A Novel Matching Paradigm: Unified Generative and Discriminative LLM with Prompt Compression for Relevance Learning

Guoliang Zhao[1], Zixin Cui[2], Chao Ye[2]*, Dengwu He[2], Fei Huang[2], Yubo Liu[2], Shuanglong Li[2], Tzungren Kuo[2], Bin Ding[2], Shuang Zhang[2], Kunhong Zhu[2], Zhi Guo[2], Lin Liu[2]

[1]Institute of Automation Science and Engineering, Xi'an Jiaotong University
[2]Baidu Search Ads, Baidu Inc.

Abstract

The matching paradigm is fundamental to large-scale information retrieval and is widely used in industrial search and advertising systems. Existing approaches employ Large Language Models (LLMs) primarily as feature extractors, underutilizing their full modeling capabilities. To address this limitation, we propose a novel matching paradigm, termed the Unified Generative and Discriminative LLM (UGD). It integrates two-tower, single-tower, and generative tasks within a unified LLM framework via attention-mask partitioning, enabling generative tasks to serve as auxiliary supervision for discriminative learning and facilitating distillation from single-tower to two-tower architectures through a multi-task fine-tuning mechanism. To satisfy online latency constraints, we further introduce a self-distillation variant of UGD with a KMeans-enhanced linearized RQ-VAE for prompt compression and quantization. This design compresses and quantizes landing-page documents during inference, improving serving efficiency and reducing storage overhead. Extensive experiments show that UGD achieves superior performance and strong practical value. The framework has been deployed in an industrial search engine serving hundreds of millions of users and hundreds of thousands of advertisers, significantly enhancing search experience. Open access upon publication.

1 Introduction

With the rapid growth of information on the Internet, search and recommendation systems have become essential for meeting users' information needs (Kobayashi and Takeda, 2000; Adomavicius and Tuzhilin, 2005). Although their objectives differ, they can be unified as a matching problem from a technical perspective (Garcia-Molina et al., 2011). In industrial systems, matching models must achieve both high recall accuracy and low-latency retrieval (Su et al., 2023). Two-tower and single-tower architectures have proven effective for large-scale matching tasks (Huang et al., 2020; Yang et al., 2020; Yu et al., 2021; Jang et al., 2023), and are widely used in the coarse- and fine-ranking stages of search and advertising systems.

The two-tower model uses separate encoders for queries and documents, computing relevance scores via a simple fully connected layer or dot product (Reimers, 2019; Huang et al., 2013). Its main limitation lies in low matching accuracy due to limited feature interaction. The single-tower model allows bottom-layer interactions between query and document but still lacks interpretability (Lu et al., 2022; Devlin, 2018; Liu, 2019), providing only a correlation score without explaining the underlying reasons. In applications such as advertising, e-commerce, and search, understanding the rationale behind relevance scores is crucial. For instance, in search advertising, advertisers need not only the correlation between user queries and landing pages (LP) but also insights into why the correlation exists. This information enables continuous optimization of products and landing pages, improving competitiveness and user experience. Consequently, innovating traditional matching paradigms with generative LLMs (Du et al., 2021; Yang et al., 2025; Liu et al., 2024; Team et al., 2025) is of particular importance. However, applying LLMs introduces corresponding challenges for online inference, especially due to the substantially increased model size and the excessive length of documents.

To address these challenges, we propose the **Unified Generative and Discriminative LLM (UGD)**, a novel matching paradigm that integrates two-tower, single-tower, and generative tasks within a single LLM framework via attention-mask partitioning. Using a multi-task fine-tuning mechanism, UGD leverages generative tasks to enhance discriminative learning and distills single-tower knowledge into two-tower architectures. To meet online

*corresponding author

latency requirements, we further develop a **self-distillation variant with a KMeans-enhanced linearized RQVAE** for prompt compression and quantization. Attention mask isolation enables self-distillation from the full document prompt to a compressed prompt, and the RQVAE (Lee et al., 2022) quantizes the compressed vectors, improving serving efficiency and reducing storage overhead. In summary, our contributions are as follows:

- **UGD** is proposed integrating two- and single-tower and generative tasks via attention-mask partitioning to enhance discriminative learning while enabling single-to-two-tower distillation to improve two-tower performance.
- **Self-distillation with KMeans-enhanced linearized RQVAE** is developed compressing and quantizing prompts to support efficient online inference and reduce storage overhead.
- **Query Landing Page Quality (QLQ) dataset** is provided from an industrial search advertising engine. Experiments show that UGD achieves superior performance and strong practical value.

2 Method

In this section, we introduce our unified generative and discriminative matching paradigm (UGD) and further develop a self-distillation variant with a KMeans-enhanced linearized RQVAE for prompt compression and quantization.

2.1 UGD Matching Paradigm

Attention mask Partition. Distinguished from traditional matching paradigms, our method employs customized CoT prompts and partitioned attention masks as inputs. As shown in Figure 1(a), query-document pairs along with correlation discrimination reasons are fed into the generative LLM through CoT prompt engineering. Specifically, the input is structured as "query + [CLS] + document + [CLS] + because [gMask], the correlation score is [CLS] + [Start] + reason + [EOS]", where the first two [CLS] tokens represent the feature embeddings of the query and document. To ensure the independence of the query and document in the two-tower model, an attention mask is applied so that each can only interact with itself.

To enhance interpretability and logical consistency, a discriminative reason generative task is introduced alongside the correlation discrimination task, with [gMask] denoting the reason to be generated. Its label is '[Start] + reason + [EOS]', and the corresponding attention mask is a lower-triangular matrix. For knowledge distillation from single-tower to two-tower discrimination, an additional [CLS] token is added after the reason token to represent the single-tower discriminative feature, which has access to the query, document, and reason tokens. Through this design, the two-tower UGD paradigm is realized via customized prompts and attention mask partitioning. As shown in Figure 1(b), the single-tower UGD follows a similar design.

Multi-Task Learning. The architecture of the two-tower UGD model is illustrated on the left side of Figure 1(a). It consists of three task-specific head networks for the two-tower discriminative task, the single-tower discriminative task, and the generative task. The vectors corresponding to the first two special tokens ([CLS]) in the encoder output F are extracted as the query and document representations, denoted as V_Q and V_{Doc}. After concatenation, the representation is projected by FC_t to obtain F_t, which is further processed by FC and FC_t^{kl} to produce $Logits_t$ and $\hat{F}_t$. Similarly, the single-tower representation V_s is processed by FC and FC_s^{kl} to obtain $Logits_s$ and $\hat{F}_s$, which impose KL-based constraints on $Logits_t$ and $\hat{F}_t$ for knowledge distillation. For the generative task, an LMHead maps the encoder output F from the embedding dimension D to the vocabulary size C_{vocab}. Specifically, the multi-task loss functions are defined as follows:

$$\mathcal{L}_{cls}^{t} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[-\sum_{i=1}^{C} y_i \log(p_i^t)\right] \quad (1)$$

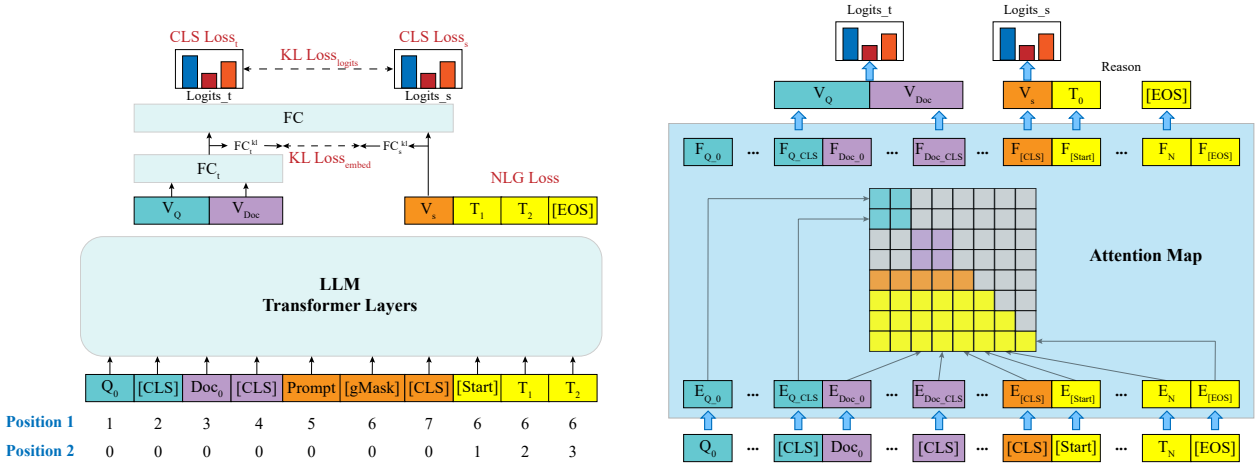
$$\mathcal{L}_{cls}^{s} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[-\sum_{i=1}^{C} y_i \log(p_i^s)\right] \quad (2)$$

$$\mathcal{L}_{gen} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[-\frac{1}{T}\sum_{t=1}^{T} \log q(y_t|y_{<t})\right] \quad (3)$$

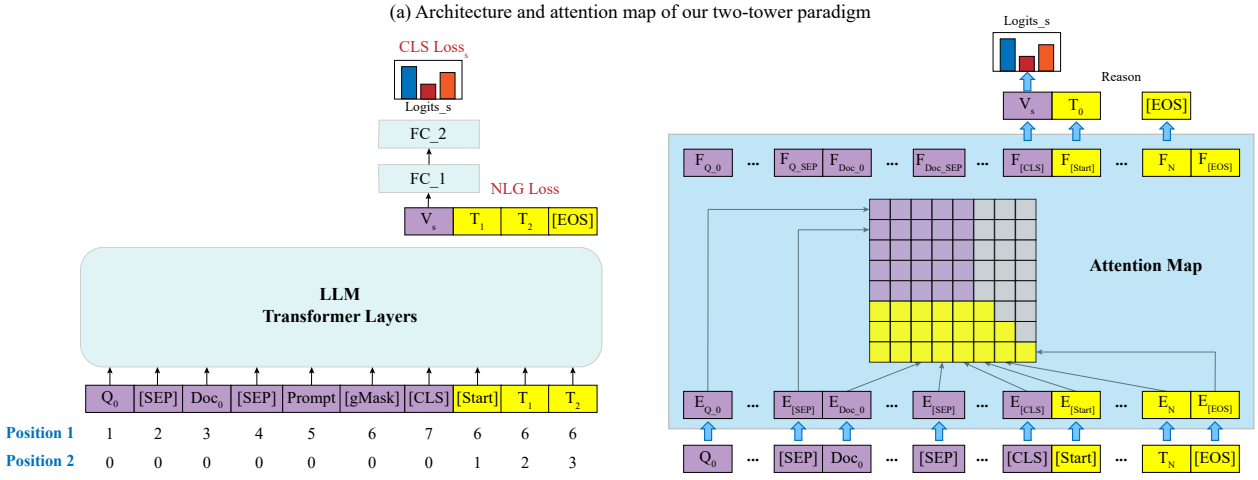
$$\mathcal{L}_{kl} = \sum_{x\in X} P(x) \log\left(\frac{P(x)}{Q(x).detach}\right) \quad (4)$$

$$\mathcal{L}_{kl}^{emb} = \sum_{x\in X} T(x) \log\left(\frac{T(x)}{S(x).detach}\right) \quad (5)$$

$\mathcal{L}_{cls}^{t}$ and $\mathcal{L}_{cls}^{s}$ denote the two- and single-tower cross-entropy losses, $\mathcal{L}_{gen}$ the generative loss, and



(a) Architecture and attention map of our two-tower paradigm



(b) Architecture and attention map of our single-tower paradigm

Figure 1: An overview of the unified generative and discriminative two-tower matching paradigm. In the attention mask, the green and purple squares represent the attention mask of all-to-all interaction within query and document respectively. The orange square represents the unidirectional interaction between the single-tower [CLS] token and query and document. The yellow square indicates the attention mask of the generated task. Grey squares are masked out. [CLS] tokens represent the features of the discriminative task and [gMask] token indicates the content to be generated. 2D positional encoding represents inter- and intra-span positions.

$\mathcal{L}_{kl}$ and $\mathcal{L}_{kl}^{emb}$ the self-distillation KL losses. Therefore, the objective function of our two-tower UGD paradigm is as follows:

$$\mathcal{L}_t = \alpha\mathcal{L}_{cls}^t + \beta\mathcal{L}_{cls}^s + \gamma\mathcal{L}_{gen} + \lambda\mathcal{L}_{kl} + \mu\mathcal{L}_{kl}^{emb} \quad (6)$$

where $\{\alpha, \beta, \gamma, \lambda, \mu\}$ represents the weight coefficients of each loss function. Similarly, the objective function of the single-tower UGD paradigm is composed of $\mathcal{L}_{cls}^s$ and $\mathcal{L}_{gen}$ as follows:

$$\mathcal{L}_s = \beta\mathcal{L}_{cls}^s + \gamma\mathcal{L}_{gen} \quad (7)$$

2.2 Self-distillation Variant with a KMeans-enhanced Linearized RQVAE for Prompt Compression and Quantization

In search advertising engines, single-tower models for fine-grained ranking require online query–document interaction, which incurs substantial latency as LLM sizes and landing page context lengths increase. To mitigate this issue, we propose a self-distillation landing page prompt compression method, along with a KMeans-enhanced Linearized RQVAE for further prompt quantization. The proposed approach enables efficient inference for search advertising systems in the LLM era.

Self-distillation Prompt Compression. As shown in Figure 2, the input is structured as Query + LP Placeholder + [CLS] + LP + Prompt + [CLS] + Reason, and embedded via word embeddings. The LP Placeholder uses N special tokens. We introduce an encoder LLM to compress LP: LP and the N special tokens are encoded to produce N vectors representing the compressed LP, denoted E_{LP-h}, which replace the original LP Placeholder

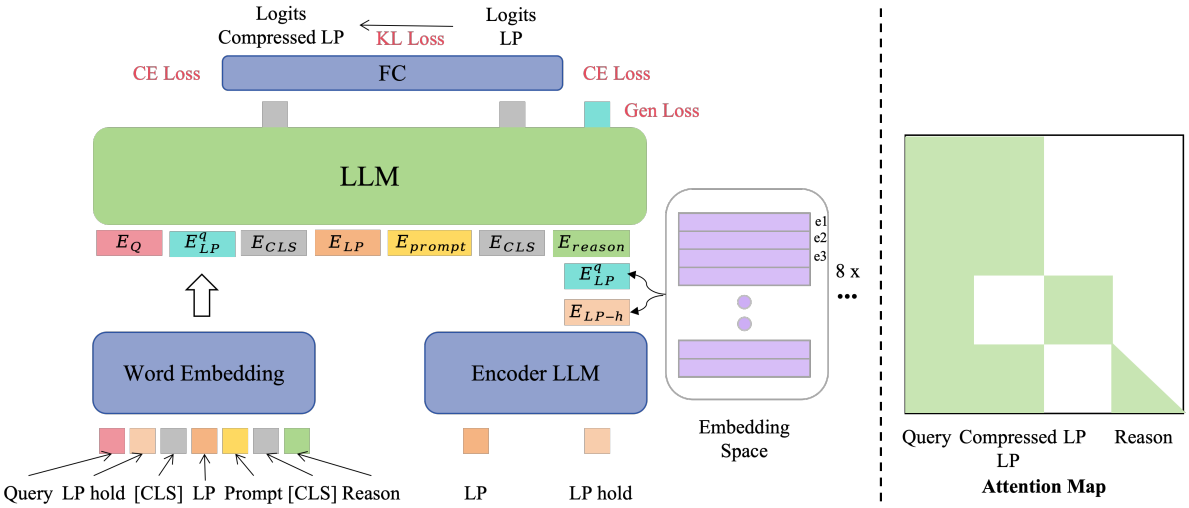


Figure 2: An overview of the self-distillation variant of UGD with a KMeans-enhanced linearized RQVAE for prompt compression and quantization.

embeddings.

Directly compressing the landing page often results in significant performance degradation. To address this, we propose a self-distillation-based prompt compression method. Specifically, as illustrated by the attention mask in Figure 2, the LP is introduced during training, and self-distillation from $\langle \text{Query}, \text{LP} \rangle$ to $\langle \text{Query}, \text{Compressed LP} \rangle$ is enforced via attention mask isolation and Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951). In detail, attention mask isolation ensures that $\langle \text{Query}, \text{LP} \rangle$ and $\langle \text{Query}, \text{Compressed LP} \rangle$ interact independently during training, making LP and Compressed LP mutually invisible. This guarantees that efficient inference can be achieved by inputting only $\langle \text{Query}, \text{Compressed LP} \rangle$. The loss functions are analogous to those used in UGD.

Quantization: KMeans-enhanced Linearized RQVAE. In search advertising engines, the LP collection is extremely large, often hundreds of millions, making direct offline storage of all Compressed LP vectors highly costly. To address this, we propose a KMeans-enhanced linearized RQVAE for quantizing the Compressed LP vectors.

Specifically, first, the RQVAE applies multi-stage residual quantization to the LP compression vectors obtained from the encoder LLM, reducing information loss at each stage. Second, the codebook space is linearized to enhance cohesion, avoid suboptimal local solutions, and improve codebook utilization. Finally, KMeans clustering is employed to initialize the codebook, aligning it with the distribution of LP compression vectors and mitigating biases from random initialization. As a result, each Compressed LP vector can be represented by storing only $8 \times N$ codebook indices.

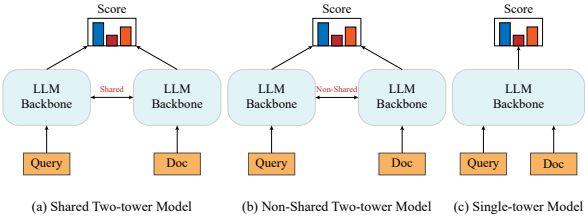


Figure 3: The architectures of baseline models.

3 Experiments

3.1 Setup

Datasets. We construct the **Query Landing Page Quality (QLQ)** dataset from an industrial search advertising engine. Each sample contains a query, landing page description, quality score (0–3), and corresponding reason. The training set contains 2741326 non-snapshot samples, and the test set 24227 snapshot samples. We also reconstructed several public text-matching datasets: **ATEC**, **BQ-Corpus** (Chen et al., 2018), **LCQMC** (Liu et al., 2018), **NLI** (Bowman et al., 2015), and **QQP** (Le et al., 2021). Details in Appendix A.

Baselines. As presented in Figure 3, we categorize traditional matching paradigms as: Shared Two-Tower Model (Shared-TTM) (Xi et al., 2021), Non-Shared Two-Tower Model (TTM) (Reimers, 2019; Khattab and Zaharia, 2020) and Single-Tower Model (STM) (Devlin et al., 2019).

Training Details. In our experiments, we pre-

Type	Methods	Chinese (ACC / AUC)				English (ACC / AUC)	
		QLQ	BQ	ATEC	LCQMC	NLI	QQP
TTM	Shared TTM	0.7183 / 0.9141	0.8170 / 0.8892	0.8139 / 0.6940	0.8215 / 0.9182	0.7646 / 0.9253	0.9223 / 0.9622
	TTM	0.7236 / 0.9137	0.7172 / 0.7849	0.8176 / 0.6297	0.7478 / 0.8459	0.7444 / 0.9130	0.8977 / 0.9397
	UGD TTM	**0.7311 / 0.9185**	**0.8448 / 0.9180**	**0.8284 / 0.8288**	**0.8331 / 0.9185**	**0.7887 / 0.9388**	**0.9300 / 0.9807**
STM	STM	0.7443 / 0.9343	**0.8614 / 0.9351**	0.8685 / 0.8966	0.8874 / 0.9616	0.8701 / 0.9744	0.9752 / **0.9905**
	UGD STM	**0.7525 / 0.9369**	0.8604 / 0.9342	**0.8728 / 0.9022**	**0.8908 / 0.9619**	**0.8746 / 0.9786**	**0.9803 / 0.9868**

Table 1: Performance comparisons. The best results are in bold.

Model (UGD)	Multi-Task Loss					Metric (ACC / AUC)
	$\mathcal{L}_{cls}^{t}$	$\mathcal{L}_{cls}^{s}$	$\mathcal{L}_{gen}$	$\mathcal{L}_{kl}$	$\mathcal{L}_{kl}^{emb}$	
TTM	✓	✓	✓	w/o	w/o	0.7225 / 0.9140
	✓	✓	✓	✓	✓	**0.7311 / 0.9185**
	✓	✓	✓	✓	×	0.7289 / 0.9185
	✓	✓	✓	×	×	0.7212 / 0.9129
	✓	✓	×	×	×	0.7149 / 0.9074
	✓	×	×	×	×	0.7158 / 0.9087
STM	-	✓	✓	-	-	**0.7525 / 0.9369**
	-	✓	×	-	-	0.7443 / 0.9343

Table 2: Ablation experiments on the QLQ dataset. We conduct ablation experiments for UGD TTM and UGD STM from the loss functions of $\mathcal{L}_{cls}^{t}$, $\mathcal{L}_{cls}^{s}$, $\mathcal{L}_{gen}$, $\mathcal{L}_{kl}$ and $\mathcal{L}_{kl}^{emb}$, as well as their corresponding head networks. w/o represents "KL Loss without stop gradient". The best results are in bold.

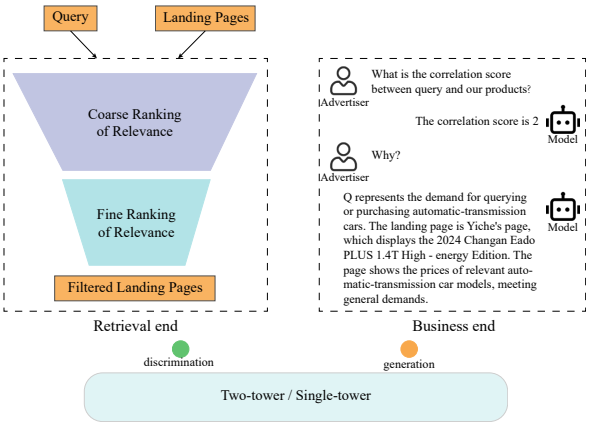


Figure 4: An illustration of the application of our UGD matching paradigm in search advertising.

trained a 1B-parameter LLM backbone using both general-domain and industrial search advertising corpora. Full-parameter fine-tuning was performed with a learning rate of $5e-6$, batch size of 32, and the AdamW optimizer, implemented on 8 NVIDIA A800 GPUs. Details in Appendix B.

3.2 Results and Online Testing for UGD

Main Results. As shown in Tables 1, our UGD matching paradigm has made significant improvements in ACC and AUC indicators compared with the traditional matching paradigm, which demonstrates that our UGD can better stimulate the capabilities of generative LLMs.

Specifically, UGD TTM significantly improves matching performance by integrating two-tower, single-tower, and generative tasks within a unified LLM framework, while leveraging knowledge distillation to enhance discriminative learning. On the industrial search advertising dataset QLQ, it achieves improvements of 0.75% in ACC and 0.48% in AUC, and gains of 0.77%–2.78% are observed on several public datasets. Similarly, UGD STM also yields consistent improvements, where enhanced query–document interaction introduced by generative tasks leads to stable performance gains across domains.

Beyond performance improvements, UGD also enables the model to generate discriminative reasons, providing interpretable explanations for correlation judgments in practical applications. Detailed case studies are presented in Appendix C.

Ablation Study. As shown in Tables 2, we conduct ablation experiments for UGD TTM and UGD STM around the loss functions of $\mathcal{L}_{cls}^{t}$, $\mathcal{L}_{cls}^{s}$, $\mathcal{L}_{gen}$, $\mathcal{L}_{kl}$ and $\mathcal{L}_{kl}^{emb}$.

Ablation studies on UGD TTM show that self-distillation from single-tower to two-tower discrimination is essential for performance gains, while removing distillation terms consistently degrades ACC. Meanwhile, removing the reason generative loss $\mathcal{L}_{gen}$ also reduces performance for both UGD TTM and UGD STM, demonstrating the effectiveness of generative supervision.

Online Testing. We deploy our UGD on the industrial search advertising engine and randomly take a small percentage of traffic as the test group. Figure 4 illustrates the application of our UGD in the search advertising domain. On the left side is the retrieval end, whose responsibility is to filter the landing pages corresponding to a specified

Table 3: Experimental results for prompt compression and quantization strategies based on QLQ dataset. Here, "a" denotes non-snapshot data, "b" denotes snapshot data, "ph" indicates placeholder special token compression, and "sd" denotes self-distillation from ⟨Query, LP⟩ to ⟨Query, Compressed LP⟩.

Train Data	Compression Strategy	Quantization Strategy	Token Nums	Metric (ACC / AUC)
a	-	-	600	0.7525 / 0.9369
a+b	-	-	600	**0.7855 / 0.9445**
a+b	ph	-	40	0.7607 / 0.9311
a+b	ph + sd	-	40	**0.7818 / 0.9406**
a+b	ph + sd	VQVAE	40	0.6234 / 0.8191
a+b	ph + sd	RQVAE (4)	40	0.7263 / 0.9125
a+b	ph + sd	linearized RQVAE (4)	40	0.7647 / 0.9280
a+b	ph + sd	linearized RQVAE (8)	40	0.7714 / 0.9320
a+b	ph + sd	linearized RQVAE (8) + Kmeans warm	40	**0.7798 / 0.9389**

query. On the right side is the business end, which is tasked with notifying advertisers of the relevance score and the reasons behind the score between their products and the query. At the retrieval end, the coarse- and fine-rank filtering of candidate landing pages is realized through the discrimination functions of UGD TTM and UGD STM. At the business end, advertisers can gain insights into the reasons for relevance discrimination to improve their products and optimize the landing page design. As a result, a virtuous feedback loop is established. Through continuous improvement based on the provided reasons, the quality of products and landing pages can be steadily enhanced, leading to better user experiences, advertising campaigns and higher income and revenue growth.

During the A/B testing period, which typically spans at least one week, we closely monitor the performance of our UGD. Compared with the previously deployed model, the UGD TTM effectively reduces the proportion of 0-score landing pages by **1.87%**. Meanwhile, the UGD STM achieves an even more substantial reduction for the proportion of 0-score landing pages by **3.2%**. Details are in the Appendix D.

3.3 Exploratory Results for Prompt Compression and Quantization

In this subsection, for UGD STM, we investigate the effects of our self-distillation prompt compression and the KMeans-enhanced linearized RQVAE quantization strategy. To this end, we further augmented the QLQ dataset by incorporating 213367 snapshot samples and employed a Conditional Fine-Tuning (CFT) training data strategy. Detailed descriptions are provided in Appendix A.

As shown in Table 3, the introduction of snapshot samples and the CFT strategy substantially enhances UGD STM performance (+3.3% ACC / +0.76% AUC). Building upon this, experiments with the prompt compression strategy indicate that self-distillation markedly improves compression effectiveness (+2.11% ACC / +0.95% AUC), resulting in performance comparable to the baseline.

We further investigate the quantization strategy. Notably, the initial VQVAE quantization results in significant performance degradation. By progressively optimizing the codebook space—RQVAE $\rightarrow$ linearized RQVAE (4 & 8) $\rightarrow$ linearized RQVAE (8) with KMeans warm-start—the performance loss due to quantization is substantially reduced (ACC: +10.29% $\rightarrow$ 4.51% $\rightarrow$ 0.84%). Compared to the baseline, the final performance experiences only a slight decline (-0.57% ACC / -0.56% AUC).

Under essentially comparable performance, the LP sequence length is compressed from 600 to 40 tokens, significantly improving token utilization, reducing online resource consumption, and accelerating inference. Additionally, each LP vector requires storing only 8×40 codebook indices, alleviating offline storage demands.

4 Conclusion

In this work, we presented UGD, a unified generative and discriminative LLM-based matching paradigm that integrates two-tower, single-tower, and generative tasks within a single framework. By leveraging attention-mask partitioning and multi-task fine-tuning, UGD enables generative tasks to provide auxiliary supervision for discriminative learning and facilitates knowledge distillation from single-tower to two-tower architectures. To meet online latency requirements, we further proposed a self-distillation variant with a KMeans-enhanced linearized RQVAE for prompt compression and quantization, enabling efficient inference and reduced storage overhead. Extensive experiments demonstrate the effectiveness and superiority of UGD along with the proposed compression and quantization strategies. Our methods have been deployed in an industrial search advertising engine, significantly enhancing search experience and user satisfaction, serving hundreds of millions of end users and supporting the operations of hundreds of thousands of advertisers.

Limitations

Our UGD framework with the proposed prompt compression and quantization methods, has demonstrated significant benefits in real-world search advertising systems. Nonetheless, several limitations remain. First, industrial systems primarily emphasize discriminative performance metrics, and despite insights gained from case studies, a systematic evaluation on production tasks is still lacking. Second, while prompt compression incurs minimal performance loss, quantization introduces slightly larger degradation, which warrants further optimization in future work. So far, the quantization strategy could be applied in industry with careful consideration of the specific deployment context.

References

Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The bq corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 4946–4951.

Lizhe Chen, Binjia Zhou, Yuyao Ge, Jiayi Chen, and Shiguang Ni. 2025. Pis: Linking importance sampling and attention mechanisms for efficient prompt compression. *arXiv preprint arXiv:2504.16574*.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*.

Hector Garcia-Molina, Georgia Koutrika, and Aditya Parameswaran. 2011. Information seeking: convergence of search, recommendations, and advertising. *Communications of the ACM*, 54(11):121–130.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2553–2561.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Jiho Jang, Chaerin Kong, Donghyeon Jeon, Seonhoon Kim, and Nojun Kwak. 2023. Unifying vision-language representation space with single-tower transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 980–988.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pages 5583–5594. PMLR.

Mei Kobayashi and Koichi Takeda. 2000. Information retrieval on the web. *ACM computing surveys (CSUR)*, 32(2):144–173.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Huong T Le, Dung T Cao, Trung H Bui, Long T Luong, and Huy Q Nguyen. 2021. Improve quora question pair dataset for question similarity task. In *2021 RIVF International Conference on Computing*

and Communication Technologies (RIVF), pages 1–5. IEEE.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11523–11532.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. 2025a. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7182–7195.

Zongqian Li, Yixuan Su, and Nigel Collier. 2025b. 500xcompressor: Generalized prompt compression for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25081–25091.

Zhenghao Lin, Yeyun Gong, Xiao Liu, Hang Zhang, Chen Lin, Anlei Dong, Jian Jiao, Jingwen Lu, Daxin Jiang, Rangan Majumder, and 1 others. 2023. Prod: Progressive distillation for dense retrieval. In *Proceedings of the ACM Web Conference 2023*, pages 3299–3308.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th international conference on computational linguistics*, pages 1952–1962.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Haoyu Lu, Nanyi Fei, Yuqi Huo, Yizhao Gao, Zhiwu Lu, and Ji-Rong Wen. 2022. Cots: Collaborative two-stream vision-language pre-training model for cross-modal retrieval. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 15692–15701.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 255–262.

Liangcai Su, Fan Yan, Jieming Zhu, Xi Xiao, Haoyi Duan, Zhou Zhao, Zhenhua Dong, and Ruiming Tang. 2023. Beyond two-tower matching: Learning sparse retrievable cross-interactions for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 548–557.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Aaron Van Den Oord, Oriol Vinyals, and 1 others. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Zhe Wang, Liqin Zhao, Biye Jiang, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2020. Cold: Towards the next generation of pre-ranking system. *arXiv preprint arXiv:2007.16122*.

Dongbo Xi, Zhen Chen, Peng Yan, Yinger Zhang, Yongchun Zhu, Fuzhen Zhuang, and Yu Chen. 2021. Modeling the sequential dependence among audience multi-step conversions with multi-task learning in targeted display advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3745–3755.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion proceedings of the web conference 2020*, pages 441–447.

Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM conference on recommender systems*, pages 269–277.

Yantao Yu, Weipeng Wang, Zhoutian Feng, and Daiyue Xue. 2021. A dual augmented two-tower model for online large-scale recommendation. *DLP-KDD*.

Zheng Zhang, Jinyi Li, Yihuai Lan, Xiang Wang, and Hao Wang. 2025. An empirical study on prompt compression for large language models. *arXiv preprint arXiv:2505.00019*.

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. 2025. Addressing representation collapse in vector quantized models with one linear layer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22968–22977.

A Details of Datasets

Datasets	Train set Size	Test set Size	Avg Length			CLS Label
			Query	Document	Reason	
QLQ	2741326	24227	30.87	229.55	146.48	4
BQ	100000	20000	11.63	12.09	74.94	2
ATEC	82477	20000	13.33	13.35	75.23	2
LCQMC	238766	21302	10.68	11.19	69.77	2
NLI	941445	39307	15.81	8.50	58.45	3
QQP	297708	32965	11.13	11.41	55.96	2

Table 4: Detailed information of the dataset.

Tables 4 presents detailed information about the Chinese and English datasets employed in this paper. It includes the sizes of the training and testing sets, along with the average lengths of queries, documents and reasons. Figure 5a to 5f show the length distribution of query and document for different datasets.

In addition, for the prompt compression and quantization experiments, we further augmented the QLQ dataset by incorporating 213367 snapshot samples and adopting a Conditional Fine-Tuning (CFT) training data strategy. Specifically, non-snapshot data were explicitly marked as "[NSP] + non-snapshot," while snapshot data were similarly labeled as "[SP] + snapshot."

The reconstruction process of the open-source text-matching datasets is illustrated in Figure 6.

B Implementation Details

In our experiments, we pre-trained a 1B-parameter LLM backbone with 1536 hidden dim and 48 layers using both general-domain and industrial search advertising corpora.

The proposed UGD and its variants for prompt compression and quantization are trained through full-parameter fine-tuning. During training, the learning rate is set to $5e-6$, and the batch size is 32. The AdamW optimizer with $\beta_1 = 0.9$ and $\beta_1 = 0.999$. The warmup steps are set to 4000. It is implemented on 8 NVIDIA A800 GPUs for 5 epochs.

The hyperparameters for UGD are defined as the weight coefficients of the corresponding loss terms. For the two-tower paradigm, the weights $\{\alpha, \beta, \gamma, \lambda, \mu\}$ are set to $\{1, 1, 1, 10, 10\}$, controlling the contributions of $\mathcal{L}^t_{cls}$, $\mathcal{L}^s_{cls}$, $\mathcal{L}_{gen}$, $\mathcal{L}_{kl}$, and $\mathcal{L}^{emb}_{kl}$, respectively (see Eq. 6). For the single-tower paradigm, only $\mathcal{L}^s_{cls}$ and $\mathcal{L}_{gen}$ are considered, with weights $\{\beta, \gamma\} = \{1, 1\}$ (see Eq. 7).

The training process of our UGD and the corresponding prompt compression and quantization methods is as follows. The training pipeline consists of three stages. First, the non-quantized UGD model is pretrained via a single-stage supervised fine-tuning. Second, the pretrained model is used to perform inference on the training set to obtain sampled compressed LP vectors, which are clustered using KMeans to initialize the first codebook of the linearized RQVAE, while the remaining residual codebooks are initialized to zero. Finally, the pretrained UGD model and the clustered vectors are jointly loaded to warm-start the full model, followed by conditional fine-tuning on the training data.

C Case Studies

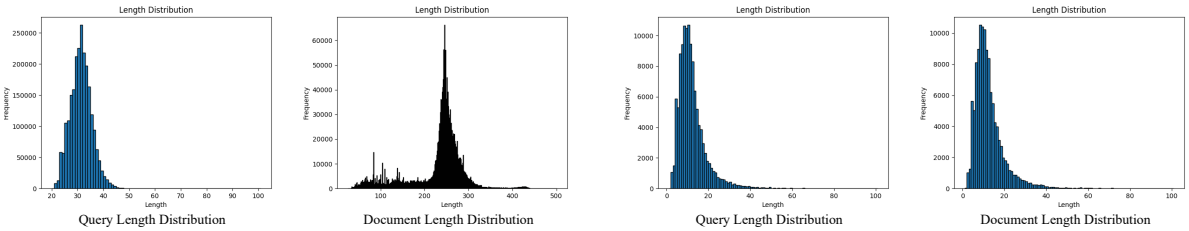
In this section, we conduct a qualitative analysis of our UGD matching paradigm from the perspective of case studies.

As shown in Figure 7, in the field of search advertising, it is necessary to filter out the most relevant landing pages from the candidate set of landing pages based on user queries. Usually, the landing page content is highly discrete and noisy. Compared with traditional matching paradigms, our UGD matching paradigm not only improves discriminative performance but also provides corresponding reasons.

As mentioned in the previous section, ATEC, BQ, and LCQMC are all Chinese question matching datasets. As shown in Figure 8, taking BQ dataset as an example, Through the innovative design of the generation task and knowledge distillation, the model can better understand the meanings of two sentences, thereby providing more logical discrimination results.

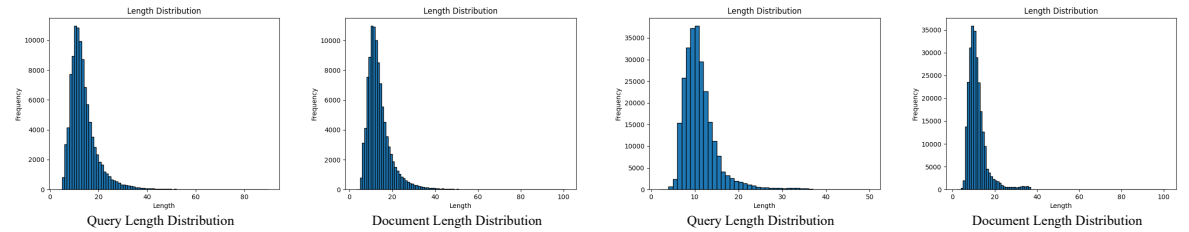
Similarly, we also conduct the case studies on the English dataset NLI in Figure 9. The semantic relationship between two sentences is marked as {"0": "entailment", "1": "neutral", "2", "contradiction"}. This can also prove the above conclusion.

In summary, the innovative integration of generation tasks and knowledge distillation within our UGD matching paradigm has significantly enhanced the model's capacity to comprehend sentence meanings. This improvement has, in turn, enabled the generation of more logically sound discrimination outcomes. The positive results across a range of experiments validate the superiority and effectiveness of the UGD matching paradigm in diverse matching scenarios, highlighting its potential to outperform traditional approaches and drive



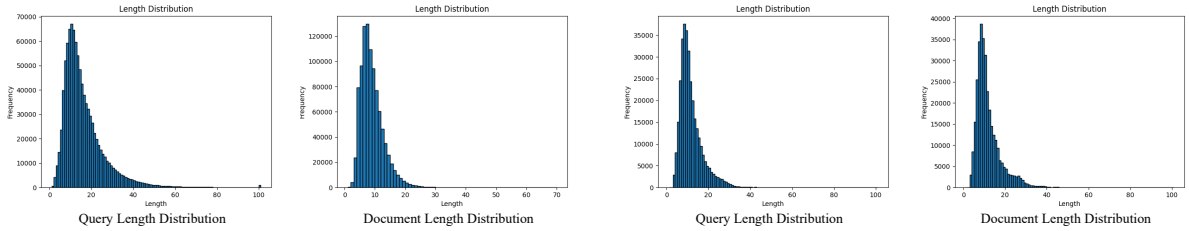
(a) QLQ

(b) ATEC



(c) BQ

(d) LCQMC



(e) NLI

(f) QQP

Figure 5: An illustration of the distribution of the lengths of queries and documents across different datasets.

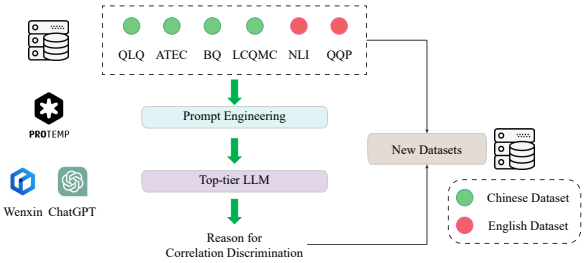


Figure 6: A flowchart for generating reason labels based on the top-tier LLMs.

Figure 7: The case studies of QLQ.

progress in related research fields.

D Online Testing and Other Applications

During the A/B testing period, we closely monitor the performance of the UGD matching paradigm. Compared with the previously deployed model, the UGD TTM matching paradigm effectively reduces the proportion of 0-score landing pages by **1.87%**. Meanwhile, the UGD STM matching paradigm achieves a **3.9%** increase in filtering accuracy, a **1.6%** increase in filtering volume, a **3.2%** decrease in 0-score landing page display, a **1.36%** decrease in 1-score landing page display, and a **3.09%** increase in 2- and 3-score landing page display.

Furthermore, the proposed approach demonstrates broad applicability across diverse domains. In medical diagnosis assistance, it not only provides confidence levels for different disease categories but also generates detailed explanations for observed symptoms and potential diagnostic suggestions. This dual functionality enriches the diagnostic process, offering medical professionals and patients more comprehensive information to support informed decision-making. In question-answering systems, the model generates accurate responses to user queries while estimating the correlation probability between the input questions and

Figure 8: The case studies of BQ.

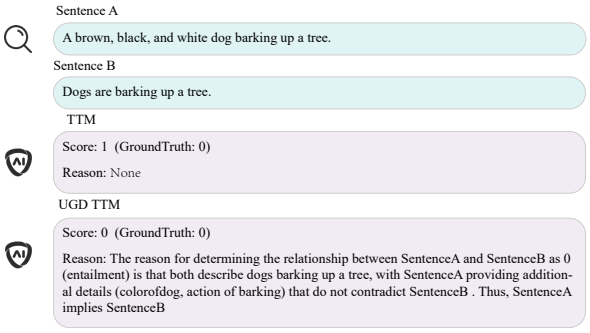


Figure 9: The case studies of NLI.

entries in the knowledge base. Such probability assessments enable a refined evaluation of answer relevance, enhancing overall response quality.

In recommendation systems, the approach provides detailed explanations for recommended items, clarifying the underlying rationale behind each suggestion. By offering this transparency, the system improves user understanding, fosters trust, and can potentially increase user engagement with the recommended products or content.

E Related Work

E.1 Traditional Matching Paradigm

From the perspective of interaction patterns, traditional matching paradigms can be classified into two-tower and single-tower approaches. Two-tower matching is the dominant paradigm in dense retrieval (Huang et al., 2013; Reimers, 2019) and is widely deployed across various applications (Covington et al., 2016; Yi et al., 2019). However, it suffers from limited feature interaction capability. To address this, some studies have incorporated SEBlocks to enhance feature representation (Wang et al., 2020) or ResNet architectures to facilitate more effective information propagation and fusion (Shan et al., 2016), yet the improvements remain modest. Other work has explored knowledge distillation to transfer knowledge from single-tower to two-tower models (Lin et al., 2023), but training the two architectures separately for distillation incurs additional computational cost and potential inconsistency. In contrast, single-tower models, benefiting from an all-to-all feature interaction pattern, have been shown to outperform two-tower models (Kim et al., 2021). Nevertheless, these traditional paradigms are insufficient to meet the requirements of the LLM era, where richer feature interactions and generative capabilities are increasingly essential.

E.2 General Language Model

With the continuous advancement of general language model research, it has become feasible to realize a unified generative and discriminative matching paradigm. General language models aim to perform well across three main categories of tasks: natural language understanding (NLU), unconditional generation, and conditional generation, as exemplified by UniLM (Dong et al., 2019) and GLM (Du et al., 2021). Inspired by these developments, we implement attention-mask partitioning to integrate generative and discriminative tasks within a single LLM framework. Moreover, as the capabilities of LLMs continue to grow, for instance, models such as GLM, Qwen, DeepSeek, and Kimi (Du et al., 2021; Yang et al., 2025; Liu et al., 2024; Team et al., 2025), they are increasingly applicable to industrial search, recommendation, and advertising, further empowering real-world business scenarios.

E.3 Prompt Compression and Quantization

Prompt compression has emerged to reduce the length and computational overhead of LLM inputs while preserving task-relevant information. Early methods identify and prune low-importance tokens (Jiang et al., 2023), and later work categorizes approaches into hard token pruning and soft continuous prompt reduction (Li et al., 2025a), including generalized compressors (Li et al., 2025b) and attention-based adaptive schemes (Chen et al., 2025). Empirical studies further examine the efficiency and effectiveness of different compression strategies across tasks (Zhang et al., 2025), highlighting ongoing efforts to make long-context prompting more scalable and efficient.

Prompt quantization aims to discretize continuous prompt or representation embeddings to reduce storage and inference costs while preserving semantic fidelity. A foundational method is the Vector-Quantized Variational Autoencoder (VQ-

VAE), which maps continuous encoder outputs to discrete codes via a learned codebook, enabling compact latent representations (Van Den Oord et al., 2017). To address the limited expressivity of single-stage quantization, Residual Quantized VAE (RQVAE) hierarchically quantizes residual errors across multiple stages, producing coarse-to-fine discrete representations and substantially increasing latent expressivity with a fixed codebook (Lee et al., 2022). Additionally, to improve codebook utilization and avoid local optima, some works apply linearization of the codebook space (Zhu et al., 2025).