From Short Video to Clickable Search: RLVR-Enabled Listwise Query Suggestion with Retrieval-Augmented Context

Mingkai Tian* **Ye Xu** **Long Meng** **Liwei Chen** **Zhiheng Qin** **Yi Wang**
Kuaishou Technology
tianmingkai@kuaishou.com

Abstract

Short-video platforms now present tappable search entries beneath the video player, making it effortless for users to shift from passively watching to actively searching for information. Prior work on bottom-bar query generation conditions on titles and OCR to generate a single query per forward pass, constrains decoding with a trie, and evaluates against a single reference using edit-distance–style supervision—making it difficult to cover the diverse intents a video can trigger and to credit semantically equivalent query variants. Motivated by these limitations, we propose four complementary improvements. First, we reformulate the task as one-shot list generation, producing multiple distinct queries per video, and build multi-query ground truth from exposure and CTR logs. Second, we redesign offline evaluation with CTR-HungF1, a CTR-weighted set-matching metric via optimal assignment over token-level F1 score. Third, we enrich context with a video-to-video-to-query (V2V2Q) RAG pipeline to provide behavior-grounded background knowledge. Finally, we apply thinking-free RLVR with deterministic format checks and CTR-HungF1 rewards to train a compact LLM without reward models or CoT distillation. The resulting system yields strong offline and online improvements, and has been deployed on Kuaishou to serve hundreds of millions of users daily.

1 Introduction

Short-video platforms increasingly surface *clickable search suggestions* on the watch-page bottom bar, enabling one-tap search and lowering the cost of turning passive viewing into active information seeking. Figure 1 (left) illustrates this interaction. We study the problem of bottom-bar query generation: given a video, generate a list of queries that users are likely to click and that lead to high-quality search sessions.

*Corresponding author.

Figure 1: Bottom-bar query suggestion on Kuaishou and key improvements to prior work: V2V2Q RAG for context enrichment, multi-query output with multi-query ground truth, and thinking-free RLVR optimized by proposed metric CTR-HungF1.

Existing approaches to video-to-query generation is largely retrieval- or generation-based. Retrieval methods reuse historical queries and often struggle with new or compositional intents, while generative methods prompt language models to synthesize queries from video text. GREAT (Shao et al., 2025) is, to our knowledge, the only work tailored to short-video bottom-bar queries: it conditions on video title and OCR, generates a single query per pass, and constrains decoding with a trie of historical queries. In our industrial setting, this design faces three limitations: (i) a single video can trigger multiple heterogeneous intents, yet GREAT emits one query per forward pass, requiring expensive beam search or re-sampling for coverage; (ii) its edit-distance metric against a single reference query penalizes benign token-order variations and under-rewards semantically equivalent queries; and (iii) title+OCR alone often misses platform-specific context such as gaming jargon, in-app streamer relationships, or film/TV actor names.

We address these limitations from four angles, illustrated in Figure 1 (right). First, we reformulate bottom-bar suggestion as *list-level generation*: given a video v, the model is required to output a list of 20 queries in one shot, and we construct high-quality multi-query ground truth from logs by

filtering on exposure and CTR.

Second, this reformulation motivates a new offline metric. Rather than comparing each generated query to a single reference with edit distance, we design a set-to-set metric that (i) computes token F1 between predicted and ground truth queries, (ii) aligns the two lists via an optimal matching, and (iii) aggregates scores with CTR-based weights. The resulting score, CTR-weighted Hungarian F1 ($\mathrm{CTR\text{-}HungF1}$), captures token overlap, list-level coverage, and user preference in a single objective that we can use for both evaluation and training.

Third, we enrich the input beyond on-video text with a video-to-video-to-query (V2V2Q) retrieval-augmented generation (RAG) pipeline. We maintain a large pool of historical videos. Given a target video, we run Approximate Nearest Neighbor (ANN) search to retrieve K neighbors, collect their previously exposed queries, and apply a lightweight hybrid scoring and nucleus filtering step to distill them into a compact retrieved query context X_{RAG}. This provides dynamic, behavior-grounded background knowledge for each video.

Finally, we apply thinking-free reinforcement learning with verifiable rewards (RLVR) (Lambert et al., 2025), using deterministic format checks and $\mathrm{CTR\text{-}HungF1}$ as the sole reward to train a compact LLM with a GRPO-style (Shao et al., 2024) objective. The resulting model is deployed in an hourly production pipeline and has been rolled out to full traffic in Kuaishou with consistent gains on core search metrics.

This work makes four primary contributions:

- We cast video-driven query suggestion as a list-generation task and propose a CTR-weighted Hungarian F1 metric that evaluates set-to-set similarity between predicted and ground truth query lists.
- We introduce a V2V2Q RAG pipeline that retrieves semantically similar videos via an ANN index, applies a hybrid scoring scheme, and performs nuclus filtering to supply background knowledge for generation.
- We realize a RLVR training paradigm that uses structural checks and the $\mathrm{CTR\text{-}HungF1}$ score as the sole reward, enabling GRPO-style optimization of a compact LLM without reward model training or CoT distillation.
- Yielding significant improvements across key online metrics, we have deployed the model in an hourly large-scale inference system on full traffic of the Kuaishou app, generating over 10^8 queries for more than 6M videos daily.

2 Proposed Framework

We formulate bottom-bar query suggestion as list-wise generation and present a three-part framework: (i) retrieval-augmented context construction, (ii) list-level alignment with verifiable rewards, and (iii) an hourly production inference pipeline.

2.1 Task Formulation and System Overview

Given a video v, the policy model generates a fixed-size query list $\hat{Q}(v) = \{\hat{q}_1, \ldots, \hat{q}_M\}$ with M=20. We treat $\hat{Q}(v)$ as an unordered set: both training and evaluation are permutation-invariant and emphasize intent coverage over output order.

Input Representation. We build a text prompt $\mathcal{X}(v)$ by serializing $X_{\textbf{ocr}}$ (text from the cover and key frames), $X_{\textbf{meta}}$ (title, uploader name, and four hierarchical category tags) and $X_{\textbf{RAG}}$ (high-quality historical queries retrieved from previous similar videos, detailed in Section 2.2). All components are concatenated into a single natural-language prompt and fed to the policy model as plain text.

System Pipeline. We run an hourly inference pipeline to ensure timely recommendation updates. For each new video, we (i) materialize its textual signals ($X_{\text{ocr}}, X_{\text{meta}}$) and embedding, (ii) retrieve and assemble X_{RAG} from semantically similar videos, (iii) prompt the policy model to generate a 20-query list, and (iv) post-process and publish the results to the online profile asynchronously. The distributed execution and data pipeline are detailed in Section 2.5.

2.2 Context-Aware Retrieval Augmentation

While LLMs carry extensive parametric knowledge, they often miss evolving, platform-specific knowledge (e.g., gaming slang, in-app streamer relations, actor names) that is not reliably present in video titles or OCR. We therefore augment the prompt with behavior-grounded queries retrieved from semantically similar videos, as shown in the upper part of Figure 2.

Candidate Pool Construction. We construct a large candidate video pool from platform logs, keeping videos that (i) have upstream 128-d multimodal embeddings and (ii) have at least one historically exposed bottom-bar query with exposure

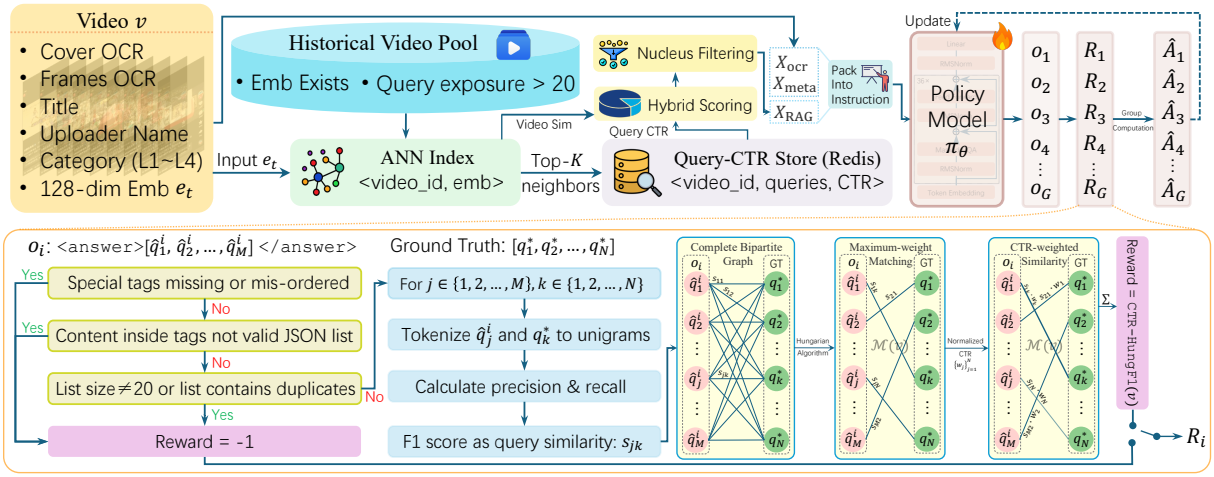

Figure 2: Overview of our framework. For each video v, we retrieve top-K similar videos from a historical pool via an ANN index. We score exposed query candidates with a hybrid video relevance–CTR function, and apply nucleus filtering to form X_{RAG}. The policy model is optimized with RLVR: malformed outputs receive a fixed penalty; otherwise the reward is $\text{CTR-HungF1}(v)$, computed by token-level F1, Hungarian matching between predicted and ground truth lists, and CTR-weighted aggregation.

> 20. This yields an vector index over ~70M videos mapping `video_id` to embeddings.

Retrieval and Context Refinement. Given a target video v with a multimodal embedding $\mathbf{e}_t$, we retrieve the top-K nearest neighbors $\mathcal{S}_v = \{v'_1, \ldots, v'_K\}$ from the candidate pool via ANN search. Each neighbor $v' \in \mathcal{S}_v$ provides a set of historically exposed queries with exposure > 20, denoted as $Q(v')$. For each occurrence of a candidate query q under a neighbor v', we compute $\text{CTR}(q, v')$ and define a pairwise retention score:

$$\text{Ret}(q, v') = \lambda \cdot \text{CosSim}(\mathbf{e}_t, \mathbf{e}_{v'}) + (1-\lambda) \cdot \text{CTR}(q, v').$$

We then assign each unique textual query a single score by taking the best supporting neighbor:

$$\text{Ret}(q) = \max_{v' \in \mathcal{S}_v} \text{Ret}(q, v').$$

Finally, we normalize $\text{Ret}(q)$ over all unique queries and apply nucleus filtering with threshold τ; the retained queries form X_{RAG}, which is injected into the prompt as a structured list of historical high-quality queries from similar videos.

2.3 Preference-Aligned List-Level Evaluation

Both the model output and supervision in our scenario are *lists* of queries: a video may trigger multiple valid search intents, and the output order is not semantically meaningful. We therefore cast evaluation as a maximum-weight bipartite matching problem, and design a permutation-invariant, CTR-aware metric to measure the set-to-set similarity between prediction and log-derived ground truth.

Query-Level Similarity. For a video v, let $\hat{Q}(v) = \{\hat{q}_1, \ldots, \hat{q}_M\}$ be the predicted query list and $Q^*(v) = \{q_1^*, \ldots, q_N^*\}$ be the ground truth set. Let $\text{Tok}(q)$ be the function that segments a query q into a sequence of tokens, and let $\mathcal{U}(q)$ be the multiset of unigrams induced by $\text{Tok}(q)$. We compute a pairwise similarity s_{ij} between $\hat{q}_i$ and q_j^* as the unigram F1: $s_{ij} = \text{F1}(\hat{q}_i, q_j^*)$, where precision and recall are defined on the overlapping token multiset between $\mathcal{U}(\hat{q}_i)$ and $\mathcal{U}(q_j^*)$.

CTR-weighted Hungarian matching. We construct a complete bipartite graph between $\hat{Q}(v)$ and $Q^*(v)$ with edge weights s_{ij}, and obtain a maximum-weight matching $\mathcal{M}(v) \subseteq \{1, \ldots, M\} \times \{1, \ldots, N\}$ via the Hungarian algorithm (Kuhn, 2010). We then compute:

$$\text{CTR-HungF1}(v) = \sum_{(i,j) \in \mathcal{M}(v)} w_j \cdot s_{ij},$$

where $w_j = \text{CTR}(q_j^*) / \sum_{k=1}^{N} \text{CTR}(q_k^*)$ is a *normalized importance weight* reflecting how strongly aggregate user behavior favors this ground truth query. We report the dataset-level performance by averaging $\text{CTR-HungF1}(v)$ over all videos.

2.4 RLVR-enabled Query Suggestion

We train a small-scale LLM π_θ to generate query lists aligned with real user preferences by applying reinforcement learning with verifiable rewards.

Reward Design For each video v, we place X_{ocr}, X_{meta}, and X_{RAG} into a fixed instruction template

to form the prompt $\mathcal{X}(v)$, and require π_θ to output a JSON-style list of exactly 20 *distinct* queries enclosed by <answer> and </answer>; prompt templates are detailed in Appendix D. We parse the tag content deterministically: if any format or parsing constraint is violated (dipicted in left-bottom of Figure 2), we assign $R = -1$; otherwise, $R = \text{CTR-HungF1}(v)$.

Policy Model Optimization. We optimize π_θ using GRPO (Shao et al., 2024) with a DAPO-style (Yu et al., 2025) token objective and asymmetric clipping. Given $\mathcal{X}$, we sample G rollouts $\{o_i\}_{i=1}^{G}$ from the behavior policy $\pi_{\theta_{\text{old}}}$ and compute the standardized advantage:

$$\hat{A}_i = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})}.$$

The training objective is:

$$\mathcal{J}(\theta) = \mathbb{E}_{(\mathcal{X},Q^*)\sim\mathcal{D},\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|\mathcal{X})} \Bigg[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \min\big(r_{i,t}(\theta)\,\hat{A}_i,\ r_{i,t}^{\text{clip}}(\theta)\,\hat{A}_i\big) - \beta\,\mathbb{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}) \Bigg],$$

where $r_{i,t}(\theta) = \dfrac{\pi_\theta(o_{i,t} \mid \mathcal{X}, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid \mathcal{X}, o_{i,<t})}$, and $r_{i,t}^{\text{clip}}(\theta) = \text{clip}\big(r_{i,t}(\theta),\ 1-\varepsilon_{\text{low}},\ 1+\varepsilon_{\text{high}}\big)$.

2.5 Online Deployment

We deploy the model within an hourly production pipeline comprising (i) a GPU inference cluster, (ii) HDFS for intermediate data exchange, and (iii) the company's Integrated Development Platform (IDP) for Hive-table management and downstream data orchestration, as illustrated in Figure 3.

Data Ingestion and Context Construction Each hour, several hundred thousand quality-filtered videos are written into an offline Hive partition with X_{ocr}, X_{meta}, and a 128-d embedding. During inference, we retrieve the top-10 neighbors via ANN, fetch their historically exposed queries and CTR stats from Redis, and assemble the retrieval-augmented context for generation.

Distributed GPU Execution Eight 4-GPU L20 machines run inference in parallel. Each node polls HDFS every minute for a new hourly partition (via _SUCCESS file), takes one eighth of the data, and can process multiple hours concurrently. To prioritize fresh data, each node has a 60-minute inference

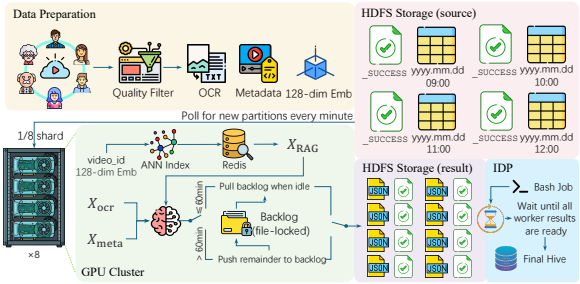

Figure 3: Online pipeline for hourly query generation.

budget: unfinished samples are appended to a per-node backlog, while idle capacity backfills older leftovers (with file locks and atomic updates). Each node uploads its finished shard to a node-specific HDFS directory and writes _SUCCESS.

Result Aggregation and Online Serving. IDP polls the eight output HDFS directories; once all _SUCCESS markers appear for an hour, it merges shards, writes the query-suggestion result Hive partition, and triggers downstream jobs to publish profiles consumed by online search. The system sustains ∼270k videos per hour (>6M/day), producing $> 10^8$ queries daily.

3 Experiments

We evaluate our framework on an offline dataset constructed from production logs of Kuaishou and a large-scale online A/B test.

3.1 Offline Dataset Construction

We build an offline dataset from production logs collected between Oct. 28 and Nov. 4 2025. Videos are sampled to match the online Level-1 category distribution; to prevent tail categories from being under-represented, we set the rarest category to N=100 videos and scale others proportionally.

Concretely, we first form a candidate pool of videos that have valid embeddings and at least 20 associated bottom-bar queries, each with impressions > 200 and CTR $> 2\%$. We then perform category-stratified sampling to follow the Level-1 proportions while meeting the per-category minimum, and split videos into train/test with a 9:1 ratio within each category. For each sampled video, all queries satisfying the same impressions/CTR filter are used as the multi-query ground truth.

3.2 Evaluation Metrics

Offline, we evaluate with the proposed metric $\text{CTR-HungF1}(v)$ and report the mean over videos.

Method	Venue	CTR-HungF1	#Train Vids	#Train Qs	Latency (s/vid) ↓
GREAT (Shao et al., 2025)	KDD '25	30.1%	439,896	419,238	2.08
Qwen3-4B-Ins-vanilla (Yang et al., 2025)	–	33.6%	–	–	0.04
Ours	–	**47.3%** (+17.2%)	9,952	238,516	**0.04** (×52)

Table 1: Comparison on the offline test set. #Train Vids and #Train Qs denote the number of videos and ground truth queries used during training, respectively. Latency is measured with single-GPU inference while generating 20 queries per video. Relative improvements over GREAT are shown in parentheses.

S-Vol	S-Dev	S-Pen	CTR^q	CTR^{rp}	PV-VV	PV-WT	LW-PV	PV-LW	AQR ↓
+0.289%	+0.254%	+0.269%	+2.50%	+0.14%	+1.01%	+1.04%	+0.31%	+1.09%	-0.54%

Table 2: Relative lifts in the online A/B test over the production baseline (metric definitions in Section 3.2). Improvements are observed on search topline (S-Vol, S-Dev, S-Pen), bottom-bar satisfaction (CTR^q, AQR), and downstream result-page engagement (CTR^{rp}, PV-VV, PV-WT, LW-PV, PV-LW).

Online, we track **Search Topline** metrics—S-Vol (total searches), S-Dev (distinct search devices), and S-Pen (search penetration rate)—as well as **Bottom-bar & Result-page** engagement metrics. We measure bottom-bar query CTR CTR^q and AQR (active query-change rate; lower is better). For the subsequent result page (each load counted as one Page View, PV), we report CTR^{rp} (average result-page CTR), PV-VV (video views per PV), PV-WT (watch time per PV), LW-PV (PV rate with $\geq$1 long-watch event), PV-LW (long-watch events per PV). Detailed definitions and measurement protocols are provided in Appendix B.

3.3 Implementation Details

We use Qwen3-4B-Instruct-2507 (Yang et al., 2025) as the base policy model and perform reinforcement training with veRL (Sheng et al., 2025) on 8 NVIDIA H800 GPUs. Retrieval uses λ=0.6 and nucleus filtering with τ=0.6; PPO-style asymmetric clipping uses $(\epsilon_{\text{low}}, \epsilon_{\text{high}})=(0.2, 0.28)$. Chinese tokenization is done with Jieba[1]. Additional details are deferred to Appendix C.

3.4 Offline Experiments

We first evaluate our approach on the offline test set described in Section 3.1, measuring alignment between generated lists and high-quality, high-CTR log queries. We compare against two baselines. (1) Qwen3-4B-Ins-2507-vanilla, the base policy model *without any training*, prompted with the same input signals as our system; (2) GREAT (Shao et al., 2025), re-implemented with the original input design and decoding strategy, and generate 20 candidates via beam search. Table 1 reports the results.

[1]https://github.com/fxsjy/jieba

Overall Issues ↓	Relevance ↑	Literal Quality ↑	Risk ↓
-2.33%	+1.67%	+0.34%	-0.67%

Table 3: Human evaluation of generated queries in the online A/B test. Risk: safety/policy issues (e.g., inappropriate content or harmful guidance for teenagers).

Our method outperforms both baselines by a large margin. Relative to the matched-input zero-shot baseline, RLVR yields substantial gains on the list-level objective, demonstrating strong generalization beyond the raw instruction-following capability of the base model. We also surpass GREAT despite using orders-of-magnitude fewer training videos, indicating a more sample-efficient learning signal from behavior-grounded rewards and retrieval context. Finally, our one-shot list generation is 50× faster than GREAT at producing 20 queries on a single GPU, avoiding the decoding overhead of beam search.

3.5 Online Experiments

We run a 7-day online A/B test on Kuaishou with 4% traffic in treatment and the production system as control. Table 2 reports relative lifts; all changes are statistically significant with p-value < 0.05.

We observe consistent improvements in both search adoption and downstream engagement. On the search topline, the system increases search activity, indicating that higher-quality bottom-bar suggestions effectively activate users' search intent in the recommendation feed. On the subsequent result page, deep-consumption metrics improve while the AQR decreases, suggesting that users are more satisfied with the suggestion and are more willing to continue exploring after entering search.

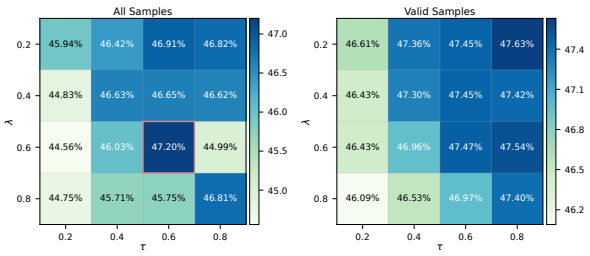

Figure 4: Hyperparameter ablation over λ and τ on the test set. Left: average reward over all test videos; Right: average reward on format-valid outputs.

X_{ocr}	X_{meta}	X_{RAG}	RL	CTR-HungF1
✓				17.00%
	✓			17.41%
✓	✓			18.94%
✓	✓	✓		33.63%
✓	✓		✓	39.28%
✓	✓	✓	✓	**47.26%**

Table 4: Ablation study on different components.

We further conduct human evaluation by sampling 300 video–query pairs from each bucket and asking professional annotators to rate the outputs. Table 3 shows improved relevance and literal quality (e.g., grammar and typos), with fewer risk-control (e.g., unsafe or inappropriate content for minors) and overall issues.

We attribute the improvements to (i) one-shot list generation that better covers the diverse intents a video can trigger, and (ii) behavior-grounded on-platform RAG context that eases the handoff to the result page and encourages sustained engagement. Taken together, the online results confirm that our framework translates offline alignment improvements into measurable production-scale benefits.

3.6 Ablation Study

We analyze the contribution of each input component and RLVR training in Table 4. Using Qwen3-4B-Instruct-2507 with only X_{ocr} or X_{meta} yields similar limited performance (17.00% vs. 17.41%). Combining them brings a modest gain (18.94%), suggesting OCR and metadata are complementary yet insufficient for platform-specific intent. Adding retrieval augmentation X_{RAG} boosts performance to 33.63%, showing that behavior-grounded queries from similar videos provide crucial context. Notably, applying RLVR without X_{RAG} already achieves 39.28%, indicating that the list-level reward signal alone substantially improves alignment with high-CTR ground truth. When both X_{RAG} and RLVR are combined, CTR-HungF1 reaches 47.26%, a further 7.98% absolute gain over RL-only, confirming that retrieval augmentation and reinforcement learning capture complementary signals and their synergy generalizes well beyond zero-shot prompting.

Figure 5: Case study on a shampoo-review video comparing ground truth high-CTR queries (GT), GREAT outputs, retrieved queries from RAG module (RQ), and our final generations (Ours). CTR-HungF1: 72.3% (Ours) vs. 32.6% (GREAT) on this example.

We further study two retrieval hyperparameters in Figure 4: λ, which controls the weight of cosine similarity in the hybrid retention score, and τ, the threshold for nucleus sampling. We set $\lambda = \tau = 0.6$, which performs best on *All Samples* and minimizes the All–Valid gap, reflecting strong quality with few format failures.

3.7 Case Studies

Figure 5 shows a representative shampoo-review video. The retrieved queries (RQ) demonstrate that our RAG pipeline recalls previously exposed, highly relevant queries. Conditioned on them, our outputs cover the major intents in the online high-CTR ground truth, including the broad intent "CCTV-recommended shampoos" and the specific product praised in the video ("SPES shampoo"). This indicates that the policy model distills valuable signals from the retrieved set rather than copying candidates. In contrast, GREAT mostly generates generic "shampoo review" queries and introduces unrelated brands absent from the video, reflecting weaker alignment with user interests.

4 Conclusion

In this paper, we propose a production-ready framework for bottom-bar query suggestion. It generates a query list with multi-query supervision, optimizes a CTR-aware set-matching objective via thinking-

free RLVR, and enriches context with a video-to-video-to-query RAG pipeline. The system achieves strong gains across offline tests and online A/B experiments and is deployed on Kuaishou at full traffic, serving hundreds of millions of users daily.

Limitations

Our performance depends on the density and freshness of behavior-grounded retrieval signals. In principle, cold-start content, rapidly emerging trends, or degraded ANN neighbors (e.g., embedding drift or index staleness) could make the V2V2Q context X_{RAG} less informative, pushing the system closer to prompt-only generation. In our production setting, however, we primarily serve high-quality, high-traffic uploads where log coverage and retrieval signals are typically sufficient, and we observe stable behavior in deployment. Nonetheless, this remains a technical dependency on retrieval quality and infrastructure.

Ethical Considerations

Training targets are derived from high-exposure, high-CTR video-related queries that have already passed multiple production filtering and policy-control pipelines. The base model (Qwen3-4B-Instruct-2507) is tuned with alignment to human values, providing a safety-aware initialization. Our RLVR uses deterministic format checks and log-derived rewards, without introducing unconstrained external supervision. In addition, human evaluation finds no risk-control issues in our sampled outputs and reports a reduction in risk-control issues for the treatment bucket (Table 3). We will continue monitoring safety metrics and enforcing platform policies during deployment to ensure responsible, user-safe query suggestions.

References

Tianchi Cai, Zhiwen Tan, Xierui Song, Tao Sun, Jiyan Jiang, Yunqi Xu, Yinger Zhang, and Jinjie Gu. 2024. Forag: Factuality-optimized retrieval augmented generation for web-enhanced long-form question answering. In *KDD*, pages 199–210.

Yapei Chang, Yekyung Kim, Michael Krumdick, Amir Zadeh, Chuan Li, Chris Tanner, and Mohit Iyyer. 2025. BLEUBERI: BLEU is a surprisingly effective reward for instruction following. In *NeurIPS*.

Jinwen Chen, Shuai Gong, Shiwen Zhang, Zheng Zhang, Yachao Zhao, Lingxiang Wang, Haibo Zhou, Yuan Zhan, Wei Lin, and Hainan Zhang. 2026. Localsug: Geography-aware llm for query suggestion in local-life services. *Preprint*, arXiv:2603.04946.

Kang Chen, Qingheng Zhang, Chengbao Lian, Yixin Ji, Xuwei Liu, Shuguang Han, Guoqiang Wu, Fei Huang, and Jufeng Chen. 2024. IPL: leveraging multimodal large language models for intelligent product listing. In *EMNLP (Industry Track)*, pages 697–711.

Cheng Cheng, Chenxing Wang, Aolin Li, Haijun Wu, Huiyun Hu, and Juyuan Wang. 2026. When & how to write for personalized demand-aware query rewriting in video search.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Xinya Du and Heng Ji. 2022. Retrieval-augmented generative question answering for event argument extraction. In *EMNLP*, pages 4649–4666.

Huawen Feng, Zekun Yao, Junhao Zheng, and Qianli Ma. 2025. Training large language models for retrieval-augmented question answering through backtracking correction. In *ICLR*.

Xian Guo, Ben Chen, Siyuan Wang, Ying Yang, Mingyue Cheng, Chenyi Lei, Yuqing Ding, and Han Li. 2026. Onesug: The unified end-to-end generative framework for e-commerce query suggestion. In *AAAI*, pages 14774–14782.

Alexander Gurung and Mirella Lapata. 2025. Learning to reason for long-form story generation. In *Second Conference on Language Modeling*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL*, pages 874–880.

Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. In *Second Conference on Language Modeling*.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan Ö. Arik. 2025. Long-context llms meet RAG: overcoming challenges for long inputs in RAG. In *ICLR*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*, pages 6769–6781.

Sunwoo Kim, Geon Lee, Kyungho Kim, Jaemin Yoo, and Kijung Shin. 2025. Itemrag: Item-based retrieval-augmented generation for llm-based recommendation. *Preprint*, arXiv:2511.15141.

Harold W. Kuhn. 2010. The hungarian method for the assignment problem. In *50 Years of Integer Programming*, pages 29–47.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *SOSP*, pages 611–626.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*.

Che Liu, Haozhe Wang, Jiazhen Pan, Zhongwei Wan, Yong Dai, Fangzhen Lin, Wenjia Bai, Daniel Rueckert, and Rossella Arcucci. 2025. Beyond distillation: Pushing the limits of medical LLM reasoning with minimalist rule-based RL. In *The Second Workshop on GenAI for Health: Potential, Trust, and Policy Compliance*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Erxue Min, Hsiu-Yuan Huang, Xihong Yang, Min Yang, Xin Jia, Yunfang Wu, Hengyi Cai, Junfeng Wang, Shuaiqiang Wang, and Dawei Yin. 2025. From prompting to alignment: A generative framework for query recommendation. *Preprint*, arXiv:2504.10208.

Jiazhen Pan, Che Liu, Junde Wu, Fenglin Liu, Jiayuan Zhu, Hongwei Bran Li, Chen Chen, Cheng Ouyang, and Daniel Rueckert. 2025. Medvlm-r1: Incentivizing medical reasoning capability of vision-language models (vlms) via reinforcement learning. In *MICCAI*, volume 15966 of *Lecture Notes in Computer Science*, pages 337–347. Springer.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

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 *NeurIPS*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Trans. Assoc. Comput. Linguistics*, 11:1316–1331.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Ninglu Shao, Jinshan Wang, Chenxu Wang, Qingbiao Li, and Xiaoxue Zang. 2025. Great: Guiding query generation with a trie for recommending related search about video at kuaishou. In *KDD*, page 4818–4826.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient RLHF framework. In *EuroSys*, pages 1279–1297. ACM.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2025. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales. In *ICLR*.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. In *NeurIPS*.

Sheng Zhang, Qianchu Liu, Guanghui Qin, Tristan Naumann, and Hoifung Poon. 2025. Med-rlvr: Emerging medical reasoning from a 3b base model via reinforcement learning. *Preprint*, arXiv:2502.19655.

Huike Zou, Haiyang Yang, Yindu Su, Liyu Chen, Chengbao Lian, Qingheng Zhang, Shuguang Han, and Jufeng Chen. 2025. Multi-value-product retrieval-augmented generation for industrial product attribute value identification. In *EMNLP (Industry Track)*, pages 2096–2105.

A Related Work

Retrieval-Augmented Generation Retrieval-augmented generation (RAG) (Lewis et al., 2020)

mitigates hallucination and knowledge staleness by coupling parametric models with non-parametric retrieval. Early work (Izacard and Grave, 2021; Du and Ji, 2022; Ram et al., 2023) typically retrieves documents via sparse or dense methods such as BM25 (Robertson and Zaragoza, 2009) and DPR (Karpukhin et al., 2020) and directly appends them to the prompt, while later systems (Wei et al., 2025; Jin et al., 2025; Feng et al., 2025) treat retrieval as a pluggable component and focus on improving generation under retrieved context for knowledge-intensive tasks. In industrial scenarios, IPL (Chen et al., 2024) adapts a multimodal LLM with RAG over visually similar products to improve C2C listing generation, and MVP-RAG (Zou et al., 2025) retrieves similar products and candidate attribute values for robust attribute value generation at Xianyu scale. Beyond e-commerce, FoRAG (Cai et al., 2024) studies web-enhanced long-form QA with an outline-aware generator and RLHF, while ItemRAG (Kim et al., 2025) performs item-based retrieval and summarization to enhance LLM-based recommendation. Our work follows this RAG philosophy in a new setting: we retrieve similar videos via a dense embedding index and aggregate their historical related queries as external knowledge, forming a video-to-video-to-query bottom-bar search term paradigm.

LLM-based Query Suggestion and Rewriting Recent work has applied preference alignment to LLM-based query generation. GQR (Min et al., 2025) uses a CTR predictor with listwise DPO for conversational query recommendation; OneSug (Guo et al., 2026) adopts behavior-weighted DPO (Rafailov et al., 2023) for e-commerce query suggestion; WeWrite (Cheng et al., 2026) and LocalSUG (Chen et al., 2026) apply GRPO with CTR-based rewards to query rewriting and suggestion; and DeepRetrieval (Jiang et al., 2025) trains LLMs via RL with Recall@K rewards for query augmentation in information retrieval. However, these methods all condition on textual queries or prefixes and optimize per-query objectives via DPO or learned reward models. In contrast, our work generates queries from short-video signals, a cross-modal setting absent from prior work, and introduces a permutation-invariant set-level reward that jointly captures coverage and preference alignment without any learned reward model, coupled with a video-to-video-to-query RAG pipeline that supplies behavior-grounded context.

RLVR Reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; Lambert et al., 2025; DeepSeek-AI et al., 2025; Yu et al., 2025) optimizes simple deterministic verifiers of answer correctness or format instead of learned reward models, and has become a core recipe for post-training reasoning-oriented LLMs on math and coding benchmarks. Beyond STEM domains, RLVR-style methods have been applied to more open-ended language and medical tasks. BLEUBERI (Chang et al., 2025) treats BLEU (Papineni et al., 2002) as a verifiable reward and uses GRPO (Shao et al., 2024) to align instruction-following models. ReasoningNCP (Gurung and Lapata, 2025) constructs long-form story generation as Next-Chapter Prediction and uses VR-CLI, a reward based on likelihood improvement of gold continuations, to train reasoning traces with GRPO. In the medical domain, Med-RLVR (Zhang et al., 2025), AlphaMed (Liu et al., 2025), and MedVLM-R1 (Pan et al., 2025) all leverage simple, automatically checkable rewards on multiple-choice QA or VQA labels to elicit emergent clinical reasoning and obtain strong out-of-distribution gains, without relying on distillation from closed-source teachers. In our setting, the model outputs an unordered set of related queries per video; we cast this list-generation task as RLVR with a deterministic set-to-set reward and train solely on this verifiable signal, without any reward model or chain-of-thought distillation, yet still obtain substantial offline and online improvements in search engagement.

B Evaluation Metrics

Our primary offline metric is the CTR-weighted Hungarian F1 score $\mathrm{CTR\text{-}HungF1}(v)$ introduced in Section 2.3; for a given evaluation set, we report the average $\mathrm{CTR\text{-}HungF1}$ over all videos in the set. Online, we monitor a set of core business metrics that capture both overall search activity and the effectiveness of the bottom-bar entry.

Overall search metrics. At the search level, we track three aggregate indicators: (i) *S-Vol*, the total number of searches; (ii) *S-Dev*, the number of distinct search devices; and (iii) *S-Pen*, the search penetration rate. These metrics measure how much the system expands search volume and how widely search is adopted across the user base.

Bottom-bar and result-page engagement. For the bottom-bar entry itself, we first measure the

CTR of generated queries (CTR^q). Since tapping a suggestion opens a search result page, we then examine user behavior on these result pages (each page load counted as one page view, PV).

On the result page, we track: (i) the *result-page CTR* (CTR^{rp}), defined as the average CTR on search result pages; (ii) *PV-VV*, the average number of video views per PV, measuring how many result-page videos a user typically watches after clicking a bottom-bar query; and (iii) *PV-WT*, the average watch time per PV, i.e., the total dwell time on the result page averaged over PVs.

To capture deeper consumption, we additionally monitor: (iv) *LW-PV*, the long-watch PV rate, defined as the fraction of result-page PVs in which at least one video reaches a long-watch threshold; and (v) *PV-LW*, the average number of long-watch events per PV, which counts, on average, how many videos per result-page achieve long-watch status (e.g., if one PV has 1 long-watch video and another has 5, this metric would be 3). Finally, we track (vi) *AQR*, the active query-change rate, i.e., the proportion of users actively replace the suggested query with a new one, indicating dissatisfaction with the original bottom-bar suggestion. Together, these metrics characterize not only whether bottom-bar suggestions are clicked, but also how much substantive and satisfactory consumption they induce on the downstream search result pages.

C Implementation Details

We implement our reinforcement learning pipeline using the veRL (Sheng et al., 2025) framework with a GRPO-style advantage estimator. The base policy model is Qwen3-4B-Instruct-2507 (Yang et al., 2025). Training is performed for four epochs on a single node with 8 NVIDIA H800 GPUs. We use a per-step training batch size of 256 and sample 16 rollouts per prompt during training. The policy is optimized with AdamW (Loshchilov and Hutter, 2019) using a peak learning rate of 8×10^{-6}. Learning rate warmup is applied for the first 10% steps, followed by cosine decay to a minimum learning rate of $0.2\times$ the peak value. PPO-style clipping is adopted with asymmetric clip ratios $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$ to stabilize policy updates.

For retrieval augmentation, the retention score for each (q, v') pair is computed by combining cosine similarity and historical CTR, with cosine similarity weighted by $\lambda = 0.6$. Candidate queries are further filtered via nucleus sampling with threshold $\tau = 0.6$. For rollout, we adopt a temperature of 1.3 and employ the vLLM (Kwon et al., 2023) backend with tensor parallelism of 4 and a GPU memory utilization cap of 0.8.

All custom rewards—including format validation and the proposed CTR-HungF1 metric—are implemented as deterministic functions and computed in batch mode. For query-level similarity computation, Chinese text is tokenized using the Jieba toolkit[2].

D Prompt Templates

System Prompt

You are a professional video content analysis assistant. Your task is to predict the search queries that a user might issue after watching a given video, based on the video's title, OCR text, publisher name, video category information, and a provided list of related search terms. You should think from the perspective of a user who has just watched the video.
Please follow the requirements below:

1. Carefully analyze the video content and extract key information points.
2. The related search term list comes from search queries associated with videos similar to the current one, which may contain valuable reference information. For entries that are worth referencing, assign a higher attention weight.
3. The generated queries should cover both the core content of the video and relevant extended topics.
4. The queries should be concise and clear, conforming to real users' search habits.
5. The output format must be a simple JSON-style list (strictly comply), in the following form:

```
[
  "related search term 1",
  "related search term 2",
  ...
```

[2] https://github.com/fxsjy/jieba

```
]
```

6. Output the final list answer within <answer> and </answer> tags. The content must strictly follow the above list format, containing only string items without any extraneous text.

Now, please generate the search queries that users might search based on the following video information and the related search term list (if available):

User Prompt Template

Current video information:

```
{
    "Title": "...",
    "Cover OCR": "...",
    "All Frames OCR": "...",
    "Publisher Name": "...",
    "Level-1 Category": "...",
    "Level-2 Category": "...",
    "Level-3 Category": "...",
    "Level-4 Category": "..."
}
```

Related search term list (for entries relevant to the current video, please assign a higher attention weight):
`["search term A", "search term B", ...]`

Please generate the search queries that users might search based on the video content and by referencing the related search terms (if available). If any text field of the video information is empty, please ignore that field.

Note: Please strictly output exactly 20 queries this time. No two or more queries should be completely identical, and no empty strings are allowed. Only provide the final answer list within <answer> and </answer> tags, strictly using the following format with exactly 20 items:

```
<answer>[
  "related search term 1",
  "related search term 2",
  ...
]</answer>
```