Can Compact Language Models Search Like Agents? Distillation-Guided Policy Optimization for Preserving Agentic RAG Capabilities

Anonymous ACL submission

Abstract

Reinforcement Learning has emerged as a dominant post-training approach to elicit agentic RAG behaviors such as search and planning from language models. Despite its success with larger models, applying RL to compact models (*e.g.,* 0.5–1B parameters) presents unique challenges. The compact models exhibit poor initial performance, resulting in sparse rewards and unstable training. To overcome these difficulties, we propose Distillation-Guided Policy Optimization (DGPO), which employs cold-start initialization from teacher demonstrations and continuous teacher guidance during policy optimization. To understand how compact models preserve agentic behavior, we introduce Agentic RAG Capabilities (ARC), a fine-grained metric analyzing reasoning, search coordination, and response synthesis. Comprehensive experiments demonstrate that DGPO enables compact models to achieve sophisticated agentic search behaviors, even outperforming the larger teacher model in some cases. DGPO makes agentic RAG feasible in computing resource-constrained environments. [1]

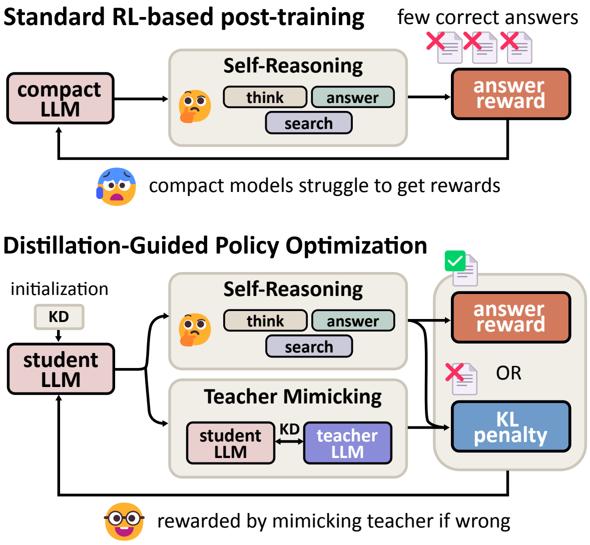

Figure 1: **Distillation-Guided Policy Optimization.** Top: Compact models struggle to earn rewards due to poor capability, which leads to training collapse. Bottom: DGPO establishes a stable reward mechanism by guiding incorrect answers through teacher mimicry.

1 Introduction

Agentic RAG (Singh et al., 2025) has emerged as a new paradigm where LLMs function as autonomous search agents, coordinating retrieval, query reformulation, and evidence integration. While externalizing knowledge storage, these systems require sophisticated reasoning abilities within the LLMs for effective search coordination. Consequently, existing agentic RAG systems predominantly rely on large language models with billions of parameters (Xu and Peng, 2025), leaving the potential of agentic RAG in resource-constrained environments largely unexplored. The emergence of small language models (SLMs) (Belcak et al., 2025), particularly compact models (e.g., 0.5–1B) raises a compelling question: *can we unlock the latent potential of compact language models to acquire the art of agentic RAG?*

Eliciting agentic search capabilities from smaller language models typically requires two approaches: reinforcement learning (RL) via self-exploration and knowledge distillation (KD) from a teacher model. We refer to the compact model under training as the *student*, regardless of the approach. Yet both approaches become largely ineffective for compact models (0.5–1B) due to their poor initial capabilty. RL (Schulman et al., 2017; Shao et al., 2024) suffers from sparse rewards and poor exploration due to weak student-generated outputs (SGOs). Standard KD (Hinton et al., 2015; Shing et al., 2025) using only teacher-generated outputs (TGOs) leads to exposure bias (Bengio et al., 2015) while on-policy distillation methods (Gu et al., 2024; Agarwal et al., 2024) also suffer from the

[1]The source code is available at: https://anonymous.4open.science/r/DGPO

noisy and low-quality nature of SGOs. Neither approach addresses the fundamental bottleneck of poor initial output quality in compact models.

To overcome this fundamental bottleneck, we propose Distillation-Guided Policy Optimization (DGPO), a novel RL framework that addresses the core issue of low-quality SGOs through the strategic integration of teacher guidance and RL (Figure 1). DGPO operates through two key mechanisms. First, cold-start initialization through KD using TGOs dramatically stabilizes early training by providing high-quality initial trajectories. Second, selective teacher guidance during RL that rewards correct self-reasoning while providing teacher mimicry for incorrect attempts. This synergy between selective KL-based teacher guidance and RL-driven self-exploration allows the compact model to discover policies that outperform the teacher in some experimental settings.

To understand how DGPO preserves agentic capability in compact models, we introduce Agentic RAG Capabilities (ARC), a fine-grained evaluation framework that decomposes the agentic search into three core dimensions: *thinking*, *query rewriting*, and *source referencing* (Figure 2). Unlike conventional metrics that focus on final accuracy, ARC evaluates the agentic search process, revealing how different aspects of agentic behavior emerge and decline across different models. Comprehensive evaluations demonstrate that DGPO consistently outperforms baselines in final accuracy. ARC reveals that DGPO improves multi-hop reasoning and coordination while maintaining competitive performance in source referencing and query rewriting. Such capability-level insights are crucial for advancing agentic RAG in compact models.

Our contributions are summarized in four key dimensions. **(i) Problem:** we pioneer the challenging domain of agentic RAG post-training for extremely compact models (0.5–1B), identifying fundamental challenges that existing methods fail to address. **(ii) Methodology:** We propose Distillation-Guided Policy Optimization (DGPO), an RL framework designed to stabilize training in compact models via cold-start initialization and selective teacher guidance. **(iii) Evaluation:** we present ARC, a capability-level evaluation framework that provides a detailed diagnosis of agentic behavior. **(iv) Results:** DGPO outperforms RL and distillation baselines across multiple model families and sizes. Remarkably, our method achieves **teacher-surpassing performance** on several datasets.

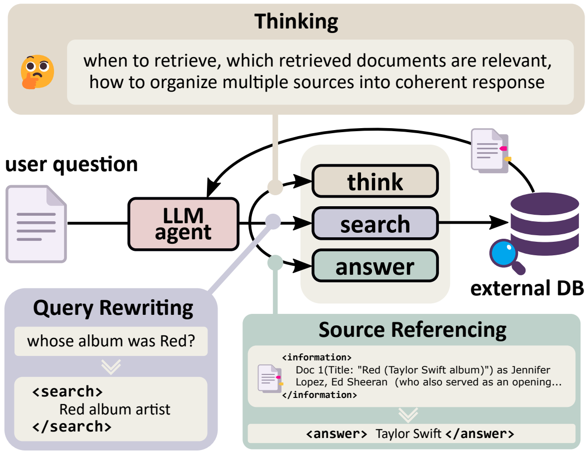

Figure 2: **Agentic RAG capability.** We introduce Agentic RAG Capability (ARC) which characterizes the core capabilities of LLMs required for agentic RAG systems. ARC is evaluated as three primary components: *thinking*, *query rewriting*, and *source referencing*.

2 Related Work

Agentic RAG. WebGPT (Nakano et al., 2022) introduced RLHF-driven browser interaction for retrieval-grounded QA. ReAct (Yao et al., 2023) generalized this idea by interleaving chain-of-thought and tool calls via special <think> or <act> tokens. To tighten the coupling between retrieval and reasoning, IRCoT (Trivedi et al., 2023) explicitly alternates each chain-of-thought (CoT) step with a retrieval. Adaptive-RAG (Wang et al., 2025) further predicts retrieval steps based on question complexity. Most recently, Search-R1 (Jin et al., 2025) leveraged RL to teach an LLM to generate multi-turn search queries, achieving state-of-the-art results. Our work specifically focuses on enabling agentic RAG in compact models and introduces a comprehensive evaluation framework for multi-dimensional capability evaluation.

Post-training for LLMs. RL algorithms such as PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) have proven effective in enhancing reasoning capabilities for LLMs (Comanici et al., 2025; Yang et al., 2025), particularly in domains like mathematical problem solving. At the initial stage of training, base models require sufficient performance to obtain meaningful rewards; otherwise, sparse reward signals lead to training instability. To address this cold-start problem, DeepSeek-R1 (Guo et al., 2025) demonstrates that SFT-based model initialization effectively warms up the model prior to RL, achieving favorable results through CoT demonstrations. Our work is the first to integrate distillation principles into both cold-start initialization and concurrent RL training, enabling stable distillation-guided learning in compact models.

Knowledge Distillation for LLMs. Knowledge distillation (KD) (Hinton et al., 2015) enables smaller student models to learn from larger teacher models by matching softened output distributions. To mitigate the capacity gap between student and teacher models (Mirzadeh et al., 2020; Zhang et al., 2023a), some methods use interpolated or smoothed intermediate student and teacher distributions (Ko et al., 2024; Shing et al., 2025). However, because these methods rely on TGOs during training while inference still uses the SGOs, a train–inference mismatch arises, leading to exposure bias (Bengio et al., 2015). To mitigate this, recent work also proposes on-policy distillation from SGOs (Agarwal et al., 2024; Gu et al., 2024; Yang et al., 2025), where the student learns directly from its own generated outputs during training. Another limitation is that distillation methods require sensitive teacher-guidance schedulers (Ko et al., 2024; Shing et al., 2025), which are hard to tune and affect stability. Meanwhile, conventional single-hop RAG distillation methods (Kang et al., 2023; Zhang et al., 2023b; Liao et al., 2025) typically use sequence-level knowledge distillation (SeqKD) (Kim and Rush, 2016), an SFT-based approach where students learn from teacher-generated CoT demonstrations rather than matching output distributions. While SeqKD improves the reasoning capability of SLMs (Guo et al., 2025), cross-entropy training with hard targets alone is insufficient to transfer the richer distributional knowledge from the teacher (Hinton et al., 2015). Our method addresses these challenges through a two-stage approach: off-policy KD on TGOs for cold-start initialization, followed by off-policy KD-driven RL optimization on SGOs to further extract model capabilities without requiring handcrafted schedulers.

3 Agentic RAG

3.1 Framework

While agentic RAG (Singh et al., 2025) encompasses diverse interpretations, we define it through two key characteristics: (1) autonomous, adaptive search coordination behaviors, and (2) three fundamental actions—*thinking*, *searching*, and *answering*—that constitute agentic intelligence. In agentic RAG, LLMs function as agents making sequential decisions at each timestep t. Given a user question

Question: Whose album was Red?
<think> To answer this question, I need to find out which artist released the album titled "Red". </think>
<search> Red album artist </search>
<information> **Doc 1(Title: "Red (Taylor Swift album)")** as Jennifer Lopez, Ed Sheeran (who also served as an ... **Doc 2(Title: "Red (band)")** deal with Christian label Essential Records in 2004. Their debut albums ... **Doc 3(Title: "Red (Dia Frampton album)")** Red (Dia Frampton album) Red is the debut ... </information>
<think> From the information provided, it is clear that the album "Red" was released by Taylor Swift. It was her fourth studio album and had a country sound. </think>
<answer> Taylor Swift </answer>

Table 1: **An example of Reasoning Processes.** The agent exhibits *thinking* through deliberative reasoning, *query rewriting* by reformulating the original query into the actual search query, and *source referencing* by incorporating retrieved information in to the final answer.

$\boldsymbol{x}$ and an external retrieval system $\mathcal{R}$, the LLM agent operates as a policy $\pi_\theta(\boldsymbol{y}_t|\boldsymbol{x}_t;\mathcal{R})$, where

$$\boldsymbol{y} \in \Big\{ \underbrace{\textsc{Think}(\cdot)}_{\text{reasoning token}},\ \underbrace{\textsc{Search}(\cdot)}_{\text{search query}},\ \underbrace{\textsc{Answer}(\cdot)}_{\text{forming an answer}} \Big\}.$$

As demonstrated in Table 1, we employ structured tokens (Jin et al., 2025) to organize the actions: <think> for reasoning, <search> for database queries, <information> for retrieved documents, and <answer> for final responses.

3.2 Agentic RAG Capability (ARC)

We propose Agentic RAG Capability (ARC) as a comprehensive metric to systematically evaluate agentic behavior across multiple dimensions. As demonstrated in Table 1, we characterize ARC through three core dimensions:

Source Referencing. Accurately incorporating retrieved information into final answers (shown in the <information> and <answer> entries).

Query Rewriting. Reformulating user questions into effective search queries, as literal keyword matching often fails to retrieve relevant documents. The agent must paraphrase key concepts and introduce related terms to maximize retrieval effectiveness (illustrated by transforming "Whose album was Red?" into "Red album artist" in <search>).

Thinking. Making informed decisions about when to retrieve information, which documents contain relevant answers, and how to synthesize

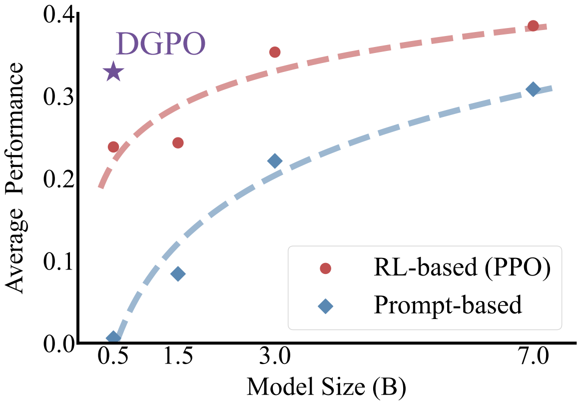

Figure 3: Comparison of prompt-based and RL-based (PPO) post-training agentic RAG across model sizes.

multiple pieces of evidence into coherent responses. This involves assessing context sufficiency and integrating retrieved sources in a logically consistent manner (demonstrated in <think> entries).

3.3 Challenges in Compact Models.

Our preliminary experiments compared the performance of prompt-based and RL-based agentic RAG models across various model sizes, evaluated on the average of seven QA datasets (Figure 3). Here prompt-based refers to Qwen2.5-instruction checkpoints and RL-based refers to post-trained models using PPO (Jin et al., 2025) tailored for agentic RAG. The experimental setup is detailed in Section 5. While RL models boosted performance overall in the context of agentic RAG, smaller models still lagged far behind their larger counterparts. We include this result here to highlight the limitations of applying RL directly to compact models—an observation that motivates our proposed approach, DGPO, introduced in the next section.

4 DGPO: Distillation-Guided Policy Optimization

4.1 Core Framework

Figure 4 depicts our framework which combines distillation and reinforcement learning to train compact agentic RAG models through a two-phase learning strategy, eliminating the need for a handcrafted scheduler. Early-stage student-generated outputs (SGOs) are often noisy and unstable, while teacher-generated outputs (TGOs) provide quality guidance but suffer from exposure bias. To address these challenges, we propose two key mechanisms:

Cold-Start Initialization via KD. In the initial phase, students learn purely from TGOs via knowledge distillation. This provides stable, high-quality trajectories that dramatically improve early training dynamics and establish a strong foundation for subsequent RL optimization.

Selective KL penalty. During the RL phase, we apply KL divergence penalties selectively—only to incorrect predictions—guiding students toward informative teacher behaviors while preserving exploration capabilities. This targeted regularization enables autonomous reasoning development without being overly constrained by the teacher model.

4.2 KD initialization with TGOs

During the cold-start phase, we initialize the student model by distilling from a strong teacher policy using a general KD loss that combines cross-entropy from hard labels and KL divergence as:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{CE}}(\pi_{\text{g}}, \pi_\theta) + \lambda D_{\text{KL}}\big[\pi_{\text{g}}(\cdot|x) \| \pi_\theta(\cdot|x)\big] \ , \qquad (1)$$

where π_θ denotes the student policy and π_{g} is the frozen teacher. We filter TGOs to retain only correct outputs, ensuring the student π_θ learns from high-quality teacher samples.

4.3 Distillation-guided RL with SGOs

Upon reaching a performance threshold, we transition to PPO-based RL using the distilled student as the initial policy. This staged approach stabilizes training dynamics and improves sample efficiency, particularly when the student model has significantly fewer parameters than the teacher. By avoiding premature exploration from a weak policy, our method ensures that RL begins with a reasonable approximation of agentic behaviors.

PPO with Search Engine Proximal Policy Optimization (PPO) (Schulman et al., 2017) is a widely used RL algorithm for LLM fine-tuning, offering stable training for compact models. Our method optimizes LLMs with search engine $\mathcal{R}$ by maximizing the following objective,

$$\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_{\text{old}}(\cdot|x;\mathcal{R})}\Bigg[\frac{1}{\sum_{t=1}^{|y|}\mathbb{1}(y_t)}\sum_{\substack{t=1\\ \mathbb{1}(y_t)=1}}^{|y|}\min\Big(\frac{\pi_\theta(y_t \mid x, y_{<t};\mathcal{R})}{\pi_{\text{old}}(y_t \mid x, y_{<t};\mathcal{R})}A_t, \\ \text{clip}\Big(\frac{\pi_\theta(y_t|x, y_{<t};\mathcal{R})}{\pi_{\text{old}}(y_t|x, y_{<t};\mathcal{R})}, 1-\epsilon,\ 1+\epsilon\Big)A_t\Big)\Bigg], \qquad (2)$$

where π_θ and π_{old} represent the current and previous student policy models, respectively. x denotes

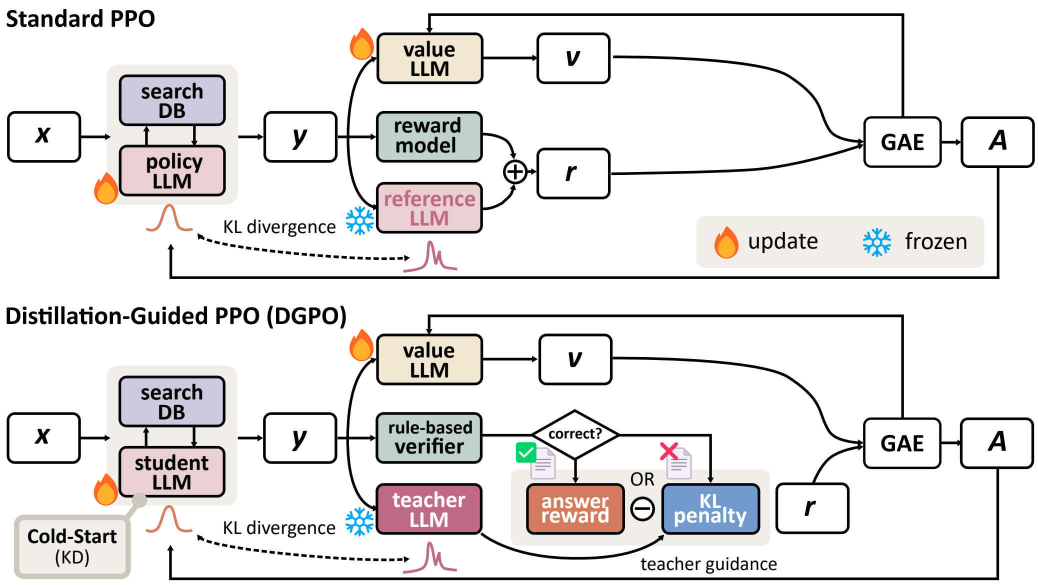

Figure 4: Top: Standard PPO pipeline for post-training LLMs. The reference LLM serves as a regularization anchor to prevent excessive deviation from the initial policy. Bottom: Our proposed distillation-guided PPO pipeline. Unlike conventional approaches where the reference model merely constrains policy drift, our framework employs the teacher model to actively guide the student toward correct behaviors when autonomous attempts fail, transforming the reference's role from passive regularization to active pedagogical guidance.

input samples and y represent the generated outputs interleaved with search engine calling results. The term ϵ is a clipping-related hyperparameter introduced in PPO to stabilize training. The advantage estimate A_t is computed using Generalized Advantage Estimation (GAE) (Schulman et al., 2018), based on future rewards and a learned value function. $\mathbb{1}(y_t)$ is a token loss masking operation. See Appendix B.1 for details on token masking.

Reward and Selective KL penalty We employ binary exact matching (EM) for answer rewards to prevent reward hacking:

$$r_{\text{answer}}(x, y) = \begin{cases} 1 & \text{if } y = y^* \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

where y is the predicted answer and y^* is the ground-truth. However, Eq. (3) provides no learning signal for incorrect predictions, causing training stagnation with poor SGOs. To address this, we introduce selective KL penalty. The student π_θ receives reward for correct self-reasoning, but when incorrect, the teacher π_g guides the student to mimic teacher behavior through KL regularization,

$$r_\phi(x, y) = \begin{cases} 1 & \text{if } y = y^* \\ -\beta D_{\text{KL}}\left[\pi_\theta(y|x;\mathcal{R}) \| \pi_g(y|x;\mathcal{R})\right] & \text{otherwise}. \end{cases} \tag{4}$$

As illustrated in Figure 4, our approach differs fundamentally from standard PPO-based LLM tuning. While conventional PPO uses a frozen initial LLM as a reference regularizer to prevent excessive drift from the initial policy, DGPO employs the teacher LLM as an active guide that steers the student toward correct behaviors when errors occur. This can be seen as a form of targeted regularization (Laroche et al., 2019), which allows free exploration during correct predictions but applies corrective guidance through KL penalties when the student fails. By selectively emphasizing high-divergence incorrect outputs, our method focuses learning on error correction while maintaining autonomous reasoning capabilities, resulting in efficient and stable training.

5 Experiments

5.1 Experimental setup

We focus our experiments on addressing the following questions:

$\mathcal{Q}1$ Do our compact models preserve the overall performance of the teacher model?

$\mathcal{Q}2$ How well do compact models retain individual ARC components? (a) *Source Referencing*, (b) *Query Rewriting*, (c) *Thinking*.

$\mathcal{Q}3$ Which components of our method contribute most to performance improvements?

Datasets. We evaluate DGPO on seven benchmark datasets, categorized as follows: (1) General Question Answering: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA

Qwen 2.5 (3B → 0.5B)	NQ	TriviaQA	PopQA	HotpotQA	2wiki	MuSiQue	Bamboogle	Avg.
Student-0.5B	0.004	0.006	0.007	0.007	0.015	0.000	0.000	0.006
Teacher-3B	0.365	0.569	0.393	0.340	0.368	0.135	0.298	0.353
PPO (Jin et al., 2025)	0.306	0.444	0.379	0.205	0.218	0.041	0.073	0.238
GKD (Agarwal et al., 2024)	0.266	0.408	0.358	0.216	0.217	0.055	0.161	0.240
SeqKD (Kim and Rush, 2016)	0.331	0.416	0.364	0.283	0.273	0.089	0.169	0.275
KD (Hinton et al., 2015)	0.331	0.431	0.373	0.286	0.284	0.091	0.290	0.298
DistiLLM (Ko et al., 2024)	0.333	0.442	0.373	0.288	0.270	0.095	0.209	0.287
TAID (Shing et al., 2025)	0.325	0.427	0.365	0.290	0.270	0.079	0.218	0.282
DGPO (ours)	**0.378**	0.481	**0.402**	**0.342**	0.303	0.120	0.274	0.329

Table 2: **Qwen 2.5 (3B → 0.5B) results** across different methods and QA benchmarks. The best and second-best results are highlighted in green and yellow, respectively. Teacher-surpassing scores are **bold and boxed**.

Model family	Qwen 2.5		Llama 3
Student size	0.5B		1B
Teacher size	3B	7B	8B
Student	0.006	0.006	0.039
Teacher	0.353	0.385	0.438
PPO	0.238	0.238	0.250
KD	0.298	0.280	0.347
DGPO	**0.329**	**0.323**	**0.389**

Table 3: Average EM scores across seven QA benchmarks under different model configurations.

(Mallen et al., 2023) datasets, which generally require single-hop searching, i.e., the answer can be derived from a single fact or passage. (2) Multi-Hop Question Answering: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023) datasets, which require multi-hop searching over multiple evidence across different documents. Please See Appendix B.4 in details.

Base Models. As the base student model, we use Qwen2.5-0.5B-instruct (Qwen et al., 2025). For the teacher model, we adopt Search-R1-PPO-3B based on Qwen2.5-3B-instruct. To assess generalizability across different model sizes and families, we also evaluate variants using Qwen2.5-7B-instruct and Llama 3 (Llama-3.2-1B-Instruct and Llama-3.1-8B-Instruct-based model) (Grattafiori et al., 2024).

Baselines. We compare our method against baselines from three categories:

- *Reinforcement Learning:* Standard PPO (Jin et al., 2025) illustrated in Figure 4 top [2].
- *On-policy Distillation on SGOs:* GKD (Agarwal et al., 2024) minimizes reverse KL divergence between teacher and student distributions on SGOs.
- *Off-policy Distillation on TGOs:* SeqKD (Kim and Rush, 2016) applies SFT on teacher outputs; KD (Hinton et al., 2015) combines cross-entropy loss with KL divergence; DistiLLM (Ko et al., 2024) adopts an adaptive off-policy strategy that integrates both SGOs and TGOs. TAID (Shing et al., 2025) employs dynamic scheduling to interpolate from student to teacher distributions. Off-policy methods, except for DistiLLM, train exclusively on correct TGOs[3] .

Detailed configurations for baseline and ablation variants can be found in Appendix C.

Evaluation Metrics. For all evaluations except the search results shown in Table 5, we use Exact Match (EM) as the evaluation metric, following Jin et al. (2025); Yu et al. (2024).

Retrieval Settings. We follow Jin et al. (2025) and use the 2018 Wikipedia (Karpukhin et al., 2020) as the knowledge source and E5 (Wang et al., 2024) as the retriever. We set the number of retrieved passages to 3.

Training Settings. We used the training sets of NQ and HotpotQA datasets. Training was conducted on NVIDIA 8 × H200 GPUs. Implementation details can be found in Appendix B.

5.2 Main Results ($\mathcal{Q}1$)

Qwen 3B→0.5B. Table 2 shows the overall performance of different methods across seven QA benchmarks. Our method consistently outperforms all baseline methods on most datasets and achieves

[2]We excluded GRPO (Shao et al., 2024) as it proved unstable for compact models, collapsing early due to poor SGOs.

[3]We observed that training on only the correct TGOs led to better performance.

Models Qwen2.5 (3B → 0.5B)	NQ w/o	 w/ thinking	MuSiQue w/o	 w/ thinking
Student-0.5B	0.386	0.034	0.166	0.013
Teacher-3B	0.589	0.560	0.413	0.357
PPO	0.547	0.581	0.258	0.242
KD	0.540	0.544	0.321	0.256
DGPO	0.565	0.593	0.312	0.287

Table 4: Source referencing and thinking performances on NQ and MuSiQue datasets.

Models Qwen2.5 (3B → 0.5B)	NQ (first hop) Hit ratio	MuSiQue (multi-hop) Hit ratio	 Search steps
Student-0.5B	0.004	0.052	3.86
Teacher-3B	0.682	0.668	1.60
PPO	0.711	0.568	1.68
KD	0.675	0.570	2.45
DGPO	0.682	0.583	2.64

Table 5: Query rewriting performance on NQ and thinking performance on MuSiQue datasets.

the highest average score. Remarkably, our method even surpasses the teacher model on three datasets, as selective teacher guidance stabilizes training while RL enables policy improvements beyond teacher imitation. Among the on-policy methods that rely solely on SGOs, both PPO and GKD exhibit lower performance compared to off-policy distillation methods, due to the difficulty of the multi-turn agentic RAG task and the student's near-zero initial performance, which makes SGOs highly noisy. This result highlights the limitations of SGOs, which tend to be noisy and less informative than TGOs. DistiLLM and TAID perform worse than standard KD. In our setting, where the student model starts with extremely low performance, interpolating between the teacher and student distributions might have created noisy or misleading targets, resulting in weaker learning.

Qwen 7B→0.5B and Llama 8B→1B. Table 3 shows the average EM scores for models with a larger capacity gap (Qwen2.5 0.5B and 7B) and another model family (Llama3 1B and 8B). DGPO consistently outperforms both PPO and KD across challenging capacity gaps (7–8B→0.5–1B) and different model architectures (Qwen vs. Llama3). While Qwen 3B→0.5B slightly outperforms Qwen 7B→0.5B due to a smaller capacity gap, DGPO effectively exploits compact model potential regardless of the teacher quality. All results can be found in Appendix D.

5.3 ARC – Source Referencing ($\mathcal{Q}$2a)

Setup. To isolate the capability of Source Referencing from other agentic behaviors, we evaluate the model's accuracy when provided only with the ground-truth supporting contexts (i.e., golden knowledge) as <information>, and forced to answer directly using the <answer> tag. For the MuSiQue dataset, which consists of multi-hop questions requiring multiple supporting documents, we concatenate all relevant ground-truth contexts and supply them as <information>. For the NQ dataset, we use the annotated long answer span as the input <information>. The final answer's correctness is measured using EM.

Results. Table 4 (w/o thinking column) shows the results for source referencing capability. Our model achieves the highest score in extracting information from a single context on the NQ dataset. However, on the MuSiQue dataset, the KD model performs best. One possible explanation is that our RL phase may have over-optimized for simpler, single-step examples during RL, leading to suboptimal performance on complex multi-hop questions.

5.4 ARC – Query Rewriting ($\mathcal{Q}$2b)

Setup. To isolate the Query Rewriting capability from other agentic behaviors, we evaluate whether the initial search query formulated by the model can retrieve documents containing the correct answer, using the NQ dataset. As the evaluation metric, we adopt Hit ratio (Ma et al., 2023), which measures whether at least one of the retrieved documents includes the correct answer.

Results. Table 5 (NQ column) shows the results for query rewriting. Interestingly, the PPO model achieves the best performance, even surpassing the teacher model. Our DGPO performs better than KD but reaches a similar hit ratio to the teacher. This may be attributed to our training setup, which mixes both single-hop and multi-hop datasets. Given the limited capacity of the student model, the PPO agent may have focused its exploration on simpler single-hop query writing tasks, rather than the more complex multi-hop reasoning required in other datasets.

5.5 ARC – Thinking ($\mathcal{Q}$2c)

Setup. To evaluate the Thinking capability, we assess *how* and *when* the model retrieves and integrates information during the reasoning process. *(How:)* We provide the ground-truth contexts as

Method	init (KD)	pipeline	KL penalty	NQ	TriviaQA	PopQA	HotpotQA	2wiki	MuSiQue	Bamboogle	Avg.
DGPO	✓	KD → PPO	selective	0.378	0.481	0.402	0.342	0.303	0.120	0.274	0.329
(a) w/o cold-start initialization	–	KD → PPO	selective	0.370	0.465	0.394	0.330	0.299	0.117	0.266	0.320
(b) w/o selective kl penalty	✓	KD → PPO	uniform	0.362	0.464	0.394	0.323	0.306	0.114	0.234	0.314
(c) w/o teacher guidance	✓	KD → PPO	–	0.353	0.455	0.384	0.316	0.287	0.098	0.250	0.306
(d) invert pipeline order	–	PPO → KD	–	0.320	0.426	0.371	0.287	0.282	0.084	0.234	0.286

Table 6: **Ablation study** on DGPO components. Results show the importance of cold-start initialization, selective KL penalty, teacher guidance during RL, and RL after KD initialization. See Table 12 in Appendix C for more detailed configurations.

<information> and allow the model to perform an additional <think> step immediately after <information> (i.e., the second <think> block in Table 1). Note that such additional thinking was disallowed in the source referencing evaluation ($\mathcal{Q}$2a). While further retrieval is technically unnecessary, the model is still allowed to perform additional search steps. *(When:)* We allow multiple retrieval steps and examine whether the model can determine the necessity of additional searches based on intermediate results. In this case, we evaluate both the final Hit ratio and the average number of search steps taken as metrics of efficiency.

Results. As shown in Table 4 (w/ thinking column), many models, including the teacher, exhibit performance degradation when additional <think> steps are introduced. This suggests that under our smaller model setting, deliberate reasoning through thinking is not crucial for information extraction. Only the RL models improve on the NQ dataset. They may have learned to use thinking to double-check their answers for simpler setting.

As shown in Table 5 (MuSiQue column), while the PPO model performs well in the first retrieval step, our method achieves the highest score for more complex multi-hop reasoning. To achieve higher hit ratios, the distilled model tends to take more search steps. Compared to the teacher, which achieves strong performance with fewer steps due to its larger capacity, our method enables the student to compensate by exploring more extensively.

5.6 Ablation Study ($\mathcal{Q}$3)

Table 6 presents the results of our ablation study. (a) w/o cold-start initialization by KD, the performance drop is relatively small; however, training becomes unstable and collapses around step 800, so we report the score just before the collapse. (b) w/o selective KL penalty applies KL regularization uniformly across all trajectories, regardless of whether the student's attempt is correct or incorrect. (c) w/o teacher guidance denotes KD initialization followed by standard PPO without KL regularization during RL. Both variants (b) and (c) result in performance degradation for our method. (d) Reversing the order (PPO before KD) causes substantial performance loss. These results confirm that all proposed components are essential: KD initialization prevents collapse, pipeline KD→PPO with selective KL penalty is crucial.

6 Conclusion

We propose Distillation-Guided Policy Optimization (DGPO), a novel RL framework that overcomes the core challenge of poor SGOs in compact models via cold-start initialization and selective teacher guidance. DGPO transforms the reference model from a passive regularizer to an active guidance mechanism, enabling performance improvements rather than merely preventing degradation. Our two-phase approach achieves consistent improvements without complex scheduling. Beyond end-to-end gains, our ARC-based analysis provides a fine-grained breakdown of how DGPO improves agentic behavior, highlighting its strengths across dimensions such as source referencing, query rewriting, and multi-hop reasoning.

Can compact language models search like agents? Our findings suggest **yes**. Starting from a 0.5B model with minimal performance (0.006), DGPO achieves a 55× improvement (0.329), approaching the 3B teacher's performance (0.353). Remarkably, our student model even surpasses the teacher on several datasets. Given that 0.5B models can run efficiently on CPUs, our method democratizes access to search agents across computing resource-constrained devices like laptops and smartphones, opening possibilities for more practical agentic RAG deployment. As a foundational study on enabling agentic RAG in compact models, we focus on QA tasks for comprehensive evaluation. Future work will extend this approach to diverse tasks requiring agentic reasoning.

Limitations

Our experiments are restricted to Qwen2.5 (3B→0.5B, 7B→0.5B) and Llama3 (8B→1B) model families. Given the rapid advancement of LLMs, comprehensive evaluation across all available models is impractical within current research timelines. Due to computational limitations, we restrict our investigation to student models of 0.5–1B parameters and teacher models up to 8B parameters. While larger teacher models are available, this work specifically targets compact models for computing resource-constrained environments, making exploration of massive teacher models beyond both our computational capacity and research scope. As stated in Section 5, while our model achieves strong overall performance, optimization across all capacity dimensions remains an open challenge. We believe that our ARC analysis framework and proposed DGPO approach provide essential foundations for enabling compact models to acquire sophisticated agentic behaviors.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small language models are the future of agentic ai. *Preprint*, arXiv:2506.02153.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3290 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. 2023. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. 2019. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3652–3661.

Huanxuan Liao, Shizhu He, Yao Xu, Yuanzhe Zhang, Kang Liu, and Jun Zhao. 2025. Neural-symbolic collaborative distillation: Advancing small language models for complex reasoning tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24567–24575.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5191–5198.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2018. High-dimensional continuous control using generalized advantage estimation. *Preprint*, arXiv:1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

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.

Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. 2025. TAID: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. In *The Thirteenth International Conference on Learning Representations*.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *Preprint*, arXiv:2501.09136.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Xi Wang, Procheta Sen, Ruizhe Li, and Emine Yilmaz. 2025. Adaptive retrieval-augmented generation for conversational systems. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 491–503.

Renjun Xu and Jingwen Peng. 2025. A comprehensive survey of deep research: Systems, methodologies, and applications. *Preprint*, arXiv:2506.12594.

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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chen Zhang, Yang Yang, Jiahao Liu, Jingang Wang, Yunsen Xian, Benyou Wang, and Dawei Song. 2023a. Lifting the curse of capacity gap in distilling language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4535–4553.

Jianyi Zhang, Aashiq Muhamed, Aditya Anantharaman, Guoyin Wang, Changyou Chen, Kai Zhong, Qingjun Cui, Yi Xu, Belinda Zeng, Trishul Chilimbi, and Yiran Chen. 2023b. Reaugkd: Retrieval-augmented knowledge distillation for pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1128–1136.

Appendix

A RL for Agentic RAG

We ground the reinforcement learning framework on the skeletal formalization of Search-R1 (Jin et al., 2025), which is one of the state-of-the-art agentic RAG frameworks. We model the agentic search process as a sequential decision-making problem where the LLM agent must learn to coordinate reasoning and retrieval operations. At each step, the agent can either generate text to advance its reasoning or issue queries to the external search engine $\mathcal{R}$ to gather additional information.

Learning Objective. The Reinforcement Learning for agentic RAG framework is formulated as:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x;\mathcal{R})}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\text{KL}}\left[\pi_\theta(y \mid x;\mathcal{R}) \,||\, \pi_{\text{ref}}(y \mid x;\mathcal{R})\right], \quad (5)$$

where π_θ denotes the trainable agent policy that generates action trajectories y conditioned on the input user question $\boldsymbol{x}$ and an external retrieval system $\mathcal{R}$. The reward function $r(\boldsymbol{x},\boldsymbol{y})$ evaluates accuracies of generated answers. The KL-divergence term with coefficient β provides regularization against the frozen reference policy π_{ref}.

B Implementation Details

B.1 Token Masking

Following prior work (Jin et al., 2025), we employ token masking during training. Eq. (2), $\mathbb{1}(y_t)$ is the loss-masking operator defined as,

$$\mathbb{1}(y_t) = \begin{cases} 1 & \text{if } y_t \in \{\texttt{LLM-generated tokens}\} \\ 0 & \text{if } y_t \in \{\texttt{external tokens}\}\,. \end{cases} \quad (6)$$

In agentic RAG, the token sequence contains both LLM agent-generated tokens (<search> , <think> , and <answer>) and externally retrieved content from the search system $\mathcal{R}$ (<information>). Computing gradients over retrieved tokens is counterproductive, as it encourages the model to learn how to generate external content rather than focusing on the core agentic capabilities of when and how to search. To prevent this misallocation of model capacity and stabilize training, we apply loss masking to retrieved tokens and documents, ensuring optimization focuses solely on agent-generated content.

B.2 Prompt Template

We used the system template for Qwen2.5 series and the instruction template following Jin et al. (2025). Table 7 shows these templates.

System Template for qwen2.5 series
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.

Instruction Template
Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search> , and it will return the top searched results between <information> and </information> . You can search as many times as you want. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer> without detailed illustrations. For example, <answer> xxx </answer> . Question: question.

Table 7: System and instruvtion template for agentic RAG. question is replaced with the specific question during training and inference.

Qwen 2.5 (7B → 0.5B)	NQ	TriviaQA	PopQA	HotpotQA	2wiki	MuSiQue	Bamboogle	Avg.
Student-0.5B	0.004	0.006	0.007	0.007	0.015	0.000	0.000	0.006
Teacher-7B	0.393	0.610	0.397	0.370	0.414	0.146	0.368	0.385
PPO (Jin et al., 2025)	0.306	0.444	0.379	0.205	0.218	0.041	0.073	0.238
KD (Hinton et al., 2015)	0.338	0.428	0.371	0.288	0.223	0.100	0.210	0.280
DGPO (ours)	0.371	0.474	0.396	0.334	0.257	0.113	0.315	0.323

Table 8: **Qwen 2.5 (7B → 0.5B) results across different methods and QA benchmarks**. The best and second-best results are highlighted in green and yellow, respectively.

Llama 3 (8B → 1B)	NQ	TriviaQA	PopQA	HotpotQA	2wiki	MuSiQue	Bamboogle	Avg.
Student-1B	0.052	0.080	0.044	0.027	0.042	0.001	0.024	0.039
Teacher-8B	0.475	0.647	0.448	0.427	0.443	0.179	0.444	0.438
PPO (Jin et al., 2025)	0.354	0.499	0.394	0.222	0.181	0.037	0.065	0.250
KD (Hinton et al., 2015)	0.406	0.508	0.405	0.369	0.355	0.119	0.266	0.347
DGPO (ours)	0.448	0.553	0.437	0.412	0.379	0.155	0.339	0.389

Table 9: **Llama 3 (8B → 1B) results across different methods and QA benchmarks**. The best and second-best results are highlighted in green and yellow, respectively.

Config	Parameter	Value
RL parameters	Total training steps	1000
	Batch size	512
	KL divergence coefficient β	0.001
	Maximum prompt length	4096
	Maximum response length	500
	Maximum conversation turns	4
	Top-k retrieved documents	3
	Actor learning rate	1e-6
	Critic learning rate	1e-5
KD parameters (initialization)	Tortal epochs	5
	Batch size	64
	Learning rate	1e-4
	KL divergence ratio λ	1.0
DistiLLM (Ko et al., 2024)	Skew KLD target weight	0.1
TAID (Shing et al., 2025)	t_{start}	0.4
	t_{end}	1.0
	Updating interpolation (α)	5e-4
	Momentum coefficient (β)	0.99

Table 10: Parameters for DGPO and baselines.

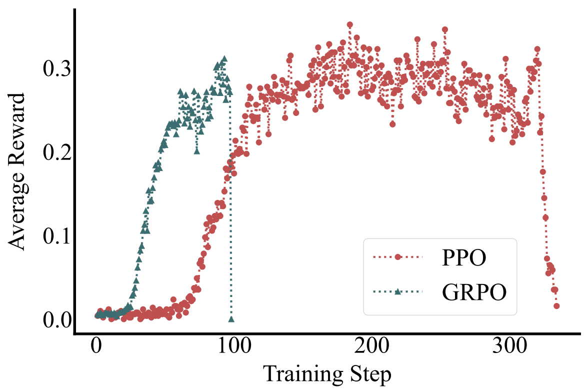

Figure 5: Training curve of PPO and GRPO.

B.3 Training Details

On-policy distillation or RL methods were trained for up to 1000 steps. However, PPO training with a small model is inherently unstable; thus, we report the results at step 200, before training collapse. All models were initialized from the same pretrained checkpoints and trained once. Training took approximately one day on 8×H200 GPUs. The hyperparameters and libraries used for implementation followed those of prior work (Jin et al., 2025; Shing et al., 2025). Table 10 shows training parameters.

B.4 Dataset Details

We used preprocessed seven QA datasets following Jin et al. (2025). Table 11 shows dataset statistics. These datasets are originally designed for QA tasks, and our use aligns with their intended purpose.

B.5 Source Code

The source code is available at: `https://anonymous.4open.science/r/DGPO`.

C Ablation and Baseline Settings

Table 12 summarizes the ablation and baseline settings used in our study, indicating which components (e.g., KD, PPO loss, GRPO loss, selective or uniform KL penalties) are included in each variant, along with references to the corresponding figures or tables where results are reported.

Dataset	Training samples	Test samples	License
Natural Questions (NQ) (Kwiatkowski et al., 2019)	79,168	3,610	CC BY-SA 3.0
TriviaQA (Joshi et al., 2017)	–	11,313	Apache-2.0
PopQA (Mallen et al., 2023)	–	14,267	MIT
HotpotQA (Yang et al., 2018)	90,447	7,405	CC BY-SA 4.0
2WikiMultiHopQA (Ho et al., 2020)	–	12,576	Apache-2.0
MuSiQue (Trivedi et al., 2022)	–	2,417	CC BY 4.0
Bamboogle (Press et al., 2023)	–	125	MIT

Table 11: Statistics of training and test datasets.

Setting	Results	KD (initialization)	PPO Loss	GRPO Loss	Selective KL penalty	Uniform KL penalty
DGPO	Tab. 2	✓	✓		✓	
w/ GRPO	Fig. 6	✓		✓	✓	
w/o cold-start initialization	Tab. 6		✓		✓	
w/o selective KL penalty (uniform KL penalty)	Tab. 6	✓	✓			✓
w/o teacher guidance (KD→PPO)	Tab. 6	✓	✓			
invert pipeline order (PPO→KD)	Tab. 6	✓	✓			
KD→GKD	Fig. 6	✓				✓
PPO (Jin et al., 2025)	Tab. 2		✓			
KD (Hinton et al., 2015)	Tab. 2	✓				
GKD (Agarwal et al., 2024)	Tab. 2					✓

Table 12: Ablation and baseline settings and their components.

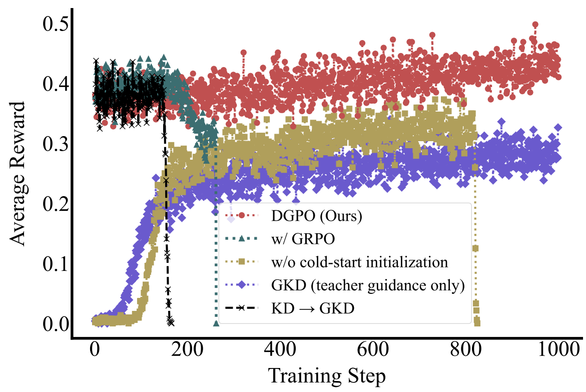

Figure 6: Training curves comparing DGPO and its ablations: (1) GRPO version; (2) without cold-start initialization; (3) GKD; and (4) KD→GKD.

D Evaluation across Different Model Families and Larger Capacity Gaps

We evaluate our method across different model families and larger capacity gaps using state-of-the-art compact models (0.5–1B stduents). Table 8 shows Qwen2.5 (7B $\rightarrow$ 0.5B) results and Table 9 shows Llama 3 (8B $\rightarrow$ 1B) results. Both configurations demonstrate DGPO's consistent superiority over baseline methods, confirming broad applicability across diverse architectures and capacity gaps.

E Training Dynamics

E.1 Performance Plateau in Compact Models.

Figure 5 presents the RL training curves of Qwen2.5-0.5B-instrtuct model with PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) for agentic RAG. Smaller models converge faster but tends to become unstable relatively early in training (Jin et al., 2025), preventing further performance gains beyond that point. PPO provides more stable optimization than GRPO but converges slower.

E.2 DGPO and Its Variants

Figure 6 illustrates the training stability of DGPO and its variants across different RL algorithms and initialization strategies. DGPO maintains a stable training curve beyond 1000 steps, achieving the best overall performance. However, (1) replacing PPO with GRPO leads to an early collapse during RL. Even with KD initialization and teacher guidance, GRPO remains unstable for compact models. (2) When removing KD initialization from our model, training remains more stable until 800 steps compared to the standard PPO but collapses at around 800 steps. (3) Using GKD, i.e., teacher guidance only, results in stable learning; however, the absence of self-exploration in RL leads to significant worse performance. (4) When KD-based initialization is further combined with GKD, training collapses prematurely due to the excessive constraints imposed by the teacher.