PaperScout: An Autonomous Agent for Academic Paper Search with Process-Aware Sequence-Level Policy Optimization

Anonymous ACL submission

Abstract

Academic paper search is a fundamental task in scientific research, yet most existing approaches rely on predefined workflows, which limits their flexibility when handling complex queries. We propose PaperScout, an autonomous agent that formulates paper search as a sequential decision-making process, enabling the agent to dynamically decide when and how to invoke search and reference expansion actions over multiple turns. To train such an agent stably, we introduce Proximal Sequence Policy Optimization (PSPO), a process-aware, sequence-level policy optimization method that aligns optimization with agent–environment interaction. Comprehensive experiments show that PaperScout trained with PSPO achieves superior retrieval performance compared to existing methods. Our code is publicly available at `https://anonymous.4open.science/r/PaperScout-4BC6`.

1 Introduction

Academic paper search is a fundamental task in scientific research that underpins effective knowledge discovery (Timmins and McCabe, 2005; Marchionini, 2006). Most traditional approaches rely on lexical or semantic matching, representing articles with sparse keyword-based features or dense semantic embeddings over largely static corpora (Vine, 2006; Zhu et al., 2023). When a search query is issued, relevant papers are retrieved through nearest-neighbor match in the resulting representation space. Such approaches are generally effective when queries are well-formed and relatively simple (Shi et al., 2025).

However, as scholarly publications continue to grow rapidly and originate from increasingly heterogeneous sources, maintaining static academic databases becomes increasingly costly (Manghi, 2024). Meanwhile, traditional methods exhibit limited capability when faced with fine-grained and conditional queries (Gusenbauer and Haddaway, 2020). For instance, a researcher may seek studies such as *"Apply reinforcement learning to protein folding while excluding Transformer-based architectures"*, which requires cross-domain reasoning and constraint-aware filtering. Existing semantic retrieval approaches often return thematically related papers but fail to satisfy such conditions, forcing repeated query reformulation and manual screening, and substantially increasing search cost (Gusenbauer and Haddaway, 2021).

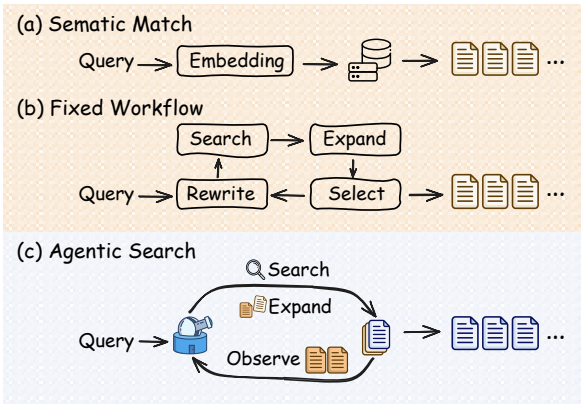


Figure 1: Comparison of different academic search paradigms: Semantic Match, Fixed Workflow, and Agentic Search.

Recent advances in large language models (LLM) have introduced new opportunities for academic paper search. Going beyond single-round querying and static ranking, LLMs leverage accumulated context to guide subsequent steps and produce structured outputs, such as rewritten queries, candidate inspection criteria, and citation cues (Zhu et al., 2023), thereby enabling fine-grained control over the retrieval process. Unlike traditional semantic matching, this interaction-driven retrieval better reflects practical search processes, effectively handling complex queries and gradually narrowing the search space (Shi et al., 2025).

Despite the increased expressive capacity of LLM-based approaches for academic paper search, most existing methods remain fundamentally workflow-driven. These approaches decompose retrieval into predefined stages, such as query rewriting, search, and reference expansion, and rely on fixed execution logic to determine when and how each stage is applied. As a result, retrieval decisions are implicitly encoded in workflow design rather than being made autonomously based on the evolving search context. This design assumes that a single retrieval paradigm can accommodate diverse and evolving queries, limiting the system's ability to adapt its retrieval behavior as new information is accumulated over time.

To address the above limitations, we propose **PaperScout**, an autonomous paper search agent that reframes retrieval as a sequential decision-making process rather than predifined workflow execution. Instead of following a fixed search–expand pipeline, PaperScout explicitly decides at each step *whether*, *when*, and *how* to invoke search and reference expansion actions based on the papers accumulated so far. By granting the agent direct control over retrieval decisions, PaperScout enables flexible, context-dependent exploration of paper, allowing search strategies to evolve dynamically as the retrieval process unfolds.

However, training such an autonomous multi-turn retrieval agent poses a fundamental optimization challenge. The mismatch between interaction granularity and token-level optimization assumptions dilutes learning signals during propagation, thereby increasing credit assignment uncertainty and making value function estimation more difficult to stabilize. To address this granularity mismatch, we propose **Proximal Sequence Policy Optimization (PSPO)**, a process-aware, sequence-level policy optimization method that performs advantage estimation at the sequence level while explicitly leveraging process feedback, enabling stable and efficient training of multi-turn retrieval agents. Our main contributions are as follows:

- We formulate academic paper search as a partially observable sequential decision-making problem and propose PaperScout, an autonomous agent that adaptively controls search and reference expansion based on accumulated retrieval context.
- We introduce Proximal Sequence Policy Optimization (PSPO), a process-aware, sequence-level policy optimization method that aligns optimization granularity with multi-turn agent interactions.
- Extensive experiments and behavioral analyses demonstrate more efficient multi-turn retrieval and improved optimization stability over existing methods.

2 Related Work

2.1 Query-Centric Academic Search

Traditional academic paper search methods predominantly follow a query-centric retrieval setting, in which retrieval is driven by a single user query and focuses on single-round query–document relevance modeling (Schütze et al., 2008; Guo et al., 2020). Early approaches retrieval papers by measuring term-level similarity between the query and documents (Qaiser and Ali, 2018). To mitigate the limitations of semantic mismatch, later work moves toward dense retrieval, which improve query–document matching by encoding queries and documents into semantic vector representations (Jiang et al., 2019; Reimers and Gurevych, 2019). More recently, LLMs have been employed to rewrite or expand queries prior to retrieval in order to better capture user intent (Ma et al., 2023; Gusenbauer and Haddaway, 2020). Despite these advances, such methods still treat paper search as a single-round query–document matching, which limits their ability to handle complex and diverse search queries (Gusenbauer and Haddaway, 2020).

2.2 Multi-Turn Academic Paper Search

To move beyond the query-centric, single-round retrieval paradigm, recent studies have explored multi-turn approaches for academic paper search (Cheng and Suo, 2025; Aytar et al., 2025; Park et al., 2025). These methods typically formulate retrieval as a sequence of operations, enabling systems to iteratively search, inspect candidate papers, and expand along citation links to progressively collect relevant papers. Representative work such as PaSa (He et al., 2025) applies reinforcement learning to optimize specific search and expand behaviors, while SPAR (Shi et al., 2025) adopts a modular framework that integrates query rewriting, reference exploration, and result re-ranking to improve retrieval performance for complex queries. Despite their effectiveness, these approaches generally use predefined retrieval workflows, offering

limited flexibility in adapting retrieval strategies to different queries or evolving search contexts.

2.3 Reinforcement Learning for LLM

Most reinforcement learning methods for LLMs are originally developed for single-turn generation. Proximal Policy Optimization (PPO) (Schulman et al., 2017) is a widely used baseline, but its reliance on a learned value function makes performance sensitive to critic accuracy, particularly under long-horizon and sparse-feedback settings (Yuan et al., 2025). To mitigate this issue, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) replaces critic-based advantages with group-wise reward comparisons, while Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) further extends this idea to sequence-level optimization through importance weighting and clipping over complete responses. Despite these advances, multi-turn agents pose an additional granularity mismatch: agent–environment interaction occurs at the level of full responses, while feedback may include both outcome-level and intermediate process signals. Group-based trajectory methods mainly emphasize final outcome comparisons, whereas token-level PPO updates are misaligned with sequence-level interaction and feedback. This gap motivates policy optimization methods that operate at the same granularity as agent–environment interaction while effectively incorporating intermediate process-level rewards.

3 Methodology

In this section, we present PaperScout, an autonomous LLM-based agent for academic paper search. An overview of the framework is shown in Figure 2. We first formalize paper search as a Partially Observable Markov Decision Process (POMDP) in Section 3.1. Building on this formulation, Section 3.2 instantiates each POMDP component in a concrete multi-tool agent system. To optimize the agent policy under sequence-level feedback, Section 3.3 introduces Proximal Sequence Policy Optimization (PSPO).

3.1 Problem Definition

We formalize academic paper search as a Partially Observable Markov Decision Process (POMDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O}, \gamma \rangle$. The state $s_t \in \mathcal{S}$ represents a latent *paper pool* that contains all papers accumulated up to step t. Due to limited context, the agent cannot access s_t directly and instead receives an observation $o_t = \mathcal{O}(s_t) \in \Omega$, which provides a partial view of the pool (e.g., a small subset of highly relevant papers). Based on o_t, the agent chooses an action $a_t \in \mathcal{A}$ to acquire new information. In our setting, an action is instantiated by invoking one or more external retrieval tools (such as web search or reference expansion), whose execution returns additional candidate papers. The transition $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ updates the paper pool by incorporating newly retrieved papers, and the reward $r_t = \mathcal{R}(s_t, a_t)$ measures the marginal utility of this acquisition with respect to the user query. This formulation casts paper search as an iterative decision-making problem rather than a one-shot ranking task.

Tool	Description
`Search(query)`	Calls scholarly search APIs to retrieve new papers.
`Expand(paper)`	Retrieves reference papers cited by the input paper.

Table 1: Tools available to the agent.

3.2 PaperScout

PaperScout instantiates the POMDP in a modular agentic system (Figure 2, left). While Section 3.1 provides an abstract formulation, we now describe the concrete design for state tracking, observation construction, and policy execution.

State and Observation Space. We implement the latent state s_t as a *paper pool* $\mathcal{B}_t$. Each paper in $\mathcal{B}_t$ stores (i) its content and metadata, (ii) a relevance score $\rho(\cdot)$ with respect to the user query, and (iii) an expansion flag indicating whether its references have been explored. Together, $\mathcal{B}_t$ maintains the agent's current search frontier.

The observation o_t summarizes $\mathcal{B}_t$ under the agent's limited context budget. We use a dual-list view that partitions top-ranked papers into (a) expanded papers, which provide stable context from previously explored nodes, and (b) unexpanded papers, which are candidates for further exploration. We further augment o_t with the interaction history $\mathcal{H}_{t-1}$ to reduce redundant exploration, including past queries and previously expanded papers.

Policy and Execution Mechanism. The environment provides a structured observation o_t, which we serialize into the LLM input sequence x_t using a fixed prompt template together with the

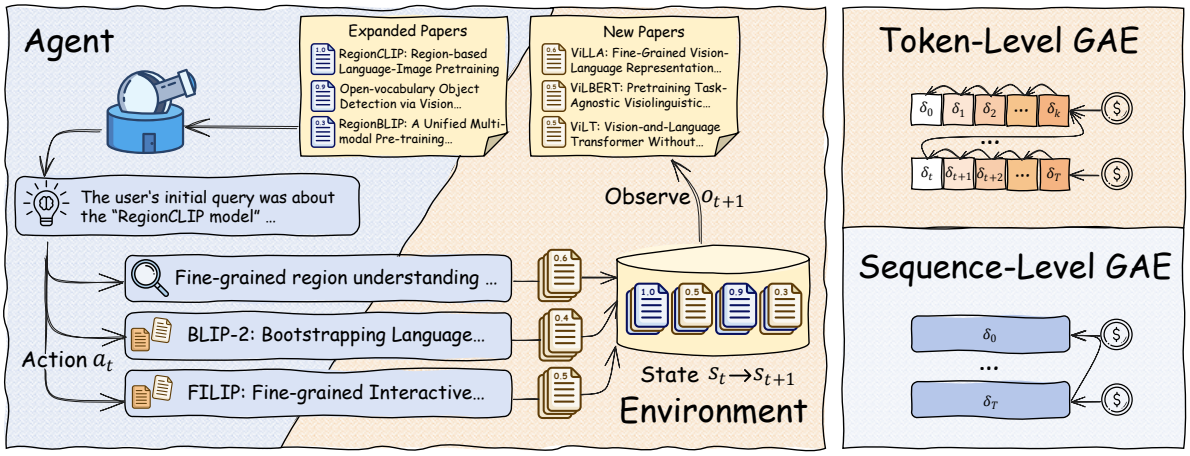


Figure 2: **Overview of PaperScout and the motivation for PSPO. Left:** PaperScout models multi-turn paper search as a POMDP by maintaining a paper pool as the latent state; at each step, the agent observes a summarized view of the pool and issues Search or Expand tool calls to retrieve new papers, which are then merged to update the pool. **Right:** *Granularity mismatch* in credit assignment: PPO attributes a single step reward to many tokens in the response, leading to diluted learning signals, whereas PSPO treats each complete response as the atomic action and performs advantage estimation and policy updates at the sequence level.

user query and tool description. The LLM then samples an output sequence $y_t \sim \pi_\theta(\cdot \mid x_t)$. We parse y_t into $y_t = (z_t, \mathcal{C}_t)$, where z_t is a brief reasoning trace and $\mathcal{C}_t$ is a set of tool invocations defined in Table 1. The environment executes $\mathcal{C}_t$ and returns raw results $\mathcal{D}(\mathcal{C}_t)$. We then filter these results to obtain the accepted candidate set $\mathcal{V}_t$:

$$\mathcal{V}_t = \{\, p \in \mathcal{D}(\mathcal{C}_t) \;\mid\; \rho(p) \geq \tau \;\wedge\; p \notin \mathcal{B}_t \}, \quad (1)$$

where τ is a minimum relevance threshold. Only papers in $\mathcal{V}_t$ are merged into the pool, yielding the state transition $s_t \to s_{t+1}$.

3.3 Proximal Sequence Policy Optimization

To effectively train the agent within the POMDP framework, we introduce Proximal Sequence Policy Optimization (PSPO). This section details the reward formulation and the core optimization algorithm designed to bridge the gap between token-level generation and sequence-level utility.

Reward Formulation. PaperScout aims to maximize the recall of relevant papers. We design the reward so that the expected return matches the expected amount of relevant papers acquired over the search process. Let $\mathcal{V}_t$ denote the set of novel papers accepted into the pool at step t. To reduce variance, we compute the gain on the top-k papers in $\mathcal{V}_t$ ranked by relevance (using all papers if $|\mathcal{V}_t| < k$). The step reward is:

$$r_t = \underbrace{\sum_{p \in \text{top-}k(\mathcal{V}_t)} \rho(p)}_{\text{relevance gain}} - \underbrace{\eta \sum_{c \in \mathcal{C}_t} \mathbb{I}(c \in \mathcal{H}_{t-1})}_{\text{repetition penalty}}, \quad (2)$$

where $\rho(p) \in [0, 1]$ is the relevance score of paper p, interpreted as the probability that p is relevant to the user query. With this interpretation, the cumulative relevance gain corresponds to the expected number of relevant papers retrieved, while the repetition term discourages redundant tool usage.

Algorithm Implementation. In our setting, the agent interacts with the environment at the *sequence level*: each step corresponds to generating a complete response that triggers tool executions and yields a single step reward. This induces a granularity mismatch for token-level optimization, as illustrated in Figure 2: sparse sequence-level feedback must be attributed to tokens, leading to noisy credit assignment and unstable value learning.

PSPO resolves this mismatch by optimizing policies at the sequence level. Let x_t be the prompt obtained by serializing the observation o_t, and y_t be the complete response at step t. For a trajectory $\mathcal{T} = \langle x_0, y_0, \ldots, x_{T-1}, y_{T-1} \rangle$ of length T, we define $r_t = r(x_t, y_t)$ and learn a value function $V_\phi(x_t)$. We estimate advantages with GAE:

$$\hat{A}_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \, \delta_{t+l}, \quad (3)$$

where $\delta_t = r(x_t, y_t) + \gamma V_\phi(x_{t+1}) - V_\phi(x_t)$ denotes the temporal-difference (TD) error.

The actor is trained with a clipped surrogate objective:

$$\mathcal{L}_{\text{actor}}(\theta) = \mathbb{E}_{\mathcal{T}\sim\pi_{\theta_{\text{old}}}}\Big[\min\big(w_t(\theta)\hat{A}_t, \text{clip}(w_t(\theta), 1-\varepsilon_{low}, 1+\varepsilon_{high})\hat{A}_t\big)\Big], \tag{4}$$

where the sequence-level importance ratio is

$$w_t(\theta) = \frac{\pi_\theta(y_t \mid x_t)}{\pi_{\theta_{\text{old}}}(y_t \mid x_t)} = \prod_{i=1}^{L_t} \frac{\pi_\theta(y_{t,i} \mid x_t, y_{t,<i})}{\pi_{\theta_{\text{old}}}(y_{t,i} \mid x_t, y_{t,<i})}. \tag{5}$$

The critic minimizes $\mathbb{E}\big[(V_\phi(x_t) - R_t)^2\big]$ with

$$R_t = \sum_{k=0}^{T-t-1} \gamma^k r_{t+k}. \tag{6}$$

Optimization Strategies. Training the critic is particularly challenging because reward distributions vary widely across queries. Although the reward design already controls variance via filtering and top-k aggregation, the return scale can still differ substantially across samples, which slows value learning and destabilizes advantage estimation. To address this, two stabilizers are introduced primarily for the critic: value pre-training and return normalization. Specifically, we initialize the value function with value pre-training as in VAPO (Yue et al., 2025), training the critic offline under a fixed policy to reduce initialization bias. We further apply running mean–variance normalization to the discounted returns R_t, making value regression and subsequent policy updates less sensitive to reward magnitude. Finally, to encourage exploration, we adopt asymmetric clipping as in DAPO (Yu et al., 2025), using separate lower and upper bounds ε_{low} and $\varepsilon_{\text{high}}$.

4 Experiments

4.1 Experiment Setup

We conduct experiments on two benchmarks: AutoScholarQuery and RealScholarQuery. For reinforcement learning, the agent is trained on the AutoScholarQuery training split and validated on the development split. We evaluate on two test sets: (i) a filtered subset of the AutoScholarQuery test split containing 112 queries with at least five ground-truth relevant papers, and (ii) RealScholarQuery, which comprises 50 real-world scholarly queries with corresponding reference papers.

Retrieval Backends. To enable stable RL training under high concurrency, we build a fully local retrieval environment for training. Specifically, we deploy Milvus[1] over millions of paper metadata to emulate a scholarly search service, and precache millions of papers from ar5iv[2] to support reference-based expansion. If a requested paper is missing from the local cache, it would be downloaded on demand from ar5iv or arXiv[3]. During evaluation, we standardize the search backend by issuing search calls through Google Search for fair comparison across methods.

Baselines. We compare against a diverse set of baselines spanning traditional academic search engines, LLM-enhanced retrieval workflows, and recent multi-turn paper search systems:

- **Google Search** [4]**.** Standard Google Search using the original query.
- **Google Search + LLM.** Google Search with a refined query generated by LLM.
- **Google Scholar.** Direct retrieval from Google Scholar without LLM intervention.
- **PaSa (He et al., 2025).** A reinforcement learning–based paper search system that optimizes query rewriting and reference expansion under a fixed workflow .
- **SPAR (Shi et al., 2025).** A multi-turn academic search framework that integrates multiple retrieval components within a predefined structured workflow.

Our variants. We evaluate three variants of our approach to disentangle the effects of RL fine-tuning and model scale: (i) PaperScout (Qwen3-4B fine-tuned with PSPO), (ii) PaperScout-Qwen3-4B (the same backbone without RL fine-tuning), and (iii) PaperScout-Qwen3-Max (a larger backbone, Qwen3-Max, without RL fine-tuning).

Evaluation Protocol. For Google Search, Google Search + LLM, and Google Scholar, we evaluate the top 100 retrieved papers. For other

[1] https://milvus.io
[2] https://ar5iv.labs.arxiv.org
[3] https://arxiv.org
[4] https://serper.dev

Table 2: Performance comparison on the RealScholarQuery and AutoScholarQuery benchmarks. Best results are highlighted in **bold** and second-best results are underlined. Note that PaperScout develops based on Qwen3-4B.

Model	RealScholarQuery				AutoScholarQuery			
	Precision	F1	Recall	LLM-score	Precision	F1	Recall	LLM-score
Google Search	0.059	0.074	0.304	1.116	0.009	0.017	0.127	1.263
Google Search + LLM	0.067	0.077	0.254	1.237	0.010	0.019	0.138	1.322
Google Scholar	0.045	0.064	0.247	1.251	0.006	0.011	0.081	1.212
PaSa	0.415	0.417	0.541	2.111	0.095	0.129	0.442	2.186
SPAR	0.412	0.408	0.496	2.415	0.091	0.131	0.386	2.295
PaperScout-Qwen3-4B	0.404	0.411	0.497	2.261	0.092	0.128	0.382	2.202
PaperScout-Qwen3-Max	0.435	0.427	0.562	2.483	0.102	0.115	0.427	2.368
PaperScout	**0.442**	**0.441**	**0.574**	**2.576**	**0.115**	**0.134**	**0.459**	**2.467**

multi-turn methods, we use each system's final reranked and filtered outputs. For PaperScout, candidate papers are ranked by the relevance score $\rho(\cdot)$, and we retain papers with $\rho(p) \geq 0.5$ as the final results.

Metrics. We report Precision, Recall, F1-score, and an LLM-based relevance score. Precision, Recall, and F1-score are computed by exact matches between retrieved papers and ground-truth relevant papers. To mitigate evaluation bias caused by incomplete annotations, we additionally report an LLM-based score: three LLMs (DeepSeek-V3.2 (Liu et al., 2024), Qwen3-Max (Yang et al., 2025), and GPT-5.1) independently judge the relevance between each retrieved paper and the user query. Relevance is rated on a four-level scale (0–3), and we average the three ratings to obtain a final score in $[0, 3]$. Details of the scoring protocol and reliability analysis are provided in Appendix A.2.

4.2 Main Results

Table 2 summarizes the performance on RealScholarQuery and AutoScholarQuery. Across both benchmarks, multi-turn retrieval methods substantially outperform single-shot baselines, highlighting the importance of iterative tool use for academic paper search under complex query settings.

Among all multi-turn systems, PaperScout consistently achieves the strongest performance. On RealScholarQuery, PaperScout attains the highest Recall of 0.574 (vs. 0.541 for the best baseline) together with the best LLM-score of 2.576, and improves the best baseline F1-score from 0.417 to 0.441 (+5.8% relative). On AutoScholarQuery, PaperScout again ranks first, yielding the highest Recall of 0.459 and highest LLM-score of 2.467, while also achieving the best F1-score (0.134).

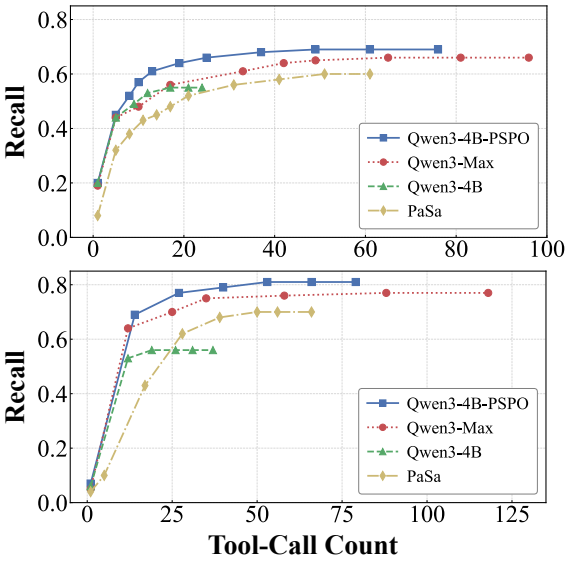


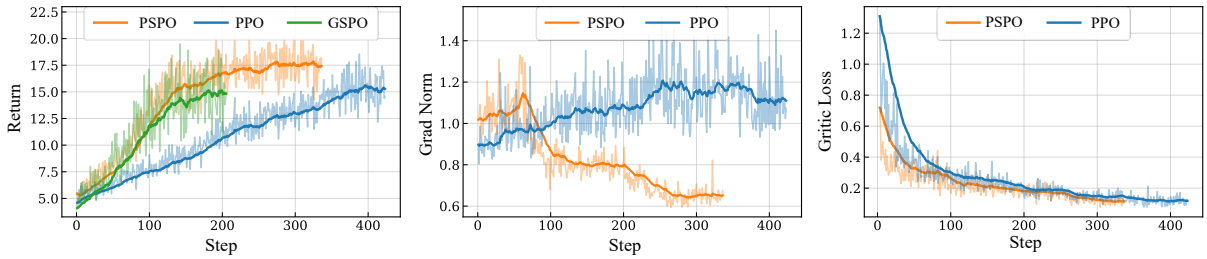
Figure 3: Average recall as tool calls accumulate on RealScholarQuery (top) and AutoScholarQuery (bottom). PSPO consistently yields higher recall with the same number of tool calls.

Notably, the RL-trained 4B PaperScout even matches or surpasses a larger untrained backbone (PaperScout-Qwen3-Max), motivating further analysis of its tool-call efficiency and optimization behavior in the following sections.

4.3 Analysis of PaperScout

To analyze the strong retrieval performance of PaperScout, we examine its behavior from the perspectives of tool-call efficiency and flexibility in multi-turn retrieval.

Tool-Call Efficiency. Figure 3 illustrates how recall evolves as tool calls accumulate on both benchmarks. Overall, PaperScout achieves consistently higher recall than all baselines under the same number of tool calls, indicating a larger re-



(a) Return of Trajectory (b) Actor Gradient Norm (c) Critic Loss

Figure 4: Training dynamics and optimization statistics for different policy optimization methods. (a) Trajectory returns during training. PSPO converges faster and reaches a higher return plateau than PPO and GSPO. (b) Actor gradient norm during policy updates. PSPO maintains a smaller gradient norm with a clearer downward trend than PPO. (c) Critic loss during training. PSPO is consistently lower than PPO, especially in early training.

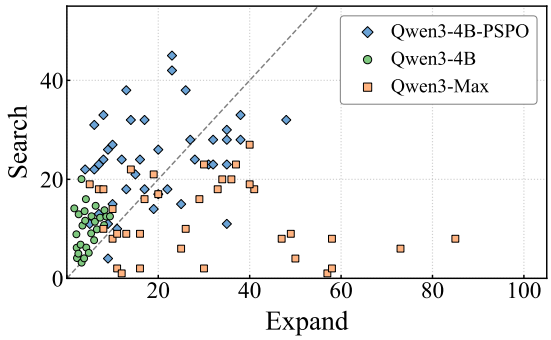


Figure 5: Search–Expand call distribution across queries for different models. The trained PaperScout (Qwen3-4B-PSPO) exhibits a broader and more balanced allocation of search and expansion actions compared with the untrained Qwen3-4B and Qwen3-Max, indicating more flexible multi-turn retrieval behavior.

trieval gain per invocation. In the low-call regime, the trained PaperScout agent exhibits a steeper increase in recall compared with PaSa and the untrained 4B variant, suggesting more effective early-stage multi-turn retrieval. As tool calls increase, the untrained 4B agent quickly saturates, whereas PaperScout continues to accumulate relevant papers and maintains a clear advantage throughout the trajectory. Notably, PaperScout matches or surpasses the much larger Qwen3-Max backbone across a wide range of tool-call counts on both datasets, demonstrating that efficient multi-turn retrieval behavior can compensate for model scale through more informative tool usage.

Analysis of Tool Distribution. Figure 5 further analyzes how PaperScout allocates search and expansion actions across queries. The untrained Qwen3-4B agent concentrates near the origin, indicating that it issues few tool calls of either type and thus performs limited multi-turn exploration.

Table 3: Performance comparison of PaperScout optimized by PPO, GSPO and PSPO on RealScholarQuery.

Model	Precision	F1	Recall	LLM-score
PPO	0.405	0.408	0.537	2.417
GSPO	0.433	0.439	0.557	2.510
PSPO	**0.442**	**0.441**	**0.574**	**2.576**

In contrast, Qwen3-Max exhibits a pronounced skew toward expansion, with many trajectories dominated by expansion-heavy behaviors and relatively few search calls, which can restrict the introduction of new retrieval directions. By comparison, PaperScout occupies a broader region of the search–expand space and distributes calls more evenly between the two actions, reflecting a more balanced trade-off between retrieval breadth and depth. Overall, these patterns indicate that PaperScout supports flexible and adaptive multi-turn retrieval strategies, rather than following fixed or biased tool-use behaviors across queries.

4.4 Effectiveness of PSPO

To evaluate the effectiveness of PSPO as a policy optimization method for multi-turn retrieval, we analyze its impact from two complementary perspectives: final retrieval quality and optimization stability, in comparison with PPO and GSPO.

Superior Retrieval Performance. Table 3 reports the retrieval performance on RealScholarQuery. PSPO achieves the best results across all metrics, improving Recall to 0.574 (vs. 0.557 with GSPO and 0.537 with PPO) and yielding the highest LLM-score of 2.576. These gains are consistent with the training dynamics shown in Figure 4(a), where PSPO converges faster and reaches a higher return plateau than both PPO and GSPO.

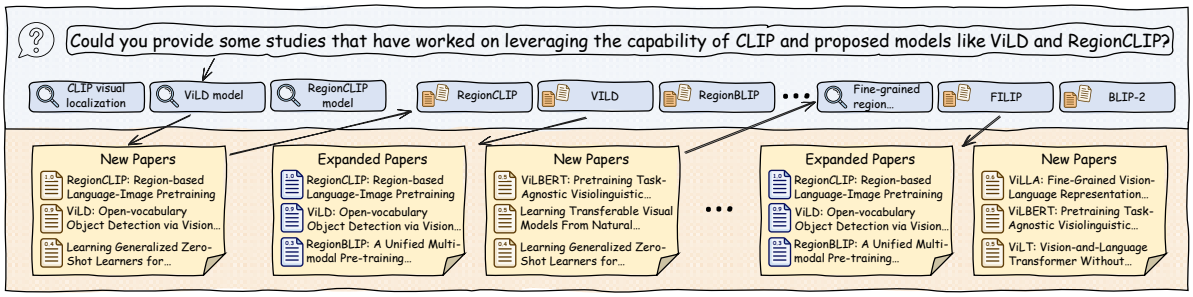


Figure 6: Case study of PaperScout. The agent alternates between search and reference expansion based on accumulated retrieval context, re-issuing search when expansion yields diminishing returns to introduce new directions.

The slower improvement of PPO can be attributed to the token–sequence granularity mismatch in multi-turn retrieval: optimizing policies at the token level under sparse sequence-level feedback amplifies credit assignment noise and makes value fitting more difficult, thereby reducing sample efficiency despite relatively smooth learning curves. GSPO, by contrast, performs sequence-level optimization and thus achieves rapid early gains; however, it exhibits larger return oscillations and an earlier performance ceiling, suggesting limited stability when intermediate process rewards need to be incorporated. PSPO bridges these behaviors by aligning policy optimization with the sequence-level interaction granularity while explicitly leveraging process rewards through the critic, leading to both fast convergence and stable performance improvements.

Stable Model Optimization. Figure 4(b) presents the actor gradient norm during training. Compared with PPO, PSPO maintains a smaller gradient norm with a clear downward trend, indicating more controlled and stable policy updates under sequence-level optimization. Figure 4(c) further shows that PSPO consistently achieves a lower critic loss than PPO, particularly in the early training stage, suggesting that its advantage estimation provides more reliable learning targets for value regression under sparse sequence-level feedback. Taken together, these observations indicate that PSPO improves optimization stability for both the actor and critic, which in turn contributes to stronger and more reliable final retrieval performance.

4.5 Case Study

Figure 6 presents a representative case illustrating PaperScout's multi-turn retrieval behavior on a complex query involving CLIP, ViLD, and RegionCLIP. At the initial stage, the agent decomposes the query into multiple semantic facets and issues parallel search actions targeting different aspects of vision–language models. Based on the retrieved seed papers, the agent performs several rounds of expand actions, progressively exploring region-level and fine-grained vision–language pretraining methods with strong semantic coherence.

Notably, after expansion along existing directions becomes saturated, the agent initiates a new search call in a later turn, opening a previously unexplored research direction, fine-grained region understanding. This re-search step moves the agent beyond local expansion by introducing fresh candidate papers, which are further refined through subsequent expand actions. Such behavior demonstrates that PaperScout does not follow a fixed search–then–expand pipeline, but instead automatically alternates between search and expand based on accumulated context, constructing a structured and evolving paper exploration trajectory.

5 Conclusion

In this paper, we present PaperScout, an autonomous agent that formulates academic paper search as a sequential decision-making process. By explicitly deciding when and how to invoke search and citation expansion based on accumulated retrieval context, PaperScout moves beyond fixed retrieval workflows and enables flexible multi-turn paper exploration. We further introduce Proximal Sequence Policy Optimization (PSPO), a process-aware, sequence-level policy optimization method that aligns optimization with agent–environment interaction and enables stable training of multi-turn retrieval agents. Experiments on two benchmarks demonstrate that PaperScout trained with PSPO achieves superior retrieval performance compared to existing methods.

Limitations

Despite the effectiveness of our approach in multi-turn paper search, several limitations remain. First, our current evaluation primarily focuses on the computer science domain, and extending the study to broader research areas would further validate the generality of the proposed framework. Second, PaperScout relies on papers that are publicly accessible through online search engines and open repositories; retrieving papers from restricted or paywalled sources remains challenging and may limit coverage in certain domains. Third, our current implementation adopts a relatively limited search backend, and incorporating multi-source retrieval from heterogeneous scholarly databases could further improve robustness and recall. Finally, citation expansion currently considers only outgoing references, while leveraging incoming citations and richer citation graph signals may provide additional retrieval cues. We leave these directions for future work.

References

Ahmet Yasin Aytar, Kamer Kaya, and Kemal Kılıç. 2025. A synergistic multi-stage rag architecture for boosting context relevance in data science literature. *Natural Language Processing Journal*, page 100179.

Yuxiao Cheng and Jinli Suo. 2025. Openlens ai: Fully autonomous research agent for health infomatics. *arXiv preprint arXiv:2509.14778*.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2020. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067.

Michael Gusenbauer and Neal R Haddaway. 2020. Which academic search systems are suitable for systematic reviews or meta-analyses? evaluating retrieval qualities of google scholar, pubmed, and 26 other resources. *Research synthesis methods*, 11(2):181–217.

Michael Gusenbauer and Neal R Haddaway. 2021. What every researcher should know about searching–clarified concepts, search advice, and an agenda to improve finding in academia. *Research synthesis methods*, 12(2):136–147.

Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and 1 others. 2025. Pasa: An llm agent for comprehensive academic paper search. *arXiv preprint arXiv:2501.10120*.

Jyun-Yu Jiang, Mingyang Zhang, Cheng Li, Michael Bendersky, Nadav Golbandi, and Marc Najork. 2019. Semantic text matching for long-form documents. In *The world wide web conference*, pages 795–806.

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

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.

Paolo Manghi. 2024. Challenges in building scholarly knowledge graphs for research assessment in open science. *Quantitative Science Studies*, 5(4):991–1021.

Gary Marchionini. 2006. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46.

Sangwoo Park, Jinheon Baek, Soyeong Jeong, and Sung Ju Hwang. 2025. Chain of retrieval: Multi-aspect iterative search expansion and post-order search aggregation for full paper retrieval. *arXiv preprint arXiv:2507.10057*.

Shahzad Qaiser and Ramsha Ali. 2018. Text mining: use of tf-idf to examine the relevance of words to documents. *International journal of computer applications*, 181(1):25–29.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

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

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Xiaofeng Shi, Yuduo Li, Qian Kou, Longbin Yu, Jinxin Xie, and Hua Zhou. 2025. Spar: Scholar paper retrieval with llm-based agents for enhanced academic search. *arXiv preprint arXiv:2507.15245*.

Fiona Timmins and Catherine McCabe. 2005. How to conduct an effective literature search. *Nursing standard*, 20(11):41–47.

Rita Vine. 2006. Google scholar. *Journal of the Medical Library Association*, 94(1):97.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. 2025. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, and 1 others. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

A Experimental Details

A.1 Dataset Description

Our experiments are conducted on two benchmarks, AutoScholarQuery and RealScholarQuery (He et al., 2025), which represent synthetic and expert-curated scholarly search scenarios. AutoScholarQuery is a large-scale synthetic benchmark constructed from recent top-tier conference papers, including ICLR, ICML, NeurIPS, and ACL, while RealScholarQuery is a human-annotated benchmark designed to evaluate realistic and challenging scholarly search settings. Detailed statistics of the datasets are summarized in Table 4.

Dataset	#Train	#Dev	#Test
AutoScholarQuery	33551	1000	1000
RealScholarQuery	–	–	50

Table 4: Details of the datasets

A.2 Implementation of PaperScout

We conduct reinforcement learning for PaperScout on four NVIDIA A800 GPUs, using Qwen3-4B-Instruct-2507[5] as the backbone model. We set the actor and critic learning rates to 1×10^{-6} and 1×10^{-5}, respectively, with a per-gpu batch size of 4. For reward computation, we set the top-k to 3 to reduce reward variance across queries with varying difficulty. We further set the score threshold $\delta = 0.01$ and repeated penalty $\eta = 0.5$. To improve training stability, we first freeze the actor and pre-train the critic for 100 steps, followed by joint optimization of both components. The agent receives up to 10 unexpanded and 10 expanded papers at each turn. If the paper pool remains unchanged for three turns, the retrieval terminates.

Scorer. Given a user query together with the title and abstract of a paper, the scorer evaluates their semantic relevance and produces a continuous score that guides the retrieval decision of PaperScout. Specifically, the scorer prompts a language model to perform a relevance judgment, and defines the relevance score as the probability of generating the "True" token in the model output, yielding a value in $[0, 1]$. We implement the scorer using a qwen2.5-7B-based model [6] that is specifically trained for this relevance assessment task, and detailed prompt is shown on Appendix C.

B Details about LLM-score

Evaluation Protocol We employ an LLM-as-a-Judge to assess the relevance between a user query and a candidate paper based on its title and abstract, using three independent evaluators—DeepSeek-V3.2, Qwen3-max, and GPT-5.1. To ensure reliable and unbiased judgments, the evaluators are explicitly instructed to focus solely on semantic relevance, while ignoring factors such as writing quality, length, style, or popularity. Relevance is rated on a discrete four-level scale from 0 to 3, with clearly defined semantic criteria for each level, which helps constrain the models' decision boundaries and reduce ambiguity in scoring. The evaluators are further required to output results in a strict JSON format, ensuring consistent, structured, and easily parsable relevance

[5] https://www.modelscope.cn/models/Qwen/Qwen3-4B-Instruct-2507

[6] https://www.modelscope.cn/models/bytedance-research/pasa-7b-selector

scores for downstream analysis. The detailed evaluation prompt is provided in Appendix C

Consistency Analysis. We randomly sample 500 query–paper pairs from each method's test results, yielding a total of 3,000 samples, and analyze the consistency of the three LLM-based relevance scores for each sample. For each query–paper pair, we compute the standard deviation (STD) of the three scores as well as their maximum score gap (Max-Gap) to quantify inter-model agreement. As shown in Table 5, most samples exhibit strong consistency: 2,629 cases receive identical scores from all three models. The remaining discrepancies are predominantly minor, with only small score differences, while larger disagreements are rare and no samples exhibit extreme conflicts. These results indicate that the LLM-based scoring provides a stable and reliable signal for relevance evaluation.

STD	0.00	0.22	0.66	0.88	> 0.88
Max-Gap	0	1	2	2	3
Count	2629	336	25	10	0

Table 5: Consistency analysis of LLM-based relevance scores across three evaluators.

C Full Prompt set

Paper Search Agent Prompt

System

You are a research agent. Your goal is to find papers relevant to the User Query.

User Query

(...Detailed User Query...)

History Actions

(...Detailed History Actions...)

Paper List

(...Detailed Paper List...)

Instructions

- Analyze the Paper List and History Actions to determine the next set of actions. Enclose your analysis of the state and decision logic within `<analysis>...</analysis>` tags.
- You support parallel tool calling. You should output multiple tool calls in a single step if several independent actions are valuable at the current state.
- Attend to the history actions and avoid expanding the same papers.

Output Format

```
<analysis>
[Your analysis of the current state and decision logic...]
</analysis>
<tool_call>
[Tool call 1]
</tool_call>
<tool_call>
[Tool call 2]
</tool_call>
...
```

Tool Schema

```
SEARCH_TOOL_SCHEMA = {
    "type": "function",
    "function": {
        "name": "search",
        "description": (
            "Search for relevant papers in the arXiv repository."
        ),
        "parameters": {
            "type": "object",
            "properties": {
                "query": {
                    "type": "string",
                    "description": (
                        "A single search query (natural language or keywords). "
                        "No field scopes (ti:/abs:) or boolean ops. Must differ from all
                        history queries."
                    ),
                }
            },
            "required": ["query"],
        },
    },
}

EXPAND_TOOL_SCHEMA = {
    "type": "function",
    "function": {
        "name": "expand",
        "description": (
            "Expand from an existing paper by following its references to surface additional
            relevant works. "
            "Use this when search is saturated and you want to broaden coverage around a known
            paper."
        ),
        "parameters": {
            "type": "object",
            "properties": {
```

```
                "arxiv_id": {
                    "type": "string",
                    "description": (
                        "The arXiv identifier (e.g., '1706.03762') of a paper already in the
                        current paper list."
                    ),
                }
            },
            "required": ["arxiv_id"],
        },
    },
}

PAPERSEARCH_TOOL_SCHEMAS = [SEARCH_TOOL_SCHEMA, EXPAND_TOOL_SCHEMA]
```

Scorer Prompt

System

You are an elite researcher in the field of AI, conducting research on

(...Detailed User Query...)

Evaluate whether the following paper fully satisfies the detailed requirements of the user query and provide your reasoning. Ensure that your decision and reasoning are consistent.

Searched Paper

- Title: (...Detailed Title...)
- Abstract: (...Detailed Abstract...)
- User Query: (...Detailed User Query...)

Output format

- Decision: True / False
- Reason: ...
- Decision:

LLM Judge Prompt

System

You are an expert LLM-as-a-Judge for academic paper search. Your task is to evaluate how relevant a candidate paper is to a given research query. You must strictly follow the scoring rubric and output only the required JSON. You should ignore writing quality, length, style, or popularity. Only evaluate semantic relevance between the query and the paper content.

Query

(...Detailed User Query...)

Paper Candidate

- Title:
- Abstract:

Scoring Rubric (relevance_level)

- - 3 = Highly relevant — Directly addresses the core research question or problem expressed in the query.
- - 2 = Moderately relevant — About a closely related topic, but only partially matches the intent or focuses on a sub-aspect.
- - 1 = Weakly relevant — Belongs to the broader area but only tangentially connected to the query.
- - 0 = Irrelevant — Not relevant to the query in any meaningful way.

Output Format (must be valid JSON)

```
{
  "relevance_level": 0/1/2/3,
  "confidence": "low/medium/high",
  "reason_short": "A brief one-sentence justification"
}
```

You MUST output valid JSON and nothing else.

D An Example for PaperScout

Query

Could you provide some studies that have worked on leveraging the capability of CLIP and proposed models like ViLD and RegionCLIP?

Step 1

Paper List:

Empty list

Analysis:

The initial paper list is empty, and there are no history actions to avoid. The user is interested in studies that leverage CLIP's capabilities, specifically referencing models like ViLD and RegionCLIP. Since there are no existing papers in the pool, we must first search for relevant papers that discuss CLIP-based models or specifically mention ViLD and RegionCLIP. To ensure comprehensive coverage, we should perform a broad search query with keywords such as "CLIP", "ViLD", and "RegionCLIP" to identify foundational and related works. Given that the user is interested in the application and development of CLIP-based models like ViLD and RegionCLIP, initiating multiple search queries with slightly different but complementary keywords (e.g., "CLIP-based visual localization", "ViLD model", "RegionCLIP model") will help capture a diverse range of relevant papers. This ensures that we cover both general CLIP applications and the specific models mentioned in the query. We will perform three initial search queries: "CLIP visual localization" "ViLD model" "RegionCLIP model" These queries will yield a set of foundational and relevant papers directly matching the user's interests, enabling us to form a robust initial pool to proceed with further expansion or refinement.

Tool Call:

```
{"name": "search", "arguments": {"query": "CLIP visual localization"}}
{"name": "search", "arguments": {"query": "ViLD model"}}
{"name": "search", "arguments": {"query": "RegionCLIP model"}}
```

Step 2

Paper List:

Paper Pool Status:

[EXP]: Paper has been expanded (already used as a seed for more papers).

[NEW]: New paper found via search or expansion, candidate for further exploration.

Format: [arxiv_id] (score) [STATUS] Title

[2112.09106] (1.00) [NEW] RegionCLIP: Region-based Language-Image Pretraining

Abstract: Contrastive language-image pretraining (CLIP) performs poorly on region-level recognition due to a domain shift between whole-image and region-text alignment. We propose RegionCLIP to learn region-level visual representations aligned with textual concepts, significantly improving open-vocabulary object detection performance. ...

[2104.13921] (0.99) [NEW] Open-vocabulary Object Detection via Vision and Language Knowledge Distillation

Abstract: We propose ViLD, a vision-language knowledge distillation framework that transfers open-vocabulary knowledge from a pretrained image-text model to a two-stage detector. ViLD achieves state-of-the-art performance on LVIS and generalizes well to COCO, PASCAL VOC, and Objects365 without finetuning. ...

[2302.00275] (0.40) [NEW] Learning Generalized Zero-Shot Learners for Open-Domain Image Geolocalization

Abstract: We introduce StreetCLIP, a foundation model that achieves state-of-the-art zero-shot performance for open-domain image geolocalization. By meta-learning with synthetic captions, our method effectively transfers CLIP's generalized zero-shot capability to the geolocalization domain. ...

(...More Papers...)

[2305.05095] (0.12) [NEW] Less is More: Removing Text-regions Improves CLIP Training Efficiency and Robustness

Abstract: We show that removing images with text regions improves both the efficiency and robustness of CLIP training. The proposed filtering strategy boosts classification and retrieval accuracy and defends against typographic adversarial attacks. ...

Analysis:

The user query includes the topic about 'RegionCLIP model" and already performed a search for related topics, including 'CLIP visual localization", 'ViLD model", and 'RegionCLIP model". The current paper list includes several relevant works, with the most directly relevant being [2112.09106] (RegionCLIP: Region-based Language-Image Pretraining), which is the primary paper matching the exact query.

From the list: [2112.09106] is the core paper on RegionCLIP and directly addresses the user's initial interest. [2104.13921] (ViLD) is closely related, as it deals with open-vocabulary object detection via vision and language distillation—complementary to RegionCLIP's goal of fine-grained region understanding. [2308.02299] (RegionBLIP) is also relevant as it extends multimodal models to regional comprehension, implying a similar focus on region-level understanding.

[2310.18340] (UrbanCLIP) is less directly relevant, as it applies CLIP to urban region profiling rather than object detection or region alignment. [2302.00275] and [2309.16020] focus on geolocation, which is tangentially related via vision-language alignment but not directly on region-level features.

We should expand beyond the initial RegionCLIP paper to explore related works that enhance region-level understanding, especially those that build upon or extend CLIP for regional visual representations. Given that [2112.09106] is the primary paper, we should expand it to surface more foundational or downstream works that might be relevant.

Moreover, [2104.13921] and [2308.02299] are strong secondary candidates because they also involve region-image-text alignment in a multimodal setting. However, they were not previously expanded, so expanding these papers would provide a more robust understanding of the ecosystem.

Therefore, the next actions should be: Expand the primary paper [2112.09106] to uncover related foundational or downstream works. Expand [2104.13921] (ViLD) to explore its references, as it shares a similar vision (open-vocabulary detection via language vision alignment). Expand [2308.02299] (RegionBLIP) to discover related works on regional comprehension in multimodal models.

These expansions will enrich the paper pool without repeating prior searches or expanding redundant papers (e.g., earlier ones not in the current list).

Tool Call:

```
{"name": "expand", "arguments": {"arxiv_id": "2112.09106"}}
{"name": "expand", "arguments": {"arxiv_id": "2104.13921"}}
{"name": "expand", "arguments": {"arxiv_id": "2308.02299"}}
```

Step 3

...

Step 4

Paper List:

[2112.09106] (1.00) [EXP] RegionCLIP: Region-based Language-Image Pretraining

Abstract: CLIP is ineffective for region-level object detection due to the domain shift between whole-image–text alignment and region–text alignment. RegionCLIP extends CLIP to learn region-level visual representations, significantly improving open-vocabulary and zero-shot object detection performance. ...

[2104.13921] (0.99) [EXP] Open-vocabulary Object Detection via Vision and Language Knowledge Distillation

Abstract: We propose ViLD, a vision-language knowledge distillation framework that transfers open-vocabulary knowledge from a pretrained image-text model to a two-stage detector. ViLD achieves strong generalization on multiple detection benchmarks and outperforms prior methods on novel categories without additional finetuning. ...

[1908.02265] (0.54) [NEW] ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks

Abstract: We propose ViLBERT, a task-agnostic vision-and-language pretraining model with a two-stream architecture and co-attentional transformers. Pretrained on large-scale image-text data, ViLBERT achieves state-of-the-art performance across multiple vision-language tasks with minimal task-specific modification. ...

(...More Papers...)

[2301.12597] (0.05) [NEW] BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models

Abstract: We propose BLIP-2, an efficient vision-language pretraining framework that leverages frozen image encoders and frozen large language models. With a lightweight querying transformer, BLIP-2 achieves state-of-the-art performance and strong zero-shot capabilities with significantly fewer trainable parameters. ...

Analysis:

The user cares about the studies that leverage CLIP's capabilities, and we have already identified the primary paper on RegionCLIP (arXiv:2112.09106), which is now marked as [EXP] (expanded). This paper is the central focus and has been used as a seed for expansion via the expand tool (e.g., expanded papers like 2104.13921, 2308.02299, etc.).

Currently, the paper list includes several relevant papers on vision-language pre-training and region-based models, such as: RegionBLIP (2308.02299): focuses on regional comprehension in multi-modal models. UrbanCLIP (2310.18340): uses text-enhanced urban region profiling with CLIP-like training. Oscar (2004.06165): object-semantic alignment pre-training. FILIP (2111.07783): fine-grained language-image pre-training with token-wise alignment. STAIR (2301.13081): sparse text and image representation.

These papers are closely related in theme, focusing on regional or fine-grained visual understanding in vision-language settings, which supports the broader context of RegionCLIP as a foundation. However, many of these have been already expanded (marked [EXP]), so further expansion may not yield new relevant papers.

To deepen the exploration, we should: Identify missing papers from the primary RegionCLIP literature by searching on related keywords to complement what we already found. Expand on other key papers that are relevant but not yet expanded, such as: STAIR (2301.13081): sparse representations, strong performance in retrieval. FILIP (2111.07783): fine-grained alignment, strong localization. BLIP-2 (2301.12597): foundational MLLM, influential in regional vision-language tasks.

Avoid expanding on already expanded papers (e.g., 2112.09106, 2104.13921, 2308.02299, 2310.18340, 2305.05095) to avoid redundancy.

Therefore, the next actionable steps should include: Searching for papers related to 'fine-grained region understanding in vision-language models" to complement existing ones. Expanding on key under-explored candidates like FILIP (2111.07783) and BLIP-2 (2301.12597), as they offer strong theoretical or practical advancements in regional language-image alignment.

This will ensure we cover both complementary and emerging directions in the region-based vision-language pre-training space.

Tool Call:

```
{"name": "search", "arguments": {"query": "fine-grained region understanding in
vision-language models"}}
{"name": "expand", "arguments": {"arxiv_id": "2111.07783"}}
{"name": "expand", "arguments": {"arxiv_id": "2301.12597"}}
```

More Steps

...

Final Result

Paper ID: 2112.09106

- **Title:** RegionCLIP: Region-based Language-Image Pretraining
- **Abstract:** ...
- **Score:** 0.9957

Paper ID: 2104.13921

- **Title:** Open-vocabulary Object Detection via Vision and Language Knowledge Distillatio
- **Abstract:** ...
- **Score:** 0.9949

(...More Papers...)

Paper ID: 2203.16513

- **Title:** PromptDet: Towards Open-vocabulary Detection using Uncurated Images
- **Abstract:** ...
- **Score:** 0.5027