CTR-Guided Generative Query Suggestion in Conversational Search

Erxue Min[1†], Hsiu-Yuan Huang[2,3†], Xihong Yang[1†], Min Yang[1], Xin Jia[1], Yunfang Wu[2,3], Hengyi Cai[1], Junfeng Wang[1], Shuaiqiang Wang[1], Dawei Yin[1]

[1]Baidu Inc.
[2]School of Computer Science, Peking University
[3]National Key Laboratory for Multimedia Information Processing, Peking University

Correspondence: jemmryx@pku.edu.cn

[†]Equal contribution.

Abstract

Generating effective query suggestions in conversational search requires aligning model outputs with user click preferences. However, directly optimizing for these preferences is difficult because click signals are sparse and inherently noisy. To address this, we propose Generative Query Suggestion (GQS), a generative framework that leverages click modeling to denoise implicit feedback and enables reliable preference optimization for improving real-world user engagement. GQS consists of three key components: (1) a *Multi-Source CTR Modeling* module that captures diverse contextual signals to estimate fine-grained click-through rates, thereby constructing more reliable user click-preference pairs; (2) a *Diversity-Aware Preference Alignment* strategy using CTR-weighted Direct Preference Optimization (DPO), which balances relevance and semantic diversity; and (3) a *CTR-Calibrated Iterative Optimization* process that jointly refines both the CTR model and the query suggestion model across training rounds, enabling effective data reuse. Experiments on two real-world tasks demonstrate that GQS outperforms strong baselines in CTR, relevance, and diversity.

1 Introduction

Conversational search systems such as AI assistants have emerged as a new paradigm for search, where users interact through natural language queries and receive not only direct answers but also proactive query suggestions. As shown in Figure 1, these suggestions are typically presented as clickable elements in the interaction interface, helping users refine or extend their queries with minimal effort.

Recent work on query suggestion has explored using general-purpose Large Language Models (LLMs) as the backbone, leveraging their internal knowledge to alleviate cold-start limitations and generate suggestions in conversational search (He et al., 2023; Wang et al., 2023; Liu et al., 2024; Zhao and Dou, 2024; Wang et al., 2024b). While these models produce fluent and plausible suggestions, they lack grounding in real user search behavior and often fail to align with actual user preferences. Retrieval-augmented generation (RAG) frameworks (Bacciu et al., 2024; Shen et al., 2024; Wang et al., 2024a) incorporate external search-related knowledge to bridge this gap, but still depend on LLMs' retrieval and summarization abilities rather than genuine preference alignment. In conversational search systems, user clicks provide one of the most direct signals of user preference. However, how to effectively leverage such signals to guide query suggestion remains an open challenge.

Figure 1: Examples of AI assistant interfaces where query suggestions are presented as clickable elements.

We identify three key challenges in building effective click-preference-aligned Generative Query Suggestion (GQS) framework: (1) Real-world click data is inherently noisy and biased, making it unsuitable for out-of-the-box use (Sang et al., 2024; Jin et al., 2024; Islam et al., 2025). This calls for reliable click-through rate (CTR) modeling to calibrate and denoise click signals before applying them for preference alignment. (2) Directly optimizing for CTR often reduces the semantic diver-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 2624–2634
November 4-9, 2025

sity of suggestions, resulting in repetitive or narrowly focused outputs (Peng et al., 2023; Kirk et al., 2024; Chen et al., 2024; Zhao et al., 2025). Balancing CTR alignment with diversity preservation is essential to sustain user engagement. (3) Collecting sufficient online feedback for preference optimization is time-consuming and costly, underscoring the importance of fully exploiting existing offline click data through efficient iterative improvement strategies (Chen et al., 2023; Kang et al., 2023; Chen et al., 2024).

In this paper, we propose a novel generative framework for AI assistant query suggestion that addresses these challenges through three key components: (1) a ***Multi-Source CTR Modeling*** strategy that integrates multiple signals to build a robust CTR predictor for reliable click signal estimation. (2) a ***Diversity-Aware Click Preference Alignment*** method that jointly optimizes CTR alignment and semantic diversity using Direct Preference Optimization (DPO) over diversity-enhanced preference pairs. (3) a ***CTR-Calibrated Iterative Optimization*** process that applies importance sampling to effectively reuse offline click data for iterative preference alignment, enabling continuous improvement without relying on costly online feedback collection.

Our experiments on industrial-scale AI assistant datasets demonstrate substantial improvements in CTR and diversity metrics, validating the effectiveness of our approach. To summarize, our contributions are:

- We propose an end-to-end Generative Query Suggestion (GQS) framework tailored for AI assistant query suggestion.
- We construct Multi-Source CTR Modeling for Diversity-Aware Click Preference Alignment to align generation with real user preferences, and propose a CTR-Calibrated Iterative Optimization approach for efficient improvement.
- We validate our method through large-scale A/B online experiments, showing significant improvements in user engagement and suggestion quality.

2 Background

We formulate query suggestion in conversational search as a conditional generation task, where an LLM generates novel queries tailored to the user's evolving intent instead of retrieving queries from a fixed set. This enables proactive exploration beyond explicit user input, improving both experience and search coverage.

To maximize the quality and relevance of generated suggestions, our prompt design explicitly incorporates multiple contextual sources as side information, as illustrated in Appendix A.1: (1) *Current user query* $q_{u(\tau)}$, representing the user's immediate information need at conversation turn τ; (2) *Current assistant response* $r_{(\tau)}$, showing how the system interprets and addresses the current query; (3) *Conversation history* $h_{(\tau)}$, consisting of all prior user–assistant turns before τ, providing broader dialog context; (4) *User profile features* u, including age, interests, or behavioral tags, to encode user background and preferences; **(5)** *Co-occurring queries* $\mathcal{C}_{(\tau)}$, mined from similar historical conversations as external user behaviour references (see Appendix A.2 for construction details).

Formally, we represent the context as $\mathcal{X}_{(\tau)} = \{q_{u(\tau)}, r_{(\tau)}, h_{(\tau)}, u, \mathcal{C}_{(\tau)}\}$. The LLM is prompted to jointly generate a set of candidate queries $\hat{Q}^{(t)}_{(\tau)} = \{\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_N\}$ from the conditional distribution:

$$\hat{Q}^{(t)}_{(\tau)} \sim \pi^{(t)}_{\theta}(q_1, \ldots, q_N \mid \mathcal{X}_{(\tau)}), \tag{1}$$

where $\pi^{(t)}_{\theta}$ denotes the search preference-aligned LLM under the current optimization turn t, and $\hat{Q}^{(t)}_{(\tau)}$ is the resulting set of n candidate suggestions. By explicitly emphasizing current user query alongside above side information as content $\mathcal{X}_{(\tau)}$, this prompt structure guides the model to generate suggestions $\hat{Q}^{(t)}_{(\tau)}$ related to user search intent while maintaining holistic awareness.

3 Methodology

In this section, we introduce our framework for CTR-driven generative query suggestion in conversational AI assistent search. The method consists of three major components: (1) a multi-source contextual CTR modeling to obtain click prefence signals; (2) CTR-weighted Direct Preference Optimization (DPO) with diversity regularization for click preference alignment; (3) an iterative training scheme with calibrated CTR correction to ensure efficient policy improvement.

3.1 Multi-Source Contextual CTR Modeling

User click signals extracted from logs offer a natural form of preference supervision, but are inherently noisy and biased (e.g., due to position effects),

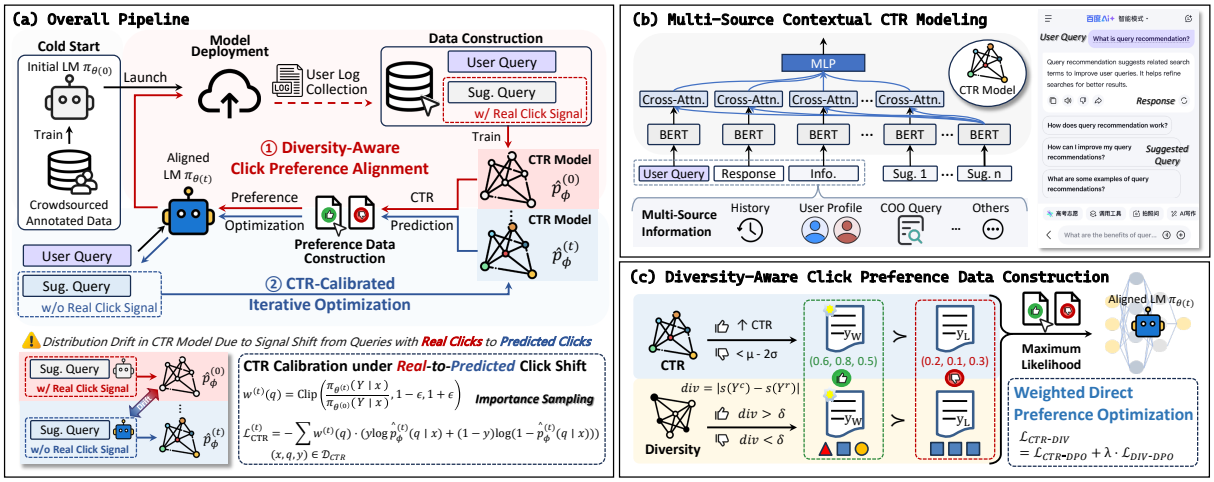

Figure 2: An overview of our proposed method. (a) The overall pipeline of the framework. (b) Multi-Source contextual Modeling for reliable CTR estimation. (c) Diversity-Aware Click Preference Data Construction for balancing CTR alignment and diversity.

making them unsuitable for direct use. A central challenge in aligning a GQS LLM with real user preferences lies in accurately estimating the click-through rate (CTR) of each generated query under diverse conversational contexts.

To address this, we propose a multi-source CTR prediction model that: (1) encodes heterogeneous contextual signals, (2) uses cross-attention to model the interaction between each context and the target query, and (3) integrates these signals for final CTR estimation, as illustrated in Figure 2(b).

Let q_n denote the n-th generated query suggestion, where $n \in \{1, 2, \ldots, N\}$. Let $\mathcal{X} = \{x_1, x_2, \ldots, x_K\}$ represent a set of contextual sources, including the user's current query, AI response, dialog summary, user profile, co-occurring queries, and prior generated queries. Each $x_k \in \mathcal{X}$ is encoded using a shared BERT encoder:

$$\mathbf{H}_k = \text{BERT}(x_k), \quad \mathbf{H}_n = \text{BERT}(q_n).$$

To model the relationship between q_n and each context x_k, we apply single-head cross-attention:

$$\mathbf{A}_k = \text{softmax}\left(\frac{\mathbf{H}_n \mathbf{W}_Q (\mathbf{H}_k \mathbf{W}_K)^\top}{\sqrt{d}}\right)(\mathbf{H}_k \mathbf{W}_V),$$

$$\mathbf{e}_k = \text{Pool}(\mathbf{A}_k),$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are learnable projection matrices, and $\text{Pool}(\cdot)$ denotes mean pooling over attended tokens to produce fixed-length representations. To account for position bias in the generated query list, we incorporate a learnable position embedding $\mathbf{e}_{\text{pos}} = \text{Embed}(p)$, where p is the position index of q_n.

The final CTR prediction for q_n is computed by concatenating all contextual embeddings and feeding them into a multi-layer perceptron:

$$\hat{y}_n = \sigma\left(\text{MLP}\left(\left[\mathbf{e}_1; \ldots; \mathbf{e}_K; \mathbf{e}_{\text{pos}}\right]\right)\right).$$

The model is trained using binary click labels as supervision, optimizing a standard binary classification loss. This attention-based architecture enables fine-grained modeling of how each type of context contributes to click likelihood, supporting CTR-informed preference data construction in the next stage.

3.2 CTR-Driven Preference Optimization for Query Generation

To align the GQS model with real user preferences, we formulate training as a preference-based learning problem. Rather than directly maximizing predicted CTR which may be poorly calibrated, we aim to optimize for response-level preference rankings inferred from CTR scores. We build upon DPO (Rafailov et al., 2023), extending it with CTR-weighted importance and an auxiliary diversity objective.

Step 1: Response Scoring and Filtering. For each prompt x, we generate a set of M candidate responses $\{Y_1, \ldots, Y_M\}$ using the current GQS model. Each response $Y_i = [q_1, \ldots, q_N]$ $(1 \le i \le M)$ consists of a list of N queries. Using the CTR model described in Section 2, we compute the total predicted click likelihood of each response:

$$r(Y_i) = \sum_{j=1}^{N} \hat{p}(q_j), \tag{2}$$

where $\hat{p}(q_j)$ is the predicted CTR of query q_j.

To select a *preferred response* Y^c_{ctr}, we filter candidate responses with low semantic diversity (below a fixed threshold δ) and choose the one with the highest $r(Y)$. Semantic diversity is assessed using GPT-4o, which takes the full list of suggested queries along with the user query as input and outputs a diversity score.

For the *rejected response* Y^r_{ctr}, we follow robust preference construction principles and select from candidates with CTR scores in the lower tail of the score distribution. Specifically, we reject candidates whose CTR falls below $\mu - 2\sigma$ (Xiao et al., 2025), where μ and σ are computed over all $r(Y_i)$ within the candidate set. This enforces that each preference pair exhibits both semantic and reward-level separation.

Step 2: Weighted CTR-DPO Loss. Let π_θ be the current generation model and π_{ref} be a frozen reference model (e.g., a snapshot before DPO training). Given a preference pair (Y^c_{ctr}, Y^r_{ctr}) for prompt x, we compute the DPO loss as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\log \sigma \left(\beta \log \frac{\pi_\theta(Y^c_{\text{ctr}} \mid x)}{\pi_{\text{ref}}(Y^c_{\text{ctr}} \mid x)} - \beta \log \frac{\pi_\theta(Y^r_{\text{ctr}} \mid x)}{\pi_{\text{ref}}(Y^r_{\text{ctr}} \mid x)} \right), \tag{3}$$

where β is a temperature hyperparameter, and σ is the sigmoid function.

To reflect the *relative importance* of each preference pair, we compute a sample-level weight:

$$\alpha = \sigma\left(\gamma \cdot (r(Y^c_{\text{ctr}}) - r(Y^r_{\text{ctr}}))\right), \tag{4}$$

where γ is a scaling factor. Intuitively, larger CTR gaps indicate more reliable preferences and therefore contribute more strongly to model updates.

The CTR-weighted DPO loss is then defined as:

$$\mathcal{L}_{\text{CTR-DPO}} = \alpha \cdot \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}). \tag{5}$$

Step 3: Diversity-Aware Preference Learning. In addition to CTR-guided preference learning, we introduce diversity-aware supervision to encourage *semantic diversity* in the generated query lists. Specifically, we construct auxiliary preference pairs $(Y^c_{\text{div}}, Y^r_{\text{div}})$ where the CTR scores are close, but their diversity scores differ significantly (i.e., $|s(Y^c_{\text{div}}) - s(Y^r_{\text{div}})| \geq \delta$). These pairs are used to apply a DPO-style objective that encourages preference toward the more diverse response, using the same loss structure as in the CTR-aligned case. The final training objective combines both preference signals:

$$\mathcal{L}_{\text{CTR-DIV}} = \mathcal{L}_{\text{CTR-DPO}} + \lambda \cdot \mathcal{L}_{\text{DIV-DPO}}, \tag{6}$$

where λ is a tunable coefficient balancing alignment accuracy and diversity regularization.

This framework enables structured and interpretable alignment of the generation model to user behavior, while preserving diversity and avoiding generic response collapse.

3.3 Iterative Training with Calibrated CTR Predictor

Collecting online user click feedback for CTR model training is often time- and resource-intensive. To improve efficiency, we aim to reuse each batch of logged feedback for multiple rounds of DPO-style fine-tuning. However, this introduces a critical challenge: after each iteration of DPO, the distribution of the generation policy π_θ shifts. Consequently, the CTR model—originally trained on real click data collected from queries generated by the initial policy $\pi_{\theta^{(0)}}$, becomes misaligned with the queries generated by the updated policy $\pi_{\theta^{(t)}}$. This distribution mismatch stems from a distribution drift in the CTR model, caused by a signal shift from real user clicks to predicted clicks, introducing bias into preference modeling.

To address this issue, we propose an iterative framework where a CTR model is explicitly calibrated via *importance weighting* using the likelihood ratio between the initial policy and the optimized t step's policy $\pi_{\theta^{(t)}}$.

Let $\pi_{\theta^{(0)}}$ denote the generation policy used to generate the response list when the CTR training data was collected, and $\pi_{\theta^{(t)}}$ denote the policy at iteration t. For each response $Y = [q_1, \ldots, q_N]$ for each prompt x in the CTR dataset $\mathcal{D}_{\text{ctr}}$, we estimate its probability under both $\pi_{\theta^{(0)}}$ and $\pi_{\theta^{(t)}}$:

$$w^{(t)}(Y) = \text{Clip}\left(\frac{\pi_{\theta^{(t)}}(Y \mid x)}{\pi_{\theta^{(0)}}(Y \mid x)}, 1-\epsilon, 1+\epsilon\right), \tag{7}$$

where $w^{(t)}(Y)$ is the importance weight used to reweight the loss contribution of generated response Y in the CTR training process.

We then train the CTR predictor $\hat{p}^{(t)}_\phi$ at iteration t using a weighted binary cross-entropy loss:

$$\begin{aligned} \mathcal{L}^{(t)}_{\text{CTR}} = -\sum_{(x,Y)\in\mathcal{D}_{\text{ctr}}} \sum_{(q,y)\in Y} w^{(t)}(Y) \cdot [\\ y \log \hat{p}^{(t)}_\phi(q \mid x) + (1-y) \log\left(1 - \hat{p}^{(t)}_\phi(q \mid x)\right)], \end{aligned} \tag{8}$$

where $y \in \{0, 1\}$ denotes the click label, and $\hat{p}^{(t)}_\phi$ is the calibrated CTR model trained for the current policy.

Once $\hat{p}^{(t)}_\phi$ is trained, we regenerate the reward estimates $r(Y)$ for candidate responses, re-sample

updated preference pairs (Y^c, Y^r) as described in Section 3.2, and perform the next round of DPO optimization over the updated $\pi_{\theta^{(t)}}$:

$$\theta^{(t+1)} \leftarrow \arg\min_{\theta} \mathcal{L}^{(t)}_{\text{CTR-DIV}}(\pi_\theta; \pi_{\theta^{(t)}}). \quad (9)$$

This iterative calibration mechanism enables effective reuse of collected CTR data across multiple DPO updates while mitigating reward drift caused by evolving generation policies.

4 Experiments

4.1 Experimental Setup

Tasks and Datasets. We evaluate our framework on two query suggestion tasks in Baidu's conversational AI assistant, with examples shown in Figure 3: **(T1) General query suggestion.** After answering a user's query, the system offers three follow-up query suggestions for preference elicitation. **(T2) Creative-writing snippet generation.** Users receive a guiding sentence and approximately eight clickable snippet suggestions (e.g., "shorter", "more creative", "more elegant") for refining creative-writing prompts.

Here, we adopt real large-scale search logs from the Baidu AI assistant for model training and evaluation. Specifically, for each task, we first build a CTR predictor training dataset using more than 20 million exposure records sampled from 14 days of click logs of the initial SFT baseline. Afterwards, we randomly sample 10,000 queries from the 15th-21st days of search logs as the training set for CTR alignment, with 10,000 queries from the 22nd day as the test set.

Evaluation Metrics and Baselines. To provide a thorough evaluation of the model's performance, we utilize three metrics: Click-Through Rate (CTR), Relevance and Diversity. Besides, we adopt ERNIE Speed (21B) (Sun et al., 2020) as our foundation model. We select three unaligned method as baselines: 1) SFT, which means the model is only supervisedly finetuned by human-annotated datasets, 2) Few-shot, which means the model is prompted by demonstrating examples, 3) RAGQS (Bacciu et al., 2024), which retrieves similar queries from query logs to construct better few-shot examples. In terms of alignment models, We evaluate our approach in comparison with four baselines: SFT, KTO (Ethayarajh et al., 2024), SimPO (Meng et al., 2024), and DPO (Rafailov et al., 2023), where preference pairs are derived

Figure 3: Illustration of the two query suggestion tasks in Baidu's conversational AI assistant: (T1) general query suggestion and (T2) creative-writing snippet generation.

from (1) user click behaviour and (2) estimated CTR values. The AI assistant presents 3 and 8 query suggestions as clickable items for Task 1 and Task 2, respectively. Besides, we set λ as 0.1 for all experiments.

4.2 Overall Results and Analysis

The overall experimental results are presented in Table 1, and we have the following observation:

The CTR performance of SFT with human-annotated training data and few-shot method is similar, suggesting that utilizing more annotated data for SFT does not necessarily enhance recommendation performance. This is reasonable because even with high-quality annotations for GQS tasks, there remains a significant discrepancy between the annotators' preferences and the users' actual click preferences.

In both the Click-Aligned and CTR-Aligned scenarios, we observe that improvements in CTR performance are accompanied by enhanced relevance but reduced diversity. This trade-off arises because high-CTR samples often share similar patterns, which biases the model toward generating less diverse suggestions.

For Click-Aligned, although real user click data were used for preference alignment, we observe an overall decline in CTR in both Task 1 and Task 2. For example, in Task 1, applying the KTO, SimPO, and DPO algorithms led to CTR decreases of 1.61%, 1.71%, and 1.42%, respectively. We attribute this performance drop to the noise and randomness inherent in users' click and non-click

Scenario	Method	Task 1			Task 2		
		CTR	Relevance	Diversity	CTR	Relevance	Diversity
Base	SFT	0.00	80.53	85.63	0.00	84.86	81.42
	Few-shot	1.14	78.12	80.17	-5.81	82.66	79.46
	RA-GQS	1.45	79.48	83.93	0.21	82.33	80.45
Click-Aligned	SFT_{clk}	4.61	81.89	83.49	0.76	84.14	78.11
	KTO_{clk}	-1.61	79.45	**86.81**	-1.69	84.55	81.97
	$SimPO_{clk}$	-1.71	78.34	84.36	1.91	86.05	78.20
	DPO_{clk}	-1.42	80.91	83.94	1.54	84.65	76.78
CTR-Aligned	SFT_{ctr}	19.44	84.08	80.12	4.45	87.32	75.14
	KTO_{ctr}	30.78	86.74	77.68	7.63	89.04	72.12
	$SimPO_{ctr}$	32.99	90.36	79.29	16.98	93.86	70.61
	DPO_{ctr}	60.15	91.15	65.73	25.51	95.83	59.43
Ours	**GQS**	**70.36**	**94.60**	86.04	**30.72**	**98.11**	**82.09**

Table 1: The overall performance of various models for GQS. Three training scenarios: Base (no click alignment), Click-Aligned (aligned with raw click data), and CTR-Aligned (aligned with calibrated CTR scores).

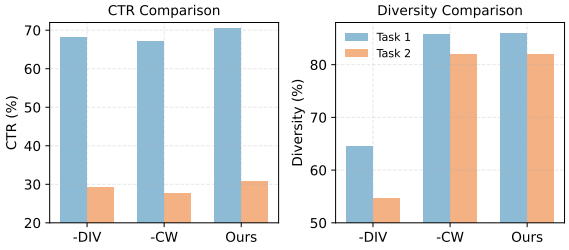

Figure 4: Ablation results of Preference Alignment

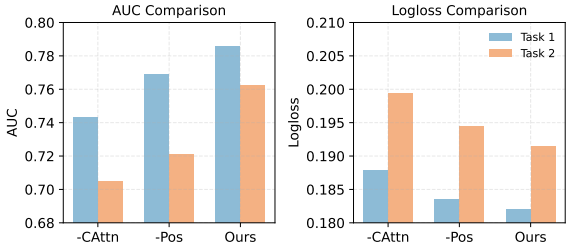

Figure 5: CTR model performance (AUC and Logloss) under different architectural components.

signals, which undermine the effectiveness of preference learning and highlight the limitations of directly using raw click data.

In comparison, methods that perform preference alignment based on predicted CTR signals consistently yield improvements. This indicates that filtering based on CTR scores predicted by CTR models is more significant than direct filtering based on click signals, thereby reducing noise and variance. This demonstrates that preference alignment is more effective than SFT alone, and understanding the patterns of negative samples is crucial for the model.

Our proposed GQS method jointly optimizes CTR-prediction preferences and diversity, thereby achieving CTR gains while simultaneously preserving high diversity and relevance. For example, in Task 2, compared to the DPO algorithm in the CTR-Aligned scenario, GQS improves CTR, relevance, and diversity by 5.21, 2.28, and 22.66 points, respectively.

4.3 Ablation Studies

In this subsection, we present experiments to evaluate the effectiveness of key components of GQS.

Ablation of preference alignment modules. We denote "-DIV" as the removal of the diversity-aware learning component, and "-CW" as the exclusion of the CTR-weighting strategy. As illustrated in Figure 4, ablating "-CW" leads to a substantial drop in CTR, highlighting the value of emphasizing preference pairs with larger CTR gaps. Removing "-DIV" results in a notable decrease in diversity scores, underscoring the role of diversity regularization in enhancing generative variability. Interestingly, CTR performance also slightly declines without the diversity objective, suggesting that diverse suggestions are more likely to capture user interest.

Analysis of CTR Model. Here, "-CAttn" refers to the removal of the cross-attention mechanism for multi-source fusion; in this variant, all contextual inputs are simply concatenated and fed into a BERT encoder. "-Pos" denotes the exclusion of positional embeddings. As shown in Figure 5, removing either component leads to a drop in AUC and an increase in Logloss, indicating degraded prediction quality. These results highlight the importance of cross-attention in effectively integrating heteroge-

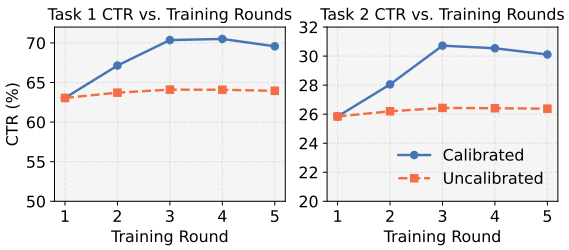

Figure 6: CTR progression over training rounds in Task 1 and Task 2, comparing calibrated vs. uncalibrated preference guidance.

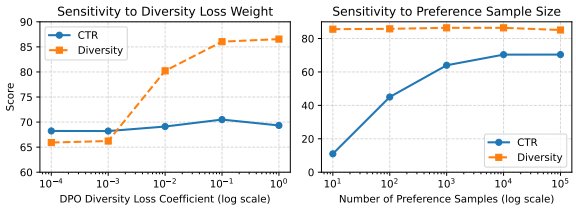

Figure 7: Sensitivity analysis results. (a) Effect of diversity loss coefficient λ on CTR and diversity. (b) Effect of alignment sample size on model performance.

neous information, as well as the crucial role of positional embeddings in modeling position bias.

4.4 Effect of Iterative Training with CTR Calibration

We evaluate the impact of our iterative training strategy, where the CTR model is progressively updated to guide preference optimization. As shown in Figure 6, CTR performance improves notably in the first few rounds—for example, Task 1 improves by 7.45 points (from 63.05 to 70.50)—and stabilizes afterward. This demonstrates that iterative calibration effectively enhances alignment quality without overfitting. In contrast, a baseline using a fixed CTR model across rounds yields only marginal improvements (e.g., Task 1 increases by 1.03 points, from 63.05 to 64.08), indicating the limited reliability of static preference signals. These results highlight the benefit of updating the reward model to reflect evolving generation behavior. We also observe a saturation trend beyond the third round, where gains become negligible or slightly decline (e.g., Task 2 drops by 0.61 points, from 30.72 to 30.11). This suggests that excessive iterations may offer diminishing returns, reinforcing the need to select a moderate number of training rounds to balance effectiveness and efficiency.

5 Sensitive Analysis

In this subsection, we conduct experiments on Task 1 to analysis the sensitivity of the hyperparameters in this paper.

Sensitivity Analysis of Diversity Loss Coefficient λ. We conduct experiments to analyze the impact of the diversity loss weight λ in our preference optimization. We test values from $\{1.0, 0.1, 0.01, 0.001, 0.0001\}$ and summarize the results in Figure 7(a). The results show that moderately weighted diversity loss leads to better balance between CTR and diversity. For instance, $\lambda = 0.1$ achieves the best CTR (70.50) while maintaining high diversity (86.04). When λ becomes too small, the diversity benefit diminishes significantly (e.g., 66.22 at $\lambda = 0.001$), though CTR still remains relatively stable. Conversely, larger values like $\lambda = 1.0$ ensure high diversity (86.55), but lead to a lower CTR (69.33). These results highlight the importance of moderate regularization strength—$\lambda = 0.1$ offers the best overall trade-off.

Sensitivity Analysis of Number of Alignment Samples. We further examine the effect of the number of preference samples used in alignment training. We vary the sample count in $\{10, 10^2, 10^3, 10^4, 10^5\}$ and report the performance on Task 1 in Figure 7(b). The results show that increasing the number of samples significantly improves CTR and diversity up to a certain threshold. For example, when using only 10 samples, CTR is 11.00 and diversity is 85.60, while using 1000 samples leads to 64.00 CTR and 86.40 diversity. The model saturates around 10^4 samples, with CTR reaching 70.36 and diversity maintaining 86.40. Further increasing the sample size to 10^5 yields marginal gains in CTR (70.39) but causes a slight drop in diversity (85.10). Therefore, we adopt 10^4 samples in our default setup, balancing computational cost and performance.

6 Conclusion

We propose **GQS**, a generative query suggestion framework for conversational search that aligns with user preferences through multi-source CTR modeling, diversity-aware preference optimization, and iterative CTR calibration. Extensive experiments on real-world tasks demonstrate that GQS consistently improves CTR, relevance, and diversity. Our findings highlight the importance of modeling user feedback and maintaining semantic diversity in preference-aligned generation.

Limitations

While GQS demonstrates strong performance in aligning query suggestions with user preferences, several limitations remain. First, our approach relies on click-through data, which may carry inherent biases such as position effects and delayed feedback. Although the CTR model helps mitigate this, residual bias could still influence optimization. Second, the iterative optimization process requires multiple rounds of reward model updates and generation training, which increases computational cost. In future work, we plan to explore more efficient update strategies and incorporate human-in-the-loop feedback for better alignment quality.

Ethics Statement

Use of AI Assistants We have employed ChatGPT as a writing assistant, primarily for polishing the text after the initial composition.

References

Andrea Bacciu, Enrico Palumbo, Andreas Damianou, Nicola Tonellotto, and Fabrizio Silvestri. 2024. Generating query recommendations via llms. *Preprint*, arXiv:2405.19749.

Jiaju Chen, Wang Wenjie, Chongming Gao, Peng Wu, Jianxiong Wei, and Qingsong Hua. 2024. Treatment effect estimation for user interest exploration on recommender systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1861–1871, New York, NY, USA. Association for Computing Machinery.

Xiaocong Chen, Siyu Wang, Julian McAuley, Dietmar Jannach, and Lina Yao. 2023. On the opportunities and challenges of offline reinforcement learning for recommender systems. *Preprint*, arXiv:2308.11336.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *Preprint*, arXiv:2402.01306.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian Mcauley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 720–730, New York, NY, USA. Association for Computing Machinery.

Md Aminul Islam, Kathryn Vasilaky, and Elena Zheleva. 2025. Correcting for position bias in learning to rank: A control function approach. *Preprint*, arXiv:2506.06989.

Jinqiu Jin, Sihao Ding, Wenjie Wang, and Fuli Feng. 2024. Understanding and counteracting feature-level bias in click-through rate prediction. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 838–841, New York, NY, USA. Association for Computing Machinery.

Yachen Kang, Diyuan Shi, Jinxin Liu, Li He, and Donglin Wang. 2023. Beyond reward: Offline preference-guided policy optimization. *Preprint*, arXiv:2305.16217.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of rlhf on llm generalisation and diversity. *Preprint*, arXiv:2310.06452.

Wenhan Liu, Ziliang Zhao, Yutao Zhu, and Zhicheng Dou. 2024. Mining exploratory queries for conversational search. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1386–1394, New York, NY, USA. Association for Computing Machinery.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

Kenny Peng, Manish Raghavan, Emma Pierson, Jon Kleinberg, and Nikhil Garg. 2023. Reconciling the accuracy-diversity trade-off in recommendations. *Preprint*, arXiv:2307.15142.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Lei Sang, Honghao Li, Yiwen Zhang, Yi Zhang, and Yun Yang. 2024. Adagin: Adaptive graph interaction network for click-through rate prediction. *ACM Trans. Inf. Syst.*, 43(1).

Xiaobin Shen, Daniel Lee, Sumit Ranjan, Sai Sree Harsha, Pawan Sevak, and Yunyao Li. 2024. Enhancing discoverability in enterprise conversational systems with proactive question suggestions. *arXiv preprint arXiv:2412.10933*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8968–8975.

Shuting Wang, Xin Yu, Mang Wang, Weipeng Chen, Yutao Zhu, and Zhicheng Dou. 2024a. Richrag: Crafting rich responses for multi-faceted

queries in retrieval-augmented generation. *Preprint*, arXiv:2406.12566.

Zhenduo Wang, Yuancheng Tu, Corby Rosset, Nick Craswell, Ming Wu, and Qingyao Ai. 2023. Zero-shot clarifying question generation for conversational search. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 3288–3298, New York, NY, USA. Association for Computing Machinery.

Zheng Wang, Bingzheng Gan, and Wei Shi. 2024b. Multimodal query suggestion with multi-agent reinforcement learning from human feedback. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1374–1385, New York, NY, USA. Association for Computing Machinery.

Yao Xiao, Hai Ye, Linyao Chen, Hwee Tou Ng, Lidong Bing, Xiaoli Li, and Roy Ka-wei Lee. 2025. Finding the sweet spot: Preference data construction for scaling preference optimization. *arXiv preprint arXiv:2502.16825*.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. 2025. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. *Preprint*, arXiv:2504.07912.

Ziliang Zhao and Zhicheng Dou. 2024. Generating multi-turn clarification for web information seeking. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1539–1548, New York, NY, USA. Association for Computing Machinery.

A Prompt Construction and COO Information Refilling

A.1 Prompt and Respones Format

Here we provide the detailed prompt and response format used for query generation.

Example Prompt and Response for Query Generation

Instruction:
You are an intelligent assistant helping users explore information in a conversational search session.
[Current User Query]:
How does intermittent fasting affect metabolism?
[Current Assistant Response]:
Intermittent fasting can influence metabolic health by...
[Conversation History Before This Turn] (h_t)**:**
User: What are effective diet methods for weight loss?
Assistant: There are various methods such as calorie restriction, low-carb diets, and intermittent fasting...
[User Profile Features]:
Age: 35; Interests: fitness, healthy diet, sustainable health practices
[Co-occurring Queries]:
Benefits of fasting for fat loss; Best time windows for intermittent fasting; Intermittent fasting vs. calorie restriction
Based on the current query and answer, previous conversation history, user profile, and related queries, generate 3 new and diverse follow-up queries that the user may find useful.

Generated Response:
Does fasting improve insulin sensitivity?
Best fasting methods for beginners
Fasting and muscle preservation tips

A.2 Co-Occurrence Query Information Construction and Refilling

We demonstrate how co-occurring query information is constructed and applied in query suggestion, as illustrated in Figure 8.

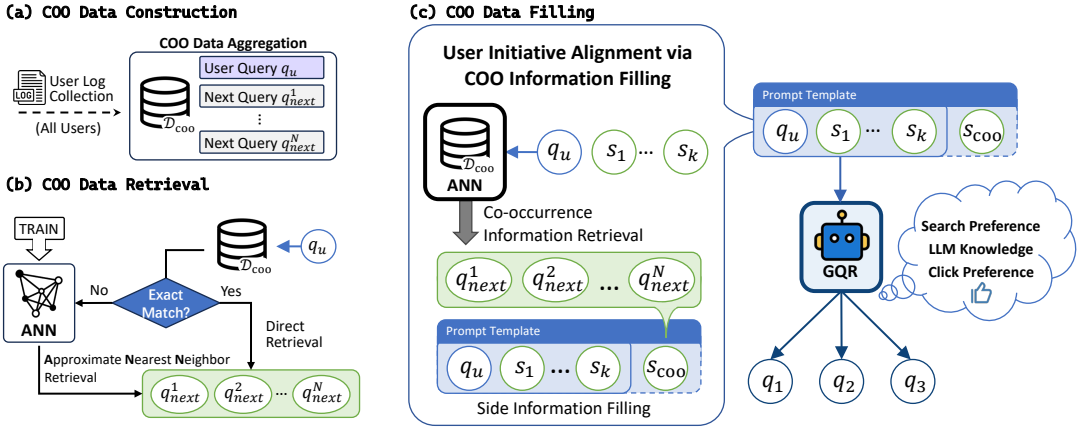

Figure 8: COO query information construction and refilling as side information.

As shown in Figure 8, given an input query q and optional side information $S = s_1, s_2, \ldots, s_k$, the LLM $\mathcal{M}$ generates suggested queries $RQ = q_1, q_2, \ldots, q_N$. The side information S is task-specific, such as session history or system responses in conversational search as shown in Section A.1. To incorporate co-occurring query signals as side information for aligning with user search preferences, we construct COO information as follows: (a) We extract co-occurrence (COO) signals $(q_u, q_{\text{next}}, c)$ from Baidu's complete user query logs, where q_u is the user query, q_{next} is a co-occurring query in the same session, and c denotes the frequency of this COO. These signals are aggregated into a COO dictionary D_{COO}. (b) We build an Approximate Nearest Neighbor (ANN) semantic matching system over D_{COO}, enabling efficient retrieval of relevant co-occurring queries based on user input. If the input query q_u has an exact match in D_{COO}, we retrieve its corresponding co-occurring queries; otherwise, we retrieve semantically similar queries from the ANN index to provide meaningful co-occurring suggestions. (c) The retrieved co-occurring queries are formatted as textual side information and incorporated into the prompt template to help align generated suggestions with user search preferences.

B Evaluation Criteria

Relevance. This metric evaluates the semantic alignment between a suggested query and the user's original query. Relevance is categorized into three levels:

- *High (1.0):* The suggested query is highly related to the user's query in both topic and intent, providing additional and pertinent information that complements the original query.
- *Moderate (0.5):* The suggested query is somewhat related but may not directly address the user's core intent, often offering tangential or loosely associated content.
- *Low (0.0):* The suggested query shows minimal topical or semantic overlap with the user's query, failing to align in intent or subject matter.

Diversity. This metric assesses the degree of novelty and non-redundancy of each suggested query along three dimensions:

- *Exclusivity from the user query:* The suggested query should not be semantically equivalent to the original query, ensuring it introduces a new perspective or subtopic.
- *Exclusivity from the AI response:* The suggested query should elicit new content from the AI model rather than reproducing information already covered in the initial response.
- *Exclusivity from other suggested queries:* Each suggested query in the recommended set should be distinct from others, contributing unique value to the overall list.

Scoring rule: A diversity score of 1.0 is assigned if all three aspects are satisfied, 0.5 if exactly two are satisfied, and 0.0 otherwise.

We utilize GPT-4o as an automatic judge to score both relevance and diversity. Prompts are carefully designed to instruct the model to assess each dimension independently and output scores following the defined criteria.