively illustrate the working process of our dual-prompt driven modality transformation approach, and vali-

Table 16: Performance comparison of different LLMs (AUC) across all datasets.

Model	All Beauty	Amazon Fashion	Digital Music	Gift Cards	Mus. Instruments
BERT	0.6755	0.7095	0.7011	0.6650	0.6958
GPT-2	0.6702	0.7511	0.7615	0.7441	0.7275
DeBERTa-v3-large	0.6955	0.7516	0.7776	0.7514	0.7312
Sheared-LLaMA-1.3B	0.7530	0.7580	0.8239	0.7967	0.7869
Hunyuan-4B-Instruct	0.7566	0.7734	0.8128	0.8113	0.7853
Qwen3-4B	0.7616	0.8102	0.8309	0.8261	0.7886

Table 17: Performance analysis of attention head number (AUC / LogLoss) across all datasets.

Num Heads	All Beauty		Amazon Fashion		Digital Music		Gift Cards		Mus. Instruments	
	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss
2	0.7487	0.5605	0.7820	0.4910	0.8083	0.4120	0.8002	0.4305	0.7648	0.4702
4	0.7588	0.5470	0.8055	0.4400	0.8227	0.3951	0.8146	0.4183	0.7779	0.4601
8	0.7616	0.5388	0.8102	0.4294	0.8309	0.3871	0.8261	0.4102	0.7886	0.4513
16	0.7610	0.5387	0.8100	0.4285	0.8311	0.3866	0.8253	0.4115	0.7880	0.4502

date its practical capability across different application domains. The complete three-stage processing pipeline is fully described in Section 4.2.

Case 1: Amazon Fashion Domain As shown in Figure 2, this case is implemented on fashion-oriented user, product and review data. The used data includes user interaction logs, fixed product attributes and user review texts. We divide the original data into user content, product attribute content and unstructured review content, and apply designed prompts to extract structured relational triples. These triples cover both objective attribute information and implicit user preference information. Several typical examples are listed as follows:

(user_83, “has_username”, “Belocia”)
 (product_68, “has_price”, “\$27.00”)
 (user_83, “suggests_preference”, “value_focused”)
 (product_68, “compares_to”, “premium_bodysuits”)

All extracted triples are merged to construct domain knowledge structures, which can systematically arrange basic attribute facts and hidden preference information in raw data.

Case 2: Musical Instruments Domain We adopt musical instrument domain data as another typical example with different domain traits. Several typical basic attribute triples extracted from structured data are shown as follows:

(product_906, “has_title”, “Acid Music 1.0”)
 (product_906, “has_rank”, “58,496 in Software”)

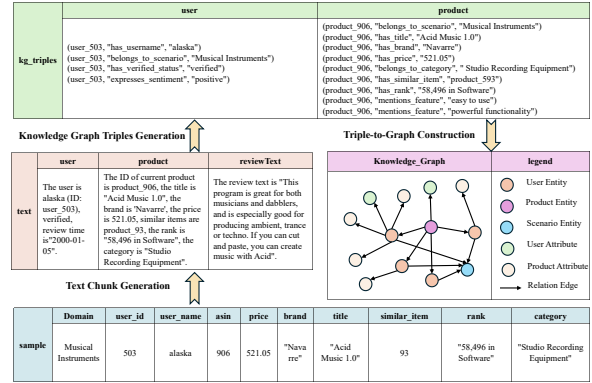


Figure 7: Case2: Dual-Prompt Driven Modality Transformation example on Musical Instruments domain data

Several typical semantic and sentiment triples mined from user reviews are listed below:

(product_906, “mentions_feature”, “easy to use”)
 (user_503, “expresses_sentiment”, “positive”)

It can be seen that our method can stably convert scattered text information into standardized triple forms, and unify heterogeneous field information into structured knowledge representation.