

# RACA-PODS: Reasoning-Informed First-Token Prediction for Query-Document Relevance in WeChat Search

Yupeng Hu<sup>2</sup> Dongliang Liao<sup>1\*</sup> Changxing Ding<sup>1</sup> Mengchuan Qiu<sup>2</sup>  
Huasheng Liang<sup>2</sup> Rolan Yan<sup>2</sup> Junhong Wang<sup>1</sup>

<sup>1</sup>South China University of Technology, Guangzhou, Guangdong, China

<sup>2</sup>Tencent WeChat Business Group, Guangzhou, Guangdong, China

winstonhu@tencent.com, liaodl@scut.edu.cn, chxding@scut.edu.cn

mecthewqiu@tencent.com, watsonliang@tencent.com,

rolanyan@tencent.com eewjh@mail.scut.edu.cn

## Abstract

Applying LLMs to query-document relevance estimation in industrial document retrieval faces two key challenges: the complexity and heterogeneity of Web content, and stringent latency and resource constraints. To address them, we propose an automated framework with two components: **Reflection-Aware Collaborative Agent (RACA)** and **Progressive Optimization Distillation Strategy (PODS)**. RACA uses collaborative reasoning and reflection to produce relevance labels and structured rationales. PODS stratifies training data by difficulty and combines supervised fine-tuning with Reasoning-Informed GRPO. Rationale supervision improves the student’s first-token relevance decision, while deployment stops after that token and avoids explicit rationale generation. Together, these mechanisms enable reasoning-informed first-token prediction that preserves the benefits of structured rationale supervision under stringent latency and resource constraints. Extensive offline experiments and a 10-day online A/B test in WeChat Search demonstrate the resulting accuracy–efficiency trade-off.

## 1 Introduction

Determining query-document relevance is a core problem in search. While lexical methods such as BM25 (Robertson and Walker, 1994) rely on term matching, neural retrievers and rerankers improve relevance estimation by modeling semantic interactions between queries and documents (Bromley et al., 1993; Huang et al., 2013; Nogueira et al., 2019). Recently, Large Language Models (LLMs) (OpenAI, 2023; Touvron et al., 2023) have advanced the field: they use direct-rank (Lu et al., 2026), step-by-step reasoning (Wu et al., 2025; Liu et al., 2025b), and iterative refinement (Kong et al., 2025; Liu et al., 2025a), allowing models to reason about relevance beyond simple semantic similarity.

\*Corresponding author.

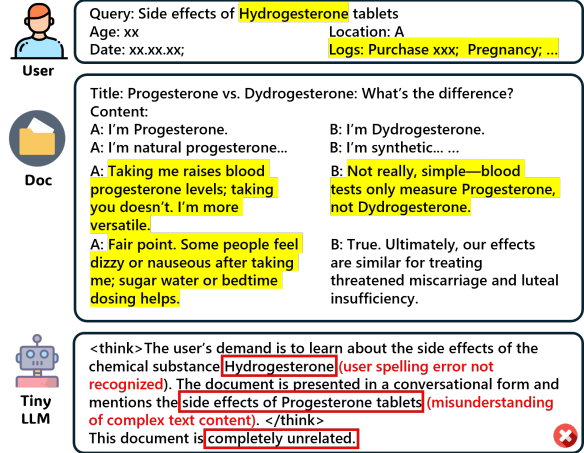


Figure 1: Challenges in the application of Tiny-LLM, which struggles to handle the complexity and heterogeneity of web content.

However, their reasoning ability is difficult to fully exploit in large-scale industrial search, where two critical gaps arise:

**Challenge 1: Complexity and Heterogeneity of Web Content.** WeChat Search must process heterogeneous long-form content, including professional and user-generated text as well as noisy OCR/ASR transcripts from multimodal sources, as shown in Fig.1. Such inputs require fine-grained intent understanding, evidence selection, and reliability assessment, which are difficult to capture with simple step-by-step prompting or off-the-shelf agentic workflows.

**Challenge 2: Stringent Latency and Resource Constraints.** Explicit LLM reasoning typically relies on large models and long test-time computation, whereas industrial search requires massive concurrency and end-to-end latency within a few hundred milliseconds. As a result, deployable relevance models must remain compact, often below 7B parameters, making long-form CoT reasoning impractical and motivating direct first-token prediction informed by rationales during training.

Together, these challenges create a bottleneck: the cases that most need sophisticated reasoning are precisely those where production constraints make explicit reasoning least affordable. To address this gap, we propose an automated framework that transfers high-level agentic intelligence into ultra-efficient models through two components. (1) **RACA (Reflection-Aware Collaborative Agent)** targets *Challenge 1* by replacing simple step-by-step prompting with an agentic reasoning process that evaluates relevance from three complementary perspectives: Intent Disambiguation, Content Salience, and Reliability. With Reflection and Refinement stages, the LLM critiques and revises its reasoning, producing more accurate, structured, and interpretable relevance rationales. (2) **PODS (Progressive Optimization Distillation Strategy)** addresses *Challenge 2* through curriculum learning. Since small models struggle to learn from complex self-corrective paths, PODS begins with simpler examples and gradually increases difficulty while alternating Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL). During SFT, a *label-first* format requires the relevance grade before the explanation. During RL, specialized rewards and resampling encourage agreement among the first-token grade, the generated rationale, and the rating guideline. At deployment, the explanation is not generated; the first-token scorer therefore benefits from rationale-guided training while retaining the low latency required for industrial serving.

In summary, our primary contributions are as follows:

- We propose RACA, a multi-dimensional agentic framework for IR through reflection-refinement loops. It generates structured reasoning paths that capture complex semantic nuances more effectively than traditional CoT.
- We introduce PODS, a progressive optimization strategy that uses RACA rationales to improve first-token relevance prediction in Tiny-LLMs via curriculum learning, enabling low-latency scoring with models as small as 1.7B parameters.
- We validate RACA-PODS through offline experiments on public and industrial datasets and online A/B testing in WeChat Search, including public component and label-noise controls under real production constraints.

## 2 Related Work

### 2.1 Query-Document Matching

Query-document matching has progressed from lexical matching to neural and LLM-based ranking. BM25 (Robertson and Walker, 1994) relies on exact term matching and suffers from vocabulary mismatch, which neural IR addresses through dual-tower representation models (Huang et al., 2013) and interaction-based architectures (Hu et al., 2014; Guo et al., 2016; Xiong et al., 2017). PLM-based Cross-Encoders further improve ranking precision via full query-document interaction (Nogueira et al., 2019; Sun et al., 2019; Gong et al., 2020; Bromley et al., 1993), but their computational cost limits large-scale deployment.

LLMs extend ranking with generative and instruction-following abilities. Prompting and fine-tuning methods such as RankGPT (Sun et al., 2023) and RankLlama (Ma et al., 2024) align LLMs with listwise ranking, while CoT and agentic methods decompose relevance or preference reasoning into intermediate steps (Wu et al., 2025; Kong et al., 2025; Zhang et al., 2024; Bougie and Watanabe, 2025). However, explicit multi-step reasoning and large model sizes are unsuitable for high-concurrency industrial search. In addition, agentic logic for user-item recommendation cannot be directly transferred to query-document matching, where relevance requires fine-grained intent, content salience, and reliability assessment. We therefore use agentic relevance rationales as training supervision for efficient Tiny-LLMs.

### 2.2 Distillation of LLMs and Agentic Reasoning

LLM distillation aims to transfer large-model capabilities to smaller models through structural alignment, representation matching, and data-centric supervision. Representative methods include Block-Level Local Distillation (Bercovich et al., 2025), CLID/MSE-based multi-level alignment (Dong et al., 2025), rationale generation (Hsieh et al., 2023), and adversarial imitation-discrimination frameworks such as LION (Jiang et al., 2023). Yet these methods do not fully remove the latency of explicit reasoning. Inspired by Implicit Chain-of-Thought (Wei et al., 2022), our work combines curriculum learning, SFT, and RL so that agent-generated rationales improve the first-token decisions of 1B–3B models without verbose inference-time reasoning.

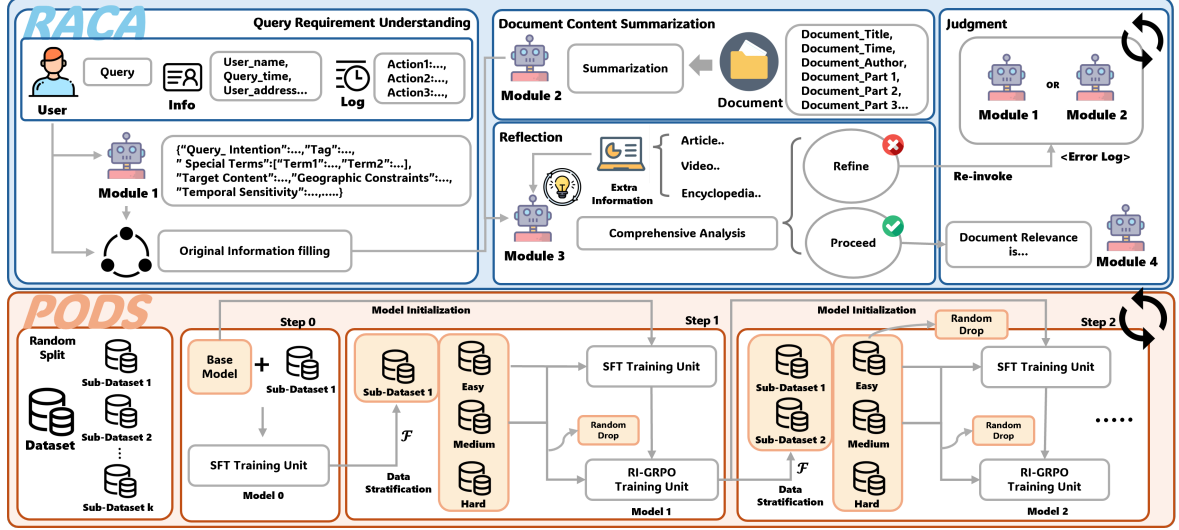


Figure 2: The Framework of RACA-PODS.

### 3 Method

**Task formulation.** We formulate query-document matching as point-wise ordinal relevance grading. Let  $\mathcal{Q}$ ,  $\mathcal{D}$ ,  $\mathcal{C}$ , and  $\mathcal{H}$  denote the query, document, search-context, and user-history spaces, respectively. Given an input tuple  $\mathbf{x} = (q, d, c, h)$ , where  $q \in \mathcal{Q}$ ,  $d \in \mathcal{D}$ ,  $c \in \mathcal{C}$ , and  $h \in \mathcal{H}$ , the model predicts a grade

$$y = f(q, d, c, h; \theta), \quad y \in \mathcal{Y}. \quad (1)$$

where  $\theta$  denotes model parameters and  $\mathcal{Y} = \{c_0, c_1, \dots, c_K\}$  is an ordered relevance set ranging from *Irrelevant* ( $c_0$ ) to *Highly Relevant* ( $c_K$ ).

#### 3.1 Reflection-Aware Collaborative Agent

RACA uses a high-capacity agentic LLM teacher to produce robust relevance labels and structured rationales for complex web content. Instead of relying on a single chain-of-thought prompt, RACA decomposes relevance assessment into three axes required by industrial search: intent disambiguation, content salience, and reliability. It consists of query requirement understanding, document summarization, reflection, and final judgment.

**Query and document analysis.** The query requirement understanding module converts the raw query, context, and user history into a structured intent representation:

$$\hat{q} = \pi_q(q, c, h, \mathcal{E}; \theta), \quad \hat{q} = \{(k_i, v_i, e_i)\}_{i=1}^m, \quad (2)$$

where  $\pi_q$  is the query analysis function,  $\mathcal{E}$  is the reflection feedback set initialized as  $\emptyset$ ,  $k_i$  is a requirement type,  $v_i$  is the extracted constraint, and

$e_i \subseteq \{q, c, h\}$  is the supporting evidence. In parallel, the document summarization module generates a compact salience-preserving summary  $\hat{d}$ :

$$\hat{d} = \pi_d(d, \mathcal{E}; \theta), \quad (3)$$

where  $\pi_d$  filters noisy or auxiliary content, such as OCR/ASR artifacts, while retaining information relevant to search judgment.

**Reflection and judgment.** To mitigate ambiguity and noise in the intermediate reasoning, RACA first audits the fused state  $\{\hat{q}, \hat{d}, \mathcal{A}\}$ , where  $\mathcal{A}$  denotes auxiliary signals from external knowledge and search-system layers:

$$\begin{aligned} [a_t, \mathcal{E}, \mathcal{R}] &= \pi_r(\hat{q}, \hat{d}, \mathcal{A}; \theta), \\ [\hat{y}, \mathcal{J}] &= \pi_J(\mathcal{R}; \theta), \quad a_t = \textit{Proceed}. \end{aligned} \quad (4)$$

The reflection policy outputs an action  $a_t \in \{\textit{Refine}, \textit{Proceed}\}$ , diagnostic feedback  $\mathcal{E}$ , and a reasoning state  $\mathcal{R}$ . If  $a_t = \textit{Refine}$ ,  $\mathcal{E}$  is fed back to the relevant upstream analyzer, e.g.,  $\pi_q$  or  $\pi_d$ , to revise  $\hat{q}$  or  $\hat{d}$ . Once  $a_t = \textit{Proceed}$ ,  $\mathcal{R}$  is passed to the judgment module  $\pi_J$ , which predicts the teacher relevance grade  $\hat{y}$  and rationale  $\mathcal{J}$ . This reflection-before-judgment loop reduces early ambiguity and noise, yielding a more reliable supervision signal for distillation into a compact model.

#### 3.2 Progressive Optimization Distillation Strategy

PODS uses RACA’s labels and rationales to train a Tiny-LLM for direct relevance scoring at inference time. It has three components: label-first

agentic data construction, curriculum-based progressive training, and Reasoning-Informed GRPO (RI-GRPO).

**Agentic data construction.** For each training instance, RACA produces a relevance grade and a structured rationale. We use a *label-first* format:

$$\begin{aligned} \text{Input} &= \langle q, c, d, h \rangle, \\ \text{Answer} &= \langle \hat{y}, \text{Reason: } \hat{q}, \hat{d}, \mathcal{J} \rangle. \end{aligned} \quad (5)$$

The grade  $\hat{y}$  appears before the explanation so that supervised fine-tuning teaches the student model a direct-scoring pattern. During deployment, generation can stop after the first grade token, while ensuring the weight updates were guided by the RACA’s reasoning path during training.

**Curriculum-based progressive training.** PODS progressively exposes Tiny-LLMs to RACA’s long self-corrective rationales, whose complexity makes one-stage distillation unstable for a limited-capacity student. We split the RACA-annotated corpus  $\mathcal{D}$  into  $\eta$  shards. Starting from a base model  $\mathcal{M}$ , we first obtain  $\mathcal{M}_0$  by cold-start SFT on  $\mathcal{D}_1$ . Each subsequent round expands the curriculum with a new shard and uses the current student to re-stratify the accumulated data into *Easy*, *Medium*, and *Hard* subsets. Optimization follows two stages: SFT on Easy/Medium samples for knowledge acquisition, followed by RI-GRPO on informative yet learnable samples, mainly from the Medium subset, to refine first-token decision boundaries under rationale supervision. To avoid overfitting and ensure continuous exploration, PODS randomly drops a portion of Easy samples with ratios  $k_1$  and  $k_2$ , while postponing high-risk samples for later iterations. Full pseudocode is provided in Appendix A.

We stratify samples by combining objective label error with student-model acceptability. For ordinal relevance grades, the logical error is  $e = |\hat{y} - y|$ , where  $\hat{y}$  is the teacher label and  $y$  is the student’s predicted grade. For non-ordinal tasks,  $e$  can be replaced by the confidence error  $1 - S_{\hat{y}}$ , where  $S = \sigma(\mathcal{P}_{\text{logits}})$  is the softmax distribution over candidate grade tokens and  $S_{\hat{y}}$  denotes the probability assigned to the teacher label. We estimate acceptability using  $\mathcal{T} = \mathcal{L} \cdot \mathcal{G}$  as an inverse proxy, where larger values indicate lower acceptability to the current student.  $\mathcal{L}$  is the normalized training loss and  $\mathcal{G}$  is the normalized  $L_2$  gradient norm. Let  $p \in [0, 1]$  be the percentile rank of  $\mathcal{T}$  within the current batch; high- $p$  samples are less acceptable

to the current student. For ordinal grades, the final stratification function is formally defined as:

$$\mathcal{F}(e, p) = \begin{cases} \text{Easy}, & e = 0, p \leq 1 - \gamma, \\ \text{Medium}, & e = 1, p \leq 1 - \gamma, \\ \text{Hard}, & \text{otherwise.} \end{cases} \quad (6)$$

where  $\gamma$  controls the fraction of high-risk samples postponed as Hard. For non-ordinal tasks, the discrete conditions on  $e$  can be replaced with task-specific confidence intervals.

**Reasoning-Informed GRPO.** RI-GRPO uses group-wise policy optimization with a deliberately structured reward. For each training instance, the student samples  $n$  responses, each encouraged to predict the relevance grade in the first token while keeping its rationale aligned with teacher reasoning and rating guidelines. The scalar reward combines answer priority, CoT alignment, and score-rationale consistency:

$$\begin{aligned} R_1 &= \mathbb{I}[s_{\text{pred}} = s_{\text{true}}], \\ R_2 &= \beta \max \left( 0, \frac{\text{sim}(\text{CoT}_{\text{pred}}, \text{CoT}_{\text{gt}}) - \alpha}{1 - \alpha} \right), \\ R_3 &= \mathbb{I}[\exists v \in \mathcal{V}_{\hat{y}} : v \text{ is satisfied by CoT}], \\ R &= GR_1 + (1 - GR_1)R_2 + R_3. \end{aligned} \quad (7)$$

Here  $s_{\text{pred}}$  and  $s_{\text{true}}$  denote the student grade decoded from the first token and the teacher grade.  $R_1$  rewards their agreement. For  $R_2$ , generated and teacher rationales are encoded with Qwen3-Embedding-4B,  $L_2$ -normalized, and compared by cosine similarity; values above  $\alpha$  are linearly rescaled into  $[0, \beta]$ .  $R_3$  checks grade-specific regular-expression markers derived from the production guideline. For example, a grade-1 rationale must identify the document subject, the user’s core demand, and a severe mismatch; higher grades specify required demand coverage and permissible missing constraints. If all  $n$  responses miss the teacher grade, we append the teacher grade to the prompt and continue generation, setting  $G = 0$  to disable  $R_1$ ; otherwise  $G = 1$ . Thus, CoT alignment and consistency still supervise resampled continuations.

After training, the Tiny-LLM outputs only the relevance grade at inference time. Thus, RACA provides agentic supervision, while PODS yields a low-latency scorer whose first-token prediction was trained with that supervision.

Table 1: Performance comparison on IQDR-HA and T2Ranking datasets.

| Method                             | IQDR-HA      |              |              |              |              |              | T2Ranking    |              |              |              |              |
|------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                                    | Acc@C1       | Acc@C2       | Acc@C3       | Acc@All      | F1           | QWK          | Acc@C1       | Acc@C2       | Acc@All      | F1           | QWK          |
| <i>Discriminative Models</i>       |              |              |              |              |              |              |              |              |              |              |              |
| Baseline (BERT-based)              | 67.5%        | 72.0%        | 66.3%        | 68.7%        | 0.687        | 0.609        | 52.8%        | 45.1%        | 49.3%        | 0.489        | −0.021       |
| <i>LLM-based Models</i>            |              |              |              |              |              |              |              |              |              |              |              |
| DeepSeek-V3 (671B)                 | 80.9%        | 86.0%        | 76.3%        | 81.3%        | 0.811        | 0.762        | 53.1%        | 61.9%        | 57.1%        | 0.571        | 0.148        |
| Qwen3-235B w/o thinking            | 80.0%        | 83.0%        | 78.8%        | 80.7%        | 0.805        | 0.775        | 51.4%        | 62.7%        | 56.6%        | 0.566        | 0.139        |
| Q3-Reranker-8B                     | 48.3%        | 49.0%        | 52.5%        | 49.7%        | 0.496        | 0.339        | 50.4%        | 46.8%        | 48.7%        | 0.486        | -0.028       |
| Vanilla SFT Q3-Reranker-8B         | 80.0%        | 77.0%        | 81.3%        | 79.3%        | 0.792        | 0.762        | 50.9%        | 60.3%        | 55.2%        | 0.552        | 0.110        |
| Vanilla SFT Qwen3-8B               | 78.3%        | 76.0%        | 80.0%        | 78.0%        | 0.778        | 0.722        | 50.7%        | 60.2%        | 55.1%        | 0.551        | 0.107        |
| Vanilla SFT Qwen3-1.7B             | 77.5%        | 75.0%        | 77.5%        | 76.7%        | 0.764        | 0.711        | 50.2%        | 60.1%        | 54.8%        | 0.547        | 0.101        |
| <i>Reason-based Models</i>         |              |              |              |              |              |              |              |              |              |              |              |
| DeepSeek-R1 (671B)                 | 85.8%        | 87.0%        | 90.0%        | 87.3%        | 0.873        | 0.855        | 57.4%        | 68.5%        | 62.5%        | 0.625        | 0.255        |
| Qwen3-235B-Thinking                | 85.0%        | 86.0%        | 86.3%        | 85.7%        | 0.856        | 0.826        | 55.1%        | 67.7%        | 60.9%        | 0.609        | 0.224        |
| GPT-4.1                            | 85.8%        | 86.0%        | 88.8%        | 86.7%        | 0.867        | 0.850        | 57.3%        | 67.8%        | 62.1%        | 0.621        | 0.248        |
| RACA w/o Reflection                | 85.0%        | 87.0%        | 90.0%        | 87.0%        | 0.871        | 0.867        | 55.9%        | 68.7%        | 61.8%        | 0.617        | 0.242        |
| <b>RACA</b>                        | <b>90.8%</b> | <b>92.0%</b> | <b>92.5%</b> | <b>91.7%</b> | <b>0.916</b> | <b>0.912</b> | 58.8%        | <b>71.2%</b> | 64.5%        | 0.645        | 0.295        |
| <i>Reason Distillation Methods</i> |              |              |              |              |              |              |              |              |              |              |              |
| RACA-SFT Qwen3-1.7B                | 78.3%        | 76.0%        | 81.3%        | 78.3%        | 0.783        | 0.732        | 50.1%        | 61.2%        | 55.1%        | 0.552        | 0.111        |
| L*-RACA-SFT Qwen3-1.7B             | —            | —            | —            | —            | —            | —            | 62.3%        | 65.6%        | 63.8%        | 0.638        | 0.277        |
| <b>RACA-PODS Q3-1.7B</b>           | 84.2%        | 85.0%        | 88.8%        | 85.7%        | 0.856        | 0.850        | 54.6%        | 67.5%        | 60.5%        | 0.605        | 0.217        |
| <b>L*-RACA-PODS Q3-1.7B</b>        | —            | —            | —            | —            | —            | —            | <b>65.8%</b> | 70.1%        | <b>67.8%</b> | <b>0.677</b> | <b>0.356</b> |

**Note:** On T2Ranking, Vanilla SFT is trained only with ground-truth labels,  $\langle q, d \rangle \rightarrow y$ ; on IQDR-HA, it uses pseudo-labels,  $\langle q, d \rangle \rightarrow \hat{y}$ . RACA-SFT/PODS additionally use RACA rationales. L\* variants replace only the pseudo-label  $\hat{y}$  with  $y$  for T2Ranking training.

## 4 Experiment

In this section, we conduct extensive experiments to evaluate our proposed RACA-PODS framework.

### 4.1 Datasets

We evaluate RACA-PODS using an industrial dataset and an open-source IR benchmark.

**Industrial Query-Document Relevance (IQDR) Dataset.** IQDR is constructed from WeChat Search. Search R&D and product specialists built a frozen pool of 3,000 query-document pairs under predefined coverage criteria: query intent, timeliness/address requirements, information carrier, content-noise level, and relevance grade. Before any system was evaluated, 300 pairs were uniformly sampled, giving 120/99/81 *Low/Moderate/High* cases. Two relevance experts independently graded every pair while blind to model predictions, rationales, and method identities; exact agreement before adjudication was 90%, and disagreements were resolved by consensus. IQDR-HA was not used for training, hyperparameter selection, or model selection. IQDR-RACA contains 124,166 RACA-annotated training pairs and a separate 15,406-pair validation set used to select  $\alpha, \beta, \gamma$ , and  $\eta$ .

**T2Ranking.** We additionally sample 40,032 training and 9,968 test pairs from the public

T2Ranking benchmark (Xie et al., 2023). We formulate binary pointwise classification by mapping grade 0 to *Irrelevant* and grades 1–3 to *Relevant*.

### 4.2 Evaluation Setup

RACA uses DeepSeek-R1-0528 (Guo et al., 2025) with deterministic decoding to generate structured reasoning paths. PODS distills these paths into Qwen3 student backbones of different sizes, with Qwen3-1.7B as the main deployable model. All inference uses temperature 0 and a label-first output format. We report accuracy, Macro-F1, and Quadratic Weighted Kappa (QWK). Baselines include BERT-based discriminative models, zero-shot and task-tuned LLMs, Qwen3-Reranker-8B, explicit reasoning LLMs, RACA ablations, and reason-distillation variants. More details, including the release scope, are provided in Appendix B.

### 4.3 Performance of Query-Doc Matching

Table 1 shows that RACA reaches 91.7% Acc@All against expert-adjudicated IQDR-HA labels (95% Wilson interval [88.0, 94.3]), close in magnitude to the 90% pre-adjudication human-human exact agreement. Its 4.4-point advantage over DeepSeek-R1 is numerical but not significant under the conservative two-proportion test ( $p = 0.079$ ). The deployable RACA-PODS Qwen3-1.7B reaches 85.7%

Table 2: Comparison of reason distillation methods on the IQDR-RACA dataset.

| Method                 | Qwen3-8B     |              |              |              |              |              | Qwen3-1.7B   |              |              |              |              |              |
|------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                        | Acc@C1       | Acc@C2       | Acc@C3       | Acc@All      | F1           | QWK          | Acc@C1       | Acc@C2       | Acc@C3       | Acc@All      | F1           | QWK          |
| Vanilla SFT            | 76.8%        | 66.5%        | 78.6%        | 74.6%        | 0.739        | 0.765        | 73.2%        | 61.8%        | 76.1%        | 71.1%        | 0.704        | 0.730        |
| Curriculum SFT         | 86.4%        | 73.8%        | 74.6%        | 78.8%        | 0.781        | 0.788        | 82.4%        | 72.3%        | 72.2%        | 76.3%        | 0.761        | 0.771        |
| SFT + Standard GRPO    | 87.6%        | 72.1%        | 79.2%        | 80.4%        | 0.796        | 0.816        | 86.3%        | 71.1%        | 77.5%        | 78.8%        | 0.779        | 0.797        |
| PODS w/o label error   | 85.8%        | 70.4%        | 77.9%        | 78.9%        | 0.780        | 0.791        | 85.2%        | 68.4%        | 76.5%        | 77.6%        | 0.766        | 0.779        |
| PODS w/o acceptability | 88.6%        | 71.2%        | 78.4%        | 80.3%        | 0.793        | 0.810        | 87.6%        | 69.9%        | 75.7%        | 78.6%        | 0.776        | 0.791        |
| <b>PODS</b>            | <b>92.9%</b> | <b>74.8%</b> | <b>80.5%</b> | <b>83.6%</b> | <b>0.827</b> | <b>0.855</b> | <b>90.3%</b> | <b>73.6%</b> | <b>79.6%</b> | <b>82.0%</b> | <b>0.811</b> | <b>0.838</b> |

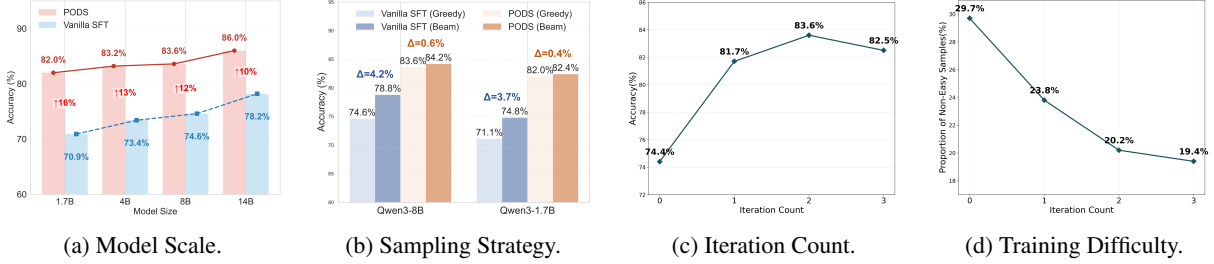


Figure 3: Further Analysis of PODS.

Table 3: PODS component decomposition on T2Ranking with Qwen3-1.7B. All rows use the same split, supervision format, and training configuration; only the named component changes.

| Method                 | Acc@All      | QWK          |
|------------------------|--------------|--------------|
| RACA-SFT               | 55.1%        | 0.111        |
| Curriculum SFT         | 56.4%        | 0.136        |
| SFT + Standard GRPO    | 57.9%        | 0.157        |
| PODS w/o label error   | 59.7%        | 0.164        |
| PODS w/o acceptability | 58.1%        | 0.170        |
| PODS w/o consistency   | 58.5%        | 0.161        |
| <b>RACA-PODS</b>       | <b>60.5%</b> | <b>0.217</b> |

Table 4: Controlled training-label noise on T2Ranking with Qwen3-1.7B. Training grades are corrupted while evaluation remains fixed.

| Noise | L*-SFT | L*-PODS | $\Delta$ |
|-------|--------|---------|----------|
| 0%    | 63.8%  | 67.8%   | +4.0     |
| 20%   | 61.1%  | 65.9%   | +4.8     |
| 40%   | 56.3%  | 62.7%   | +6.4     |

[81.3, 89.2]; it exceeds the scale-matched RACA-SFT and Vanilla-SFT students ( $p = 0.018$  and  $p = 0.005$ ) and the BERT control ( $p < 10^{-6}$ ). Among 30 sampled teacher hallucinations, reflection corrects 26, and stratification routes 22/30 to *Hard* for postponement. On T2Ranking, the teacher receives only  $\langle q, d \rangle$  ( $\text{RACA}_{q,d}$ ) and its 64.5% score is retained. Complete reported comparisons are listed in Appendix C.

#### 4.4 Comparison of Distillation Strategies

Table 2 compares strategies for learning from RACA labels and rationales. Curriculum SFT and standard GRPO improve Qwen3-1.7B to 76.3% and 78.8% Acc@All, respectively. PODS further raises accuracy to 82.0% on Qwen3-1.7B and 83.6% on Qwen3-8B. Removing either stratification signal reduces performance, showing that both objective label error and model acceptability are useful for organizing progressive training.

#### 4.5 Public Controls and Label Quality

Table 3 reproduces the full PODS component decomposition on public T2Ranking. RACA-PODS improves RACA-SFT from 55.1% to 60.5% (+5.4,  $p < 10^{-13}$ ); removing label-error stratification, acceptability, or consistency reduces accuracy by 0.8, 2.4, or 2.0 points. Holding grade quality fixed further separates the effects: ground-truth-label-only SFT scores 54.8%; adding RACA rationales with the same grades gives 63.8% (+9.0); PODS then reaches 67.8% (+4.0,  $p < 10^{-8}$ ). The pseudo-grade variant remains 7.3 points below the ground-truth-grade variant.

Table 4 corrupts training grades while keeping evaluation fixed. PODS retains a +4.0 gain at zero noise, where noise robustness alone cannot explain the improvement, and the margin grows to +6.4 at 40% noise. Thus, the results support both rationale-informed first-token prediction and robustness to noisy supervision, without establishing mechanistic reproduction of teacher reasoning.

Table 5: System-level online A/B test. Only CTR and QCR are statistically significant.

| Metric       | Lift   | 95% interval      | Result               |
|--------------|--------|-------------------|----------------------|
| CTR          | +0.60% | [0.305%, 1.029%]  | significant          |
| QCR          | +0.38% | [0.066%, 0.702%]  | significant          |
| Long-Stay QV | +0.75% | [−0.042%, 1.470%] | not significant      |
| USR          | +0.35% | N/A               | SBS; no traffic test |

#### 4.6 Further Analysis of PODS

Figure 3 analyzes PODS across model scales, decoding strategies, and iterations. The 1.7B student obtains the largest gain and surpasses a 14B Vanilla-SFT model. PODS reduces the greedy-beam gap from 4.2% to 0.6%; we treat this as behavioral evidence of reduced explicit-CoT dependence, not a mechanistic probe. Performance peaks after two iterations, while non-easy samples decrease from 29.7% to 20.2%.

#### 4.7 Online Experiments

We deploy the Qwen3-1.7B model distilled by RACA-PODS at the re-ranking stage of WeChat Search. The FP16 model runs on vLLM with NVIDIA L40 GPUs, uses a 4,096-token context with tail truncation, and stops after the first label token. Against a heavily optimized 0.3B BERT incumbent, latency rises from 13 to 57 ms (+44 ms) and throughput falls from 142 to 100 QPS. This is not a latency-matched replacement; it is a production-feasible operating point within a serving stack whose end-to-end budget is a few hundred milliseconds.

The 10-day A/B test (Nov. 28–Dec. 8, 2025) covers 1.5M users and 12M query views. As Table 5 shows, CTR and QCR improve significantly at  $p < 0.05$ . Long-Stay QV has a positive point estimate but its interval includes zero, while USR is a side-by-side human evaluation with no traffic-based test. We therefore treat this as a system-level comparison against the incumbent, not a controlled PODS ablation.

### 5 Conclusion

In this paper, we present RACA-PODS, a framework that uses reflective agent supervision and progressive optimization to improve first-token relevance prediction in compact models. Public controls show benefits from both rationale supervision and PODS under clean and corrupted labels, while the WeChat Search deployment demonstrates production feasibility within the re-ranking budget.

### 6 Limitations

Despite the encouraging results, the expert-labeled offline benchmark remains relatively small due to the cost of reliable human annotation. Our evidence for reasoning-informed first-token prediction is behavioral rather than mechanistic. Further study is needed for multi-hop retrieval, longer documents, and domains with different relevance guidelines.

#### Ethical Considerations

IQDR and the online evaluation use proprietary WeChat Search data. We report only aggregate statistics and anonymized examples; no raw queries, user identifiers, or user-level outcomes are released. As with any relevance model, residual errors may affect result quality; the model is therefore used only for re-ranking within the existing production system.

#### Acknowledgments

This research was supported by the General Program of Guangdong Provincial Natural Science Foundation of China (Grant No. 2026A1515011710) and Tencent WeChat Rhino-Bird Focused Research Program (Grant No. x2dxD9252930).

#### References

- Jader Abreu, Luis Fred, David Macêdo, and Cleber Zanchettin. 2019. Hierarchical attentional hybrid neural networks for document classification. In *International Conference on Artificial Neural Networks*, pages 396–402. Springer.
- Akhiad Bercovich, Tomer Ronen, Talor Abramovich, Nir Ailon, Nave Assaf, Mohammed Dabbah, Ido Galil, Amnon Geifman, Yonatan Geifman, Izhak Golan, Netanel Haber, Ehud Dov Karpas, Roi Koren, Itay Levy, Pavlo Molchanov, Shahar Mor, Zach Moshe, Najeeb Nabwani, Omri Puny, and 7 others. 2025. Puzzle: Distillation-based NAS for inference-optimized LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 3806–3830. PMLR.
- Nicolas Bougie and Narimawa Watanabe. 2025. Simuser: Simulating user behavior with large language models for recommender system evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 43–60.
- Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network.

- Advances in neural information processing systems*, 6.
- Rui Dong, Wentao Ouyang, and Xiangzheng Liu. 2025. Elec: Efficient large language model-empowered click-through rate prediction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2664–2668.
- Chengcong Gong, Jianfei Yu, and Rui Xia. 2020. Unified feature and instance based domain adaptation for aspect-based sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7035–7045.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-R1 incentivizes reasoning in LLMs through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international conference on information and knowledge management*, pages 55–64.
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.
- Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. *Advances in neural information processing systems*, 27.
- 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.
- Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of proprietary large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3134–3154.
- Xiaoyu Kong, Junguang Jiang, Bin Liu, Ziru Xu, Han Zhu, Jian Xu, Bo Zheng, Jiancan Wu, and Xiang Wang. 2025. Think before recommendation: Autonomous reasoning-enhanced recommender. In *Advances in Neural Information Processing Systems*, volume 38, pages 156622–156645.
- Haowei Liu, Xuyang Wu, Guohao Sun, Hsin-Tai Wu, Zhiqiang Tao, and Yi Fang. 2025a. Ract: Ranking-aware chain-of-thought optimization for llms. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 178–188.
- Jiahao Liu, Xueshuo Yan, Dongsheng Li, Guangping Zhang, Hansu Gu, Peng Zhang, Tun Lu, Li Shang, and Ning Gu. 2025b. Improving llm-powered recommendations with personalized information. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2560–2565.
- Xuan Lu, Haohang Huang, Rui Meng, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2026. [Rethinking reasoning in document ranking: Why chain-of-thought falls short](#). In *The Fourteenth International Conference on Learning Representations*.
- Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425.
- Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.
- OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR’94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 232–241. Springer.
- Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.
- Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Mingyan Wu, Zhenghao Liu, Yukun Yan, Xinze Li, Shi Yu, Zheni Zeng, Yu Gu, and Ge Yu. 2025. Rankcot: Refining knowledge for retrieval-augmented generation through ranking chain-of-thoughts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12857–12874.

Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zhijing Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, and Jin Ma. 2023. T2ranking: A large-scale chinese benchmark for passage ranking. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2681–2690.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64.

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. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pages 1807–1817.

## A Pseudocode of PODS

Algorithm 1 formalizes the progressive training loop of PODS described in Section 3.2. Given the RACA-annotated corpus, PODS first splits the data into  $\eta$  shards and performs cold-start SFT on the initial shard. At each iteration, the current student model is optimized on the stratified Easy and Medium subsets, followed by RI-GRPO to improve first-token decisions under rationale supervision. The updated student is then used to re-stratify the accumulated training pool, so that the curriculum

evolves with the model’s current capability. The Easy-set dropout ratios  $k_1$  and  $k_2$  down-sample already mastered examples before subsequent optimization stages, preventing easy samples from dominating training while preserving exploration of more challenging query-document pairs.

---

### Algorithm 1 Progressive Optimization Distillation

---

**Input:** Number of Iterations  $\eta$ , Data Difficulty Stratification  $\mathcal{G}$ , SFT Training Unit  $\varphi$ , RI-GRPO Training Unit  $\mathcal{J}$ , Reward  $\mathcal{R}$ , Dataset  $\mathcal{D}$ , foundation model  $\mathcal{M}$ , Dropout  $\mathcal{Q}$ , Dropout ratio  $k_1, k_2$ ,

**Data-Split:**

$\{\mathcal{D}_t | t \in [1, \eta]\} = \text{Split}(\mathcal{D}, \eta)$

**Cold-Start:**

$\mathcal{M}_0 = \varphi(\mathcal{M}, \mathcal{D}_1)$

$\mathcal{D}_1^{\text{Easy}}, \mathcal{D}_1^{\text{Medium}}, \mathcal{D}_1^{\text{Hard}} = \mathcal{G}(\mathcal{M}_0, \mathcal{D}_1)$

**for** Iterations  $I \leftarrow 1$  **to**  $\eta$  **do**

$\mathcal{M}'_I = \varphi(\mathcal{M}_{I-1}, \mathcal{D}_I^{\text{Easy}}, \mathcal{D}_I^{\text{Medium}})$

$\mathcal{D}_I^{\text{Easy}} = \mathcal{Q}(\mathcal{D}_I^{\text{Easy}}, k_1)$

$\mathcal{M}_I = \mathcal{J}(\mathcal{M}'_I, \mathcal{D}_I^{\text{Easy}}, \mathcal{D}_I^{\text{Medium}}, \mathcal{R})$

**if**  $I == \eta$  **then**

**break**

$\mathcal{D}_{I+1}^{\text{Easy}}, \mathcal{D}_{I+1}^{\text{Medium}}, \mathcal{D}_{I+1}^{\text{Hard}} = \mathcal{G}(\mathcal{M}_I, \sum_{t=1}^{I+1} \mathcal{D}_t)$

$\mathcal{D}_{I+1}^{\text{Easy}} = \mathcal{Q}(\mathcal{D}_{I+1}^{\text{Easy}}, k_2)$

**Output:** Optimized parameters  $\mathcal{M}_\eta$

---

## B Additional Experimental Details

### B.1 Implementation Details

For RACA, we use DeepSeek-R1-0528 (Guo et al., 2025) as the foundation reasoner. The temperature and repetition penalty are fixed to 0 and 1.0, respectively, to ensure deterministic and stable reasoning-path generation.

For PODS, we use Qwen3 models with 1.7B, 4B, 8B, and 14B parameters as student backbones (Yang et al., 2025). During inference, the temperature is set to 0, and the model follows a label-first format so that relevance prediction can be obtained from the first generated token.

The maximum sequence length is set to 4,096 tokens to support long-form web documents, and the dropout rate is set to 0.1. For the SFT stage, we use a learning rate of  $0.5 \times 10^{-5}$ , a batch size of 256, and a warm-up ratio of 0.1. For RI-GRPO, we use a smaller learning rate of  $1 \times 10^{-6}$ , a batch size of 256, and a warm-up ratio of 0.03 for stable policy optimization. The reward-scaling coefficients  $\alpha$ ,  $\beta$ , and  $\gamma$ , together with the number of PODS iterations  $\eta$ , are selected by grid search on the validation set. The best setting is  $\alpha = 0.7$ ,  $\beta = 0.8$ ,  $\gamma = 5\%$ , and  $\eta = 2$ .

**Standard GRPO baseline.** For the SFT + standard GRPO baseline, we replace RI-GRPO with standard GRPO and use the reward  $R_1 + R_2$ , where  $R_1$  rewards the first-token relevance label and  $R_2$  rewards CoT similarity to the teacher rationale. This baseline uses the same curriculum learning loader as PODS, but does not use the resampling mechanism.

**Production discriminative baseline.** The BERT-based production baseline is a 0.3B-parameter BERT-based model. This baseline has been heavily optimized for the online search system and is difficult to further improve through internal iteration. Detailed serving configurations are omitted for business confidentiality.

**Label-first inference.** The label-first format is designed for deployment efficiency rather than for improving offline accuracy by itself. It allows the online model to stop generation immediately after the first relevance-grade token. The following analysis compares label-first and CoT-first SFT to clarify this trade-off.

## B.2 Evaluation Metrics

We evaluate point-wise relevance grading with three complementary metrics. Accuracy is reported both overall and per class to measure label-wise prediction quality. Macro-F1 treats all relevance classes equally and is therefore suitable for imbalanced search data. Quadratic Weighted Kappa (QWK) is used because relevance labels are ordinal and larger label disagreements should receive stronger penalties.

## B.3 Baselines

We compare RACA-PODS with four groups of baselines. First, we include a BERT-based discriminative model as the representative production-style classification baseline. Second, we evaluate LLM-based baselines, including zero-shot general-purpose LLMs such as DeepSeek-V3 and Qwen3-235B without explicit thinking, an LLM-based reranker, Qwen3-Reranker-8B, and task-tuned Qwen3 and Qwen3-Reranker models trained on query-document pairs. For compactness, Qwen3-Reranker-8B is abbreviated as Q3-Reranker-8B in Table 1; similarly, RACA-PODS Qwen3-1.7B is abbreviated as RACA-PODS Q3-1.7B. Third, we compare against explicit reasoning models, including DeepSeek-R1, Qwen3-235B-Thinking, GPT-4.1, RACA, and RACA without reflection. Finally, we evaluate reason-distillation methods, including

Vanilla SFT, Curriculum SFT, SFT with standard GRPO, PODS, and PODS ablations.

## C Statistical Uncertainty and Release

We did not retain per-instance IQDR-HA predictions, so a paired bootstrap cannot be reconstructed. We instead report 95% Wilson intervals and two-proportion tests computed from the reported accuracies and sample sizes. The latter discard within-item correlation and are conservative when systems tend to succeed on the same examples. Table 6 reports intervals for every Acc@All entry in Table 1; Table 7 reports the comparisons discussed in the main text, including the non-significant IQDR-HA teacher comparison.

### C.1 Public Release and Proprietary Scope

We will release the T2Ranking sample indices, data-processing configuration, RI-GRPO reward implementation, and anonymized template examples. Raw IQDR data and the complete WeChat Search production guideline are proprietary; their construction and annotation procedures are documented with anonymized examples.

## D Additional Analysis of PODS

### D.1 Efficiency across Model Scales

As shown in Fig. 3a, PODS consistently improves Qwen3 students from 1.7B to 14B parameters over the Vanilla SFT baseline. The relative gain is larger for smaller models: the 14B model obtains a 10% absolute accuracy improvement, while the 1.7B model improves by 16%. Notably, PODS enables Qwen3-1.7B to outperform Qwen3-14B trained with Vanilla SFT, suggesting that progressive rationale-guided training is especially effective for resource-constrained deployment.

### D.2 CoT Dependency

Industrial search requires low-latency inference, making long-form CoT generation impractical. We therefore compare greedy decoding with beam search of size 5 under the label-first format. Since greedy decoding fixes the relevance label before generating any explanation, while beam search can be influenced by later reasoning tokens, the performance gap is a behavioral measure of explicit-CoT dependence. As shown in Fig. 3b, PODS reduces this gap from 4.2% under SFT to 0.6%. This supports reasoning-informed first-token prediction but is not a mechanistic probe of latent reasoning.

Table 6: 95% Wilson intervals for Acc@All. IQDR-HA uses  $n = 300$  and T2Ranking uses  $n = 9,968$ .

| Method                     | IQDR-HA      | T2Ranking    |
|----------------------------|--------------|--------------|
| BERT                       | [63.2, 73.7] | [48.3, 50.3] |
| DeepSeek-V3                | [76.5, 85.3] | [56.1, 58.1] |
| Qwen3-235B w/o thinking    | [75.9, 84.8] | [55.6, 57.6] |
| Q3-Reranker-8B             | [44.1, 55.3] | [47.7, 49.7] |
| Vanilla SFT Q3-Reranker-8B | [74.4, 83.5] | [54.2, 56.2] |
| Vanilla SFT Qwen3-8B       | [73.0, 82.3] | [54.1, 56.1] |
| Vanilla SFT Qwen3-1.7B     | [71.6, 81.1] | [53.8, 55.8] |
| DeepSeek-R1                | [83.1, 90.6] | [61.5, 63.4] |
| Qwen3-235B-Thinking        | [81.3, 89.2] | [59.9, 61.9] |
| GPT-4.1                    | [82.4, 90.1] | [61.1, 63.0] |
| RACA w/o Reflection        | [82.7, 90.3] | [60.8, 62.7] |
| RACA                       | [88.0, 94.3] | [63.6, 65.4] |
| RACA-SFT Qwen3-1.7B        | [73.3, 82.6] | [54.1, 56.1] |
| L*-RACA-SFT                | –            | [62.9, 64.7] |
| RACA-PODS Qwen3-1.7B       | [81.3, 89.2] | [59.5, 61.5] |
| L*-RACA-PODS               | –            | [66.9, 68.7] |

Table 7: Two-proportion tests for reported Acc@All comparisons.  $\Delta$  is in percentage points.

| Dataset   | Comparison                          | $n$   | $\Delta$ | $z$  | $p$          |
|-----------|-------------------------------------|-------|----------|------|--------------|
| IQDR-HA   | RACA-PODS vs. RACA-SFT              | 300   | +7.4     | 2.36 | 0.018        |
| IQDR-HA   | RACA-PODS vs. Vanilla SFT           | 300   | +9.0     | 2.82 | 0.005        |
| IQDR-HA   | RACA-PODS vs. BERT                  | 300   | +17.0    | 4.96 | $< 10^{-6}$  |
| IQDR-HA   | RACA vs. DeepSeek-R1                | 300   | +4.4     | 1.76 | 0.079        |
| T2Ranking | RACA <sub>q,d</sub> vs. DeepSeek-R1 | 9,968 | +2.0     | 2.93 | 0.003        |
| T2Ranking | RACA-PODS vs. RACA-SFT              | 9,968 | +5.4     | 7.72 | $< 10^{-13}$ |
| T2Ranking | L*-PODS vs. L*-SFT                  | 9,968 | +4.0     | 5.95 | $< 10^{-8}$  |

### D.3 Number of PODS Iterations

Figure 3c shows the effect of the number of PODS iterations. Accuracy improves in the early stage and peaks at two iterations, but declines afterward. This suggests that progressive optimization helps the student absorb increasingly difficult reasoning cases, while excessive iterations may overfit the current data distribution. We therefore set  $\eta = 2$  in the main experiments.

### D.4 Evolution of Sample Difficulty

To verify whether PODS progressively resolves hard cases, we track the proportion of “non-easy” samples on IQDR-RACA. As shown in Fig. 3d, this proportion decreases from 29.7% to 20.2%, with the largest reduction after the first iteration. This trend supports the design of PODS: by introducing new mini-batches at each iteration and filtering samples by difficulty and acceptability, the student model gradually absorbs reasoning patterns that are difficult to learn through one-shot SFT.

## E Hyperparameters $\beta$ and $\gamma$

The selection of hyperparameters affects final performance. We therefore examine  $\beta$  and  $\gamma$  with

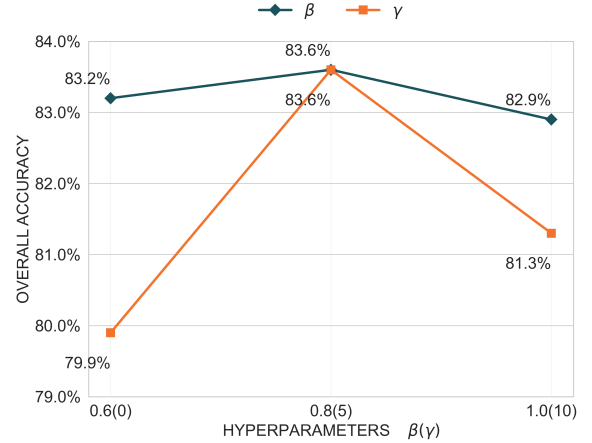


Figure 4: Comparison of  $\beta$  and  $\gamma$

Qwen3-8B (Yang et al., 2025).

$\beta$  is the coefficient of  $R_2$  in RI-GRPO and controls the contribution of CoT similarity; performance peaks at  $\beta = 0.8$ . The stratification parameter  $\gamma$  controls the proportion postponed as *Hard*; performance peaks at  $\gamma = 5\%$ . Larger values postpone too many samples and reduce performance, as shown in Fig. 4.

Table 8: Performance Comparison in Yelp-Review-Full.

| Method                                    | Model Type     | Overall Accuracy       |
|-------------------------------------------|----------------|------------------------|
| HAHNN (Abreu et al., 2019)                | CNN            | 73.28%                 |
| XLNet (Yang et al., 2019)                 | BERT           | 72.95%                 |
| BERT-ITPT-FiT (Sun et al., 2019)          | BERT           | 70.58%                 |
| BERT+UDA (Gong et al., 2020)              | BERT           | 67.92%                 |
| Qwen3-235B (Few-shot) (Yang et al., 2025) | Autoregressive | 64.34%                 |
| Qwen3-1.7B (PODS)                         | Autoregressive | 73.58% (+3.62%)        |
| Qwen3-1.7B (Vanilla SFT)                  | Autoregressive | 69.96%                 |
| Qwen3-8B (PODS)                           | Autoregressive | <b>74.85% (+2.71%)</b> |
| Qwen3-8B (Vanilla SFT)                    | Autoregressive | 72.14%                 |

## F Cross-task Generalization Validation in Yelp

### F.1 Dataset.

To assess the versatility and robustness of PODS beyond IR, we introduce Yelp-Review-Full, a five-class sentiment classification benchmark. While the domain differs, it shares an ordinal logical structure and requires nuanced semantic understanding similar to search relevance. This experiment aims to demonstrate that PODS serves as a general-purpose distillation paradigm for empowering Tiny-LLMs in broad natural language understanding tasks.

### F.2 Experiment Result.

To test transfer beyond search relevance, we apply PODS to Yelp-Review-Full with Qwen3-1.7B and Qwen3-8B (Yang et al., 2025). Table 8 compares against representative CNN- and BERT-based approaches. PODS improves Qwen3-1.7B from 69.96% to 73.58% and Qwen3-8B from 72.14% to 74.85%, with the larger gain again occurring for the smaller student.

## G Case Study: Reasoning-Informed Prediction in Complex Scenarios

To illustrate how RACA rationales can inform a Tiny-LLM’s first-token prediction, we analyze a representative search query: *"Side effects of Hydrogesterone tablets"*.

**Challenge of Latent Intent.** The query contains a critical spelling error, substituting “Hydrogesterone” for “Dydrogesterone.” Traditional relevance models often fail such cases due to lexical mismatch or lack of domain-specific reflection.

**RACA Multi-Step Reasoning.** Our RACA framework resolves this through a four-module chain:

- **Query Analysis & Summary:** Identifies the core intent as seeking medical side effects

and notes the document’s focus on comparing natural vs. synthetic progesterone.

- **Reflection (The "Re-invoke" Step):** Critically identifies the typo, recognizing that the user is an ordinary consumer likely inquiring about “Dydrogesterone.”
- **Final Judgment:** Correlates the corrected intent with the document content. Since the document focuses on drug comparisons but omits the specific side effects requested, RACA assigns a **Relevance Grade of 2** (Weakly Relevant).

**PODS First-Token Effect.** While a *Vanilla SFT* model exhibits a 3.7%–4.2% greedy-beam gap, the **PODS**-trained student reduces it to **0.6%**. In this case, the first-token prediction benefits from supervision that includes typo correction and intent reflection, while deployment avoids generating the rationale. The result demonstrates reduced explicit-CoT dependence, not reproduction of the teacher’s latent process.

## H Template of Prompt

In this section, we present the system prompt templates corresponding to four modules of RACA in detail. These modules can be dynamically adjusted according to specific business requirements and exhibit good extensibility.

### H.1 Query Requirement Understanding

You are an expert in user requirements analysis, and you are dealing with a real-world user search scenario.

#### TASK DEFINITION

Input the user’s query-related information [Query, Timeliness, Address...], and provide a core user demand analysis within N words.

– If the demand is broad, output "Broad Demand"; otherwise, output "Precise Demand".

– If the Timeliness requirement is obvious, output "Strong Timeliness"; otherwise, output "Weak Timeliness".

– If the Address intent is obvious: output "Clear Address" if a unique address can be confirmed; output "Ambiguous Address" if the address is not unique.

– If ...

You must provide the most suitable information carrier for the user’s demand, such as Encyclopedia, Article, Video, Product, etc.

#### REFLECTION NOTES

If the input contains error logs, you must analyze and reason based on the error logs.

#### EXAMPLES

You may refer to these examples:

[Query: XXX]

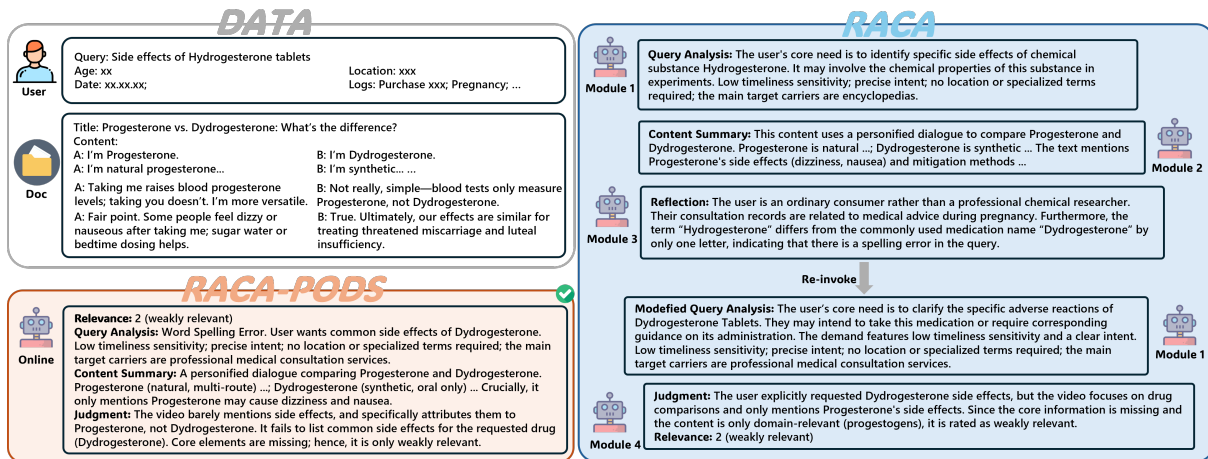


Figure 5: Case Study

[Core Demand: XXX]  
...  
**OUTPUT FORMAT**  
Finally, output in the dictionary format: {"reason": <reason>; "score": 0}. Replace <reason> with your analysis (a string within L words, no quotation marks required).

## H.2 Document Content Summarization

You are an expert in content summarization and induction, and you are dealing with a real-world user search scenario.

### TASK DEFINITION

Input the complete information of the document [Part1, Part2, Part3...], and make a comprehensive summary of the document content.

– Due to word count restrictions, overly long content will be truncated, with only the first and last xxx characters reserved. The text may also have recognition errors, disordered text order or illogical language. You must conduct comprehensive reasoning on such fragmented information and not be affected by individual wrong characters.

– It should be noted that if there is a logical sequence between different parts of the document, you should also summarize the content in accordance with the original logical order.

– You need to retain as much useful information from the original text as possible and ensure that the summary is universally applicable to any question-answering scenarios.

### REFLECTION NOTES

If the input contains error logs, you must analyze and reason based on the error logs.

### OUTPUT FORMAT

Finally, output in the dictionary format: {"reason": <reason>; "score": 0}. Replace <reason> with your summary (a string within N words and does not require quotation marks).

## H.3 Reflection

You are an expert in content error correction, and you are dealing with a real-world user search scenario.

### TASK DEFINITION

Now you are provided with a user's original query information, output information from historical modules, document information, and relevant documents in the database. You need to integrate all the above information to conduct joint error correction for the three situations.

– 1. In real-world scenarios, only two types of errors are taken into account for the user-input query and documents, namely Pinyin input method errors and glyph-similar errors. No modifications shall be made for errors other than these two types, and you must be prudent.

– 2. Some nouns in the query and documents are special nouns, whose accurate meanings can be understood by combining the current document and relevant documents in the database.

– 3. If the query presents an obvious account search intent, the query demand is to "Search for documents published by the account XXX".

– It should be noted that this does not mean the content referred to in the database is relevant. It only serves as a reference for you to grasp the precise meaning of certain words in the user's query under the current Internet context, and to infer and supplement potential timeliness or address requirements.

### EXAMPLES

You may refer to these examples:

[Input: XXX]

[Reflection: XXX]

...

### WARNING

Database documents can be used as a reference only if at least half of them refer to the same meaning for a noun; otherwise, the database information is untrustworthy and the default meaning shall be adopted.

### OUTPUT FORMAT

If errors exist, you need to state the error category, the reason for the error, and the module to be re-invoked (query demand analysis or document content summarization). The query shall not be modified to cater to the documents, nor shall the documents be modified to cater to the query. Finally, output in the dictionary format:

`{"reason": <reason>;"score":0}`. Replace the content in `<reason>` with the error category, the error reason, and the module to be re-invoked. If there are no errors, output `{"reason": "Completely correct";"score":0}`.

## H.4 Judgement

You are an expert in analysis and summarization, and you are dealing with a real-world user search scenario.

### TASK DEFINITION

The historical outputs of the input modules are now provided. You need to summarize such information to derive the final result, and the requirements for you are as follows:

You shall combine the analyses given by the preceding experts, and assign a score (ranging from 1 to k) in accordance with the criteria below:

– Score k (Perfect match): XXX.

...

– Score 1 (Irrelevant): XXX.

You must default to the assumption that the user is a novice with poor reasoning ability.

This is a highly rigorous system. You have to make conservative estimations, and prioritize giving a lower score for ambiguous cases.

### OUTPUT FORMAT

Finally, output in the dictionary format: `{"reason": <reason>;"score":0}`. Replace the content in `<reason>`, where `<score>` is your score (an integer from 1 to k, without quotation marks) and `<reason>` is your brief justification (a string within xxx characters, without quotation marks).

## I Stepwise vs. Label-First Training

Speculative decoding accelerates explicit generation but does not remove the need for reasoning tokens. We therefore compare with a stepwise CoT-first baseline. As shown in Table 9, PODS achieves higher accuracy and F1 on Qwen3-1.7B.

Table 9: Comparison with stepwise CoT-first training on Qwen3-1.7B.

| Method             | Acc@C1 | Acc@C2 | Acc@C3 | Acc@All | F1    | QWK    |
|--------------------|--------|--------|--------|---------|-------|--------|
| Stepwise CoT-first | 83.7%  | 73.1%  | 74.5%  | 77.6%   | 0.769 | 0.7863 |
| PODS               | 90.3%  | 73.6%  | 79.6%  | 82.0%   | 0.811 | 0.838  |

## J Additional Ablation Studies

### J.1 RACA Module Ablation

Table 10 evaluates the contribution of each major RACA component. Removing either the query-side module, the document-side module, or reflection reduces performance, indicating that the gains do not come from a single generic prompting step.

### J.2 Reflection in Distillation Data

Table 11 further evaluates the impact of reflection after distillation. Training PODS on data generated without reflection reduces Qwen3-1.7B per-

formance, suggesting that teacher-side reflection improves the quality of the supervision signal.

Table 11: PODS ablation using RACA data without reflection on Qwen3-1.7B.

| Method              | Acc@C1 | Acc@C2 | Acc@C3 | Acc@All | F1    | QWK   |
|---------------------|--------|--------|--------|---------|-------|-------|
| PODS w/o Reflection | 87.2%  | 69.5%  | 74.9%  | 78.1%   | 0.771 | 0.799 |
| PODS                | 90.3%  | 73.6%  | 79.6%  | 82.0%   | 0.811 | 0.838 |

### J.3 Consistency Reward

Table 12 shows that removing the consistency reward  $R_3$  harms PODS performance. This supports the role of  $R_3$  in aligning the first-token label with the generated rationale during training.

Table 12: Ablation of the consistency reward  $R_3$  on Qwen3-1.7B.

| Method         | Acc@C1 | Acc@C2 | Acc@C3 | Acc@All | F1    | QWK   |
|----------------|--------|--------|--------|---------|-------|-------|
| PODS w/o $R_3$ | 88.2%  | 70.4%  | 76.4%  | 79.2%   | 0.782 | 0.811 |
| PODS           | 90.3%  | 73.6%  | 79.6%  | 82.0%   | 0.811 | 0.838 |

### J.4 Label-First versus CoT-First SFT

Table 13 shows that label-first SFT alone does not improve performance; it slightly underperforms CoT-first SFT. This confirms that the label-first format is mainly an efficiency-oriented serving design, and the performance gains of PODS come from progressive distillation and RI-GRPO rather than from the answer position alone.

Table 13: Vanilla SFT comparison between CoT-first and label-first formats on Qwen3-1.7B.

| Method                              | Acc@C1 | Acc@C2 | Acc@C3 | Acc@All | F1    | QWK   |
|-------------------------------------|--------|--------|--------|---------|-------|-------|
| Vanilla SFT (CoT-first)             | 74.9%  | 63.5%  | 76.8%  | 72.5%   | 0.717 | 0.742 |
| Vanilla SFT (CoT-post, label-first) | 73.2%  | 61.8%  | 76.1%  | 71.1%   | 0.704 | 0.730 |

### J.5 Teacher-Model Sensitivity

Table 14 compares different teacher sources for PODS. RACA provides the strongest supervision among the tested teachers, but PODS also benefits from reasoning traces generated by other strong reasoners.

## K Analysis of Reasoning-Informed First-Token Prediction

### K.1 Consistency Reward Intuition

Let  $x$  denote the input query-document instance,  $r$  denote a generated rationale, and  $s$  denote the relevance label. To avoid overloading the search-context variable used in the main paper, we use  $r$  rather than  $c$  for the rationale. Under a high-reward

Table 10: RACA module ablation experiments.

| Method                          | IQDR-HA |        |        |         |       |       | T2Ranking |        |         |       |       |
|---------------------------------|---------|--------|--------|---------|-------|-------|-----------|--------|---------|-------|-------|
|                                 | Acc@C1  | Acc@C2 | Acc@C3 | Acc@All | F1    | QWK   | Acc@C1    | Acc@C2 | Acc@All | F1    | QWK   |
| RACA w/o Query Understanding    | 88.3%   | 89.0%  | 91.3%  | 89.3%   | 0.893 | 0.891 | 57.2%     | 70.3%  | 63.2%   | 0.632 | 0.278 |
| RACA w/o Document Summarization | 86.7%   | 88.0%  | 91.3%  | 88.3%   | 0.883 | 0.886 | 56.4%     | 69.1%  | 62.2%   | 0.622 | 0.251 |
| RACA w/o Reflection             | 85.0%   | 87.0%  | 90.0%  | 87.0%   | 0.871 | 0.867 | 55.9%     | 68.7%  | 61.8%   | 0.617 | 0.242 |
| RACA                            | 90.8%   | 92.0%  | 92.5%  | 91.7%   | 0.916 | 0.912 | 58.8%     | 71.2%  | 64.5%   | 0.645 | 0.295 |

Table 14: Teacher-model sensitivity for PODS Qwen3-1.7B.

| Method                              | IQDR-HA |        |        |         |       |       | T2Ranking |        |         |       |       |
|-------------------------------------|---------|--------|--------|---------|-------|-------|-----------|--------|---------|-------|-------|
|                                     | Acc@C1  | Acc@C2 | Acc@C3 | Acc@All | F1    | QWK   | Acc@C1    | Acc@C2 | Acc@All | F1    | QWK   |
| DeepSeek-R1-PODS Qwen3-1.7B         | 82.5%   | 81.0%  | 87.5%  | 83.3%   | 0.835 | 0.809 | 53.3%     | 66.2%  | 59.2%   | 0.592 | 0.192 |
| Qwen3-235B-Thinking-PODS Qwen3-1.7B | 80.0%   | 79.0%  | 83.8%  | 80.7%   | 0.806 | 0.758 | 52.7%     | 65.6%  | 58.6%   | 0.586 | 0.180 |
| RACA-PODS Qwen3-1.7B                | 84.2%   | 85.0%  | 88.8%  | 85.7%   | 0.856 | 0.850 | 54.6%     | 67.5%  | 60.5%   | 0.605 | 0.217 |

optimum, the consistency reward encourages the rationale and label to satisfy a deterministic mapping  $s = \sigma(r, x)$ . Define  $\mathcal{R}(s, x) = \{r : \sigma(r, x) = s\}$ . Then

$$p^*(s | x) = \sum_{r \in \mathcal{R}(s, x)} p^*(r | x). \quad (8)$$

Taking expectation over rationales gives

$$\begin{aligned} \mathbb{E}_{r|x} [p^*(s | r, x)] &= \sum_r p^*(r | x) \mathbf{1}[s = \sigma(r, x)] \\ &= \sum_{r \in \mathcal{R}(s, x)} p^*(r | x) \\ &= p^*(s | x). \end{aligned} \quad (9)$$

This provides an intuition for why optimizing label-rationale consistency can reduce dependence on explicit CoT at inference time: after training, the first-token label prediction is encouraged to preserve the utility of the rationale supervision.

## K.2 First-Token Compression Limit

To explore the limit of first-token prediction under rationale supervision, we partitioned IQDR-RACA examples by reasoning-chain token lengths using powers-of-two intervals. Accuracy for examples with reasoning chains longer than 2,048 tokens drops sharply below the overall Acc@All. This suggests that one-token relevance classification has a practical compression limit for very long reasoning traces in this setting.

## L Hallucination Robustness

We additionally analyze hallucination cases from the teacher model. Among 30 randomly sampled hallucinatory instances used by PODS, 22 were categorized as Hard, 6 as Medium, and 2 as Easy

under our difficulty stratification. This supports the design choice of postponing Hard samples during each iteration. We also compare RACA with and without reflection: among 30 hallucinations produced by RACA without reflection, full RACA corrected 26 cases and left 4 unresolved. This indicates that reflection and difficulty filtering mitigate, but do not fully eliminate, inherited teacher errors. In deployment, the distilled student outputs only relevance labels rather than CoT rationales, further reducing the surface for exposing hallucinated explanations.