PROSEARCH: TEACHING LLMS WHEN (NOT) TO SEARCH OVER STREAMING INPUTS VIA REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce ProSearch, a reinforcement-learning framework for training proactive search agents for streaming inputs. Given a continuous input stream, the agent must decide *whether* to act, *what* external knowledge to retrieve, and *how* to respond, under *streaming latency constraints*. Existing systems either retrieve on every input or delegate the trigger to an external classifier, leaving the "whether to act" decision outside the learned policy. ProSearch instead internalises it, training a single policy end-to-end with GRPO so that the trigger, query, and grounded answer are emitted by the same model and optimised against the same reward. The central challenge is reward sparsity: actionable segments are rare and correct behavior requires a three-step pipeline (trigger, retrieve, respond), so an outcome-only reward yields degenerate gradients. We address this with a structured reward decomposition whose three terms each score a different sub-event of the rollout, densifying the learning signal for GRPO relative to an outcome-only reward and yielding stable training. Across three proactive tasks spanning claim verification and knowledge-seeking turn detection, ProSearch consistently beats reactive and externally-orchestrated baselines, with up to a $2\times$ gain in end-to-end task success.

1 INTRODUCTION

A growing class of real-world applications requires AI agents that do more than respond to explicit queries: they must continuously monitor an information stream and *autonomously* decide when intervention is warranted (Liu et al., 2025; Lu et al., 2025; Zhang et al., 2024a;b; Zhao et al., 2025). A live-stream fact-checking assistant must distinguish verifiable claims from casual commentary (Setty, 2025; V & Setty, 2024); a customer-service copilot must recognize when a user's question requires an external knowledge lookup rather than a response from dialogue context alone (Kim et al., 2020; 2023). In each case the agent faces a *tripartite decision*: (1) *whether* the current segment warrants action, (2) *what* external knowledge to retrieve, and (3) *how* to integrate that knowledge into a grounded response under tight streaming latency constraints. This *proactive decision-making* capability is fundamentally different from reactive systems that treat every input as requiring engagement, yet it remains largely unexplored in LLM-based agents.

Existing approaches to knowledge-augmented LLMs fall short of this proactive requirement. Reactive RAG systems (Lewis et al., 2020; Gao et al., 2023; Asai et al., 2024) retrieve on every input or use self-reflection to decide per-query, but both assume the user has already signaled a need for a response, making the trigger decision trivial. Externally-orchestrated systems such as LLM-Augmenter (Peng et al., 2023) delegate the retrieval decision to separate policy modules, but the decision logic is not internalised within the model and cannot be jointly optimised with downstream reasoning. Search-augmented reasoning models (Jin et al., 2025) train LLMs to interleave search with chain-of-thought via RL, but assume every input requires engagement; the "whether to act" decision is not part of the learned policy. None of these paradigms address the defining challenge of proactive settings: learning a sharp decision boundary where the default state is *inaction* and the active state is the rare, high-stakes exception.

We introduce ProSearch, which bridges this gap by framing proactive decision-making as a per-segment decision process over $\{\text{WAIT}, \text{SEARCH}\}$ and training a single LLM policy end-to-end with

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

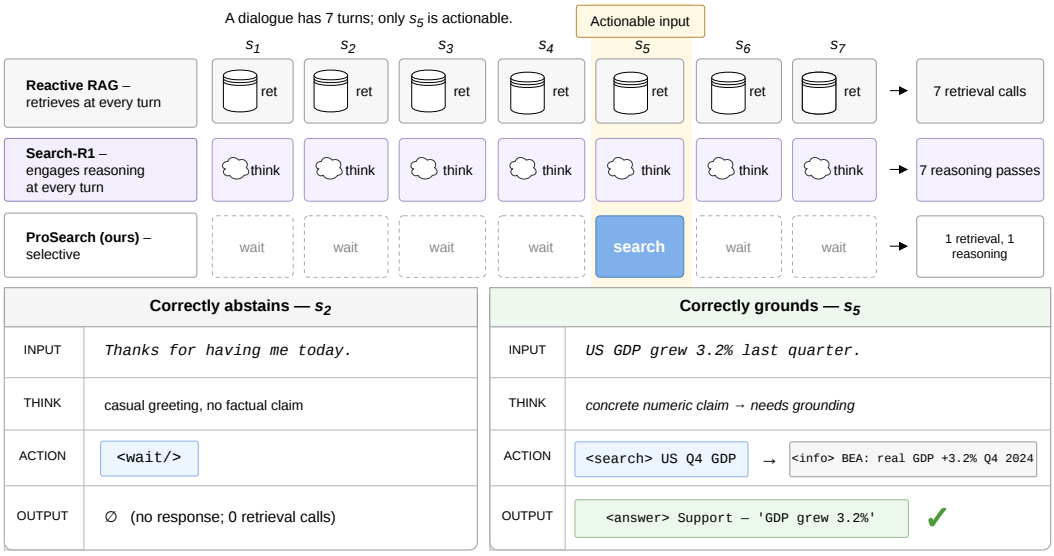

Figure 1: Comparison of reactive and proactive systems. Reactive systems retrieve or reason on every input segment; ProSearch learns to selectively engage only on actionable segments. In the example, the agent monitors a podcast transcript and must distinguish verifiable claims from casual commentaries. Upon detecting a claim, it issues a search query, retrieves evidence, and produces a grounded response.

Group Relative Policy Optimization (GRPO; Shao et al. 2024; Guo et al. 2025). Unlike modular pipelines, ProSearch internalises all three aspects of the proactive decision (trigger detection, query formulation, and grounded response) within a unified policy, enabling joint optimisation across all the stages.

Applying RL to the proactive setting, however, exposes a fundamental challenge: *reward sparsity under action rarity*. Actionable segments are only a small fraction of realistic streams, and correct behavior requires a three-step pipeline (trigger → retrieve → respond). A naive outcome-only reward yields no learning signal until the model both hits an actionable segment and executes the full pipeline correctly by chance. In practice this appears as a long cold-start with near-zero gradient on the search and answer actions and unstable GRPO training. We address it via *structured reward decomposition*: additive shaping of trigger calibration, retrieval quality, and grounded answering that densifies the learning signal relative to an outcome-only reward and stabilizes GRPO.

Figure 1 compares ProSearch to reactive systems and shows a concrete example of the proactive decision process. Our contributions are as follows:

- **The ProSearch framework**: We formulate proactive decision-making as a per-segment decision process where a single LLM policy chooses WAIT or SEARCH, unifying trigger detection, knowledge retrieval, and grounded response generation. Unlike reactive search-augmented models that engage on every input, ProSearch internalises the *whether to act* decision within the policy itself.
- **Structured reward decomposition for sparse-action RL**: We design a reward function that decomposes proactive behavior into process-level sub-objectives (trigger calibration, retrieval quality, and answer grounding) rather than relying on sparse end-to-end correctness alone. This densifies the group-relative ranking signal relative to an outcome-only reward; combined with the SFT cold-start and class-balanced sampler, it supports stable GRPO training even when actionable events are a small minority of the input stream.
- **Multi-domain evaluation and a new dataset**: We instantiate ProSearch on three diverse proactive tasks: real-time claim verification (for which we construct a new large-scale silver-labeled dataset with harvested web evidence), and knowledge-seeking turn detection in two dialogue settings (DSTC9-T1 (Kim et al., 2020) and DSTC11-T5 (Kim et al., 2023)). The same architecture and training methodology yields consistent improvements over reactive and externally-orchestrated baselines across all domains.

2 Related Work

Retrieval-augmented LLMs. The decision of *when* to retrieve is what distinguishes proactive agents from the broader retrieval-augmented generation literature, where retrieval is issued unconditionally on every input or at fixed reasoning steps (Lewis et al., 2020; Guu et al., 2020; Trivedi et al., 2023), gated by model-internal heuristics (Jiang et al., 2023; Su et al., 2024), routed per query by complexity (Jeong et al., 2024; Shi et al., 2023), or learned end-to-end with outcome rewards on the actionable subset (Asai et al., 2024; Jin et al., 2025; Guo et al., 2026; Hashemi et al., 2026). Externally-orchestrated systems instead delegate the decision to a separate module (Peng et al., 2023), and streaming work performs incremental reasoning or speech-time tool calls without a learned trigger policy (Tong et al., 2025; Arora et al., 2025). Even RL search agents such as Search-R1 (Jin et al., 2025) still train where retrieval is always warranted, so the action space never includes a dominant no-op. Self-RAG (Asai et al., 2024) can skip retrieval via reflection tokens, but still assumes a user-issued query whose answer is required; adaptive and gated RAG likewise take an already-actionable query and ask *how much* retrieval to issue, or *whether to bother retrieving*. ProSearch answers a prior question: given a stream in which most inputs warrant no response at all, at which timesteps should the system act? We learn that trigger jointly with query formulation and response generation via RL.

Proactive and autonomous agents. Proactive behavior, where agents initiate actions without explicit user commands, has roots in classical AI architectures (Wooldridge & Jennings, 1995). In the LLM era, tool-augmented agents (Schick et al., 2024; Nakano et al., 2022; Yao et al., 2023) can invoke external tools but remain *reactive*, engaging only in response to user queries. Recent work explores proactivity along two axes. *Cooperation and planning* infers teammate intentions (Zhang et al., 2024a) or asks clarifying questions before execution (Zhang et al., 2024b). *User-need anticipation* offers unsolicited assistance via reward-trained policies (Lu et al., 2025), prompted proactivity scores (Yang et al., 2025), or IDE-level triggers (Zhao et al., 2025). These axes define proactivity as anticipating *what the user wants*; ProSearch instead casts it as deciding *when external knowledge is needed* in a streaming pipeline, training the trigger jointly with query formulation and grounded response generation. Adjacent settings, knowledge-seeking turn detection (Kim et al., 2020; 2023) and multi-stage fact-checking (Thorne et al., 2018b; Guo et al., 2022), also face a when-to-engage decision, but typically train separate classifiers per subtask rather than a single policy.

3 The ProSearch Framework

The agent monitors a continuous input stream that arrives one segment at a time (a dialogue utterance, a chunk of streaming text, or a speech span from automatic speech recognition (ASR)). Each segment is an input $x \in \mathcal{X}$ paired with a binary trigger label $g_{\text{trig}}(x) \in \{0, 1\}$ that is 1 iff x warrants a response grounded in a shared retrieval corpus $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$, and a gold reference $y^\star \in \mathcal{Y}$ defining the correct response when $g_{\text{trig}}{=}1$. A fixed retriever $\mathcal{R}$ returns the top-K passages for any natural-language query and is shared across segments. On each x the agent either abstains and emits no user-facing reply, or consumes retrieved evidence and emits a grounded answer $\hat{y} \in \mathcal{Y}$ matching $y^\star$. Decisions are scored *per segment* with a supplied context window: the action on x_t does not change the label or corpus for x_{t+1}, and our main experiments use pre-segmented text with a frozen retrieval index. Because $g_{\text{trig}}{=}0$ on the majority of segments in realistic streams, abstention is the dominant correct behavior. This formulation thus differs from standard single-query RAG, in which every input warrants a response by construction.

We formalize the agent as a single language-model policy whose top-level action WAIT or SEARCH is committed by the first response token, and whose ensuing trajectory (including a `<wait>` marker, or an interleaved sequence of reasoning, retrieval, and a grounded answer) is scored end-to-end by one scalar reward. Figure 2 illustrates the resulting agent interface.

3.1 Agent and Trajectory

We instantiate the agent as a single autoregressive language model π_θ that, given an input x, emits one rollout τ drawn from the action set $\mathcal{A} = \{\text{WAIT}, \text{SEARCH}\}$. The agent's *decision* is committed by the first content token of the response; the rest of the rollout instantiates that decision.

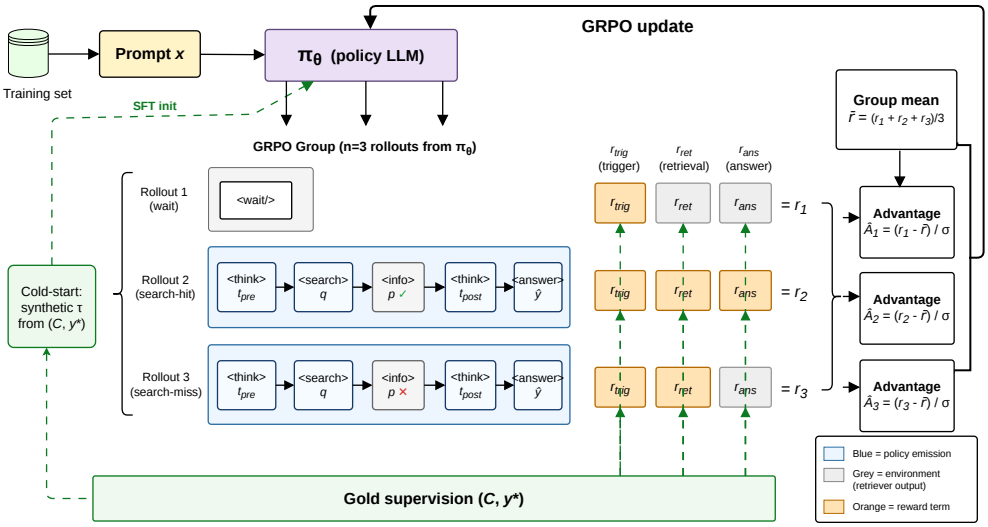

Figure 2: **Framework overview.** A group of e.g. n=3 rollouts (one `<wait>`, two `<search>` that differ in retrieval hit/miss) is scored by the three-term reward (Eq. 2); group-relative advantages (Eq. 6) update π_θ via the GRPO surrogate. Blue: policy emissions; grey: environment-injected. The same gold supervision $(\mathcal{C}, y^\star)$ (bottom rail) supplies every reward term and the SFT cold-start.

Wait branch. A WAIT rollout is the single terminal token: $\tau =$ `<wait>`. Collapsing it to one token keeps WAIT strictly cheaper than SEARCH, which is what lets the latency discount in r_{ret} (§3.2) price the WAIT-vs-SEARCH trade-off.

Search branch. A SEARCH rollout is an interleaved sequence of reasoning, retrieval, and a grounded `<answer>` emission (not a separate top-level action):

$\tau =$ `<think>` t_{s} `</think> <search>` q `</search><information>` p `</information>`
`<think>` t_{e} `</think> <answer>` $\hat{y}$ `</answer>`.

The reasoning spans $(t_{\text{s}}, t_{\text{e}})$, the query $q \in \mathcal{Q}$, and the answer $\hat{y}$ are policy emissions; the passage $p = \mathcal{R}(q) \in \mathcal{C}^K$ is environment-injected and is *not* an emission of π_θ, a distinction we exploit when constructing the loss (§3.3).

Note that WAIT/SEARCH name the top-level decision; trigger/retrieve/respond name the scored sub-events of a SEARCH rollout (Eq. 2). They are two levels of one policy, not two policies.

Learning objective. A single scalar reward $r(\tau) \in \mathbb{R}$ is attached to the terminal token of each rollout, and π_θ is trained to maximize

$$J(\theta) \;=\; \mathbb{E}_{x\sim p_{\mathcal{D}}}\, \mathbb{E}_{\tau\sim\pi_\theta(\cdot|x)}\big[\, r(\tau)\,\big], \tag{1}$$

where $p_{\mathcal{D}}$ is the empirical distribution over training inputs. §3.2 constructs r; §3.3 converts the terminal scalar into per-token credit for the GRPO update.

3.2 Reward Design

The reward $r(\tau)$ is the sum of three additive shaping terms, each targeting one lesson the policy must learn, i.e., when to trigger, what to retrieve, and what to answer:

$$r(\tau) \;=\; \underbrace{r_{\text{trig}}(\tau)}_{\text{trigger}} + \underbrace{r_{\text{ret}}(\tau)}_{\text{retrieval}} + \underbrace{r_{\text{ans}}(\tau)}_{\text{answer}}. \tag{2}$$

Each term scores a different sub-event of the rollout: r_{trig} the decision, r_{ret} the query, and r_{ans} the answer. Within a GRPO group the *total* reward ranking is often driven by different terms on different prompts (§3.3). This is additive shaping, not token-scoped or independent credit assignment.

Trigger shaping (r_{trig}). The trigger term scores the agent's WAIT/SEARCH choice against the gold label $g_{\text{trig}}(x)$:

$$r_{\text{trig}}(\tau) \;=\; \begin{cases} +\tau_{\text{TN}} & g_{\text{trig}}=0,\ \text{WAIT}, \\ -\tau_{\text{FP}} & g_{\text{trig}}=0,\ \text{SEARCH}, \\ -\tau_{\text{FN}} & g_{\text{trig}}=1,\ \text{WAIT}, \\ 0 & g_{\text{trig}}=1,\ \text{SEARCH}. \end{cases} \tag{3}$$

The constants $\tau_{\text{TN}}, \tau_{\text{FP}}, \tau_{\text{FN}} \geq 0$ control the precision/recall trade-off; the correct-search baseline at 0 delegates positive credit on a correct search to r_{ret} and r_{ans}, so r_{trig} scores the *decision* alone. We set $\tau_{\text{FN}}{=}0$: the foregone $r_{\text{ret}} + r_{\text{ans}}$ on a missed trigger already supplies the recall signal.

Retrieval shaping (r_{ret}). On a SEARCH rollout for a should-search input, the retrieval term rewards passage quality and discounts turn count:

$$r_{\text{ret}}(\tau) \;=\; \mathbb{1}[\text{SEARCH},\ g_{\text{trig}}=1] \cdot \rho(h(\tau)) \cdot b(\tau), \tag{4}$$

where $h(\tau) \in \{0,1\}$ indicates whether the top-K retrieved passages cover the gold evidence cited in $y^\star$, $\rho(h){=}\rho_{\text{hit}}$ if $h{=}1$ else ρ_{miss}, and $b(\tau) = 1/(1+\beta\,(T(\tau){-}1))$ is the latency discount on the number of agent turns $T(\tau)$. Note that under the streaming latency constraints of our target applications we default to a single retrieval call per triggered segment ($T(\tau)=1$), so the latency discount $b(\tau)$ is inactive unless we set $T(\tau) > 1$ to enable multi-hop search. The strict ordering $\rho_{\text{hit}} > \rho_{\text{miss}} > 0$ teaches the policy to write *good* queries while keeping any search strictly better than the silence basin on trigger inputs; scoping $b(\tau)$ to r_{ret} alone ensures the turn penalty hits retrieval itself, not waiting or answering.

Answer shaping (r_{ans}). The answer term fires only on rollouts that successfully retrieved the gold evidence ($h(\tau){=}1$), and rewards two complementary notions of answer quality:

$$r_{\text{ans}}(\tau) \;=\; \mathbb{1}[h(\tau){=}1] \cdot \big(w_v\, v(\tau) \;+\; w_j\, s(\tau)/5\big). \tag{5}$$

Here $v(\tau) \in \{0,1\}$ is an exact-match indicator against $y^\star$ when $y^\star$ is a discrete label (e.g. a verdict in claim verification); for free-form references it is not well-defined and we set $w_v{=}0$, relying on the LM-judge term alone. $s(\tau) \in \{0,\ldots,5\}$ is the integer score returned by an LLM-as-judge that compares the model's `<answer>` body to $y^\star$ under a 0–5 rubric. The gate $\mathbb{1}[h(\tau){=}1]$ blocks the degenerate strategy of answering from parametric memory: without it, a confident-but-ungrounded answer matching $y^\star$ would earn the same credit as a grounded one.

3.3 Optimization with GRPO

We optimise π_θ with Group Relative Policy Optimization (Shao et al., 2024): for each prompt x we sample n rollouts $\{\tau_1,\ldots,\tau_n\} \sim \pi_\theta(\cdot \mid x)$, attach the terminal reward $r(\tau_i)$ from Eq. 2 to each, and compute group-relative advantages

$$\hat{A}_i \;=\; \frac{r(\tau_i) - \text{mean}_j\, r(\tau_j)}{\text{std}_j\, r(\tau_j) + \varepsilon}. \tag{6}$$

The advantage is broadcast onto the policy tokens of τ_i and π_θ is updated with a clipped PPO-style surrogate plus a low-variance KL penalty against a frozen reference.

Structured shaping within one GRPO update. GRPO attaches a single scalar $r(\tau_i)$ (Eq. 2) to each rollout and broadcasts the resulting advantage $\hat{A}_i$ (Eq. 6) to all policy-generated tokens of that rollout, so an answer bonus can also reinforce earlier query or trigger tokens. What the decomposition provides is denser *ranking signal* inside the group, not independently scoped gradients. Concretely: on a should-WAIT input the group often mixes WAIT and SEARCH, and r_{trig} tends to dominate the total-reward gap; on a should-SEARCH input, WAIT rollouts forgo $r_{\text{ret}}{+}r_{\text{ans}}$ (we set $\tau_{\text{FN}}{=}0$) while SEARCH rollouts separate on $\{\rho_{\text{miss}}, \rho_{\text{hit}}\}$, so retrieval terms shape the ranking; among hit rollouts, r_{ans} further separates answer quality.

State-masking the retrieved passage. The `<information>` block p is environment-injected, not policy-emitted. We mask its tokens from both the policy-gradient and the KL term so that the loss flows only through the model's own emissions ($t_{\text{s}}, q, t_{\text{e}}, \hat{y}$, or the lone `<wait>` token). Without this mask, the policy is implicitly penalized for the perplexity of a string it did not produce, biasing it against issuing any `<search>` at all.

Task	Input	Knowledge	#Samples			Trig.%		
			Train	Val	Test	Train	Val	Test
ClaimVerif	Podcast utt.	Web articles (21k)	18,493	2,924	3,715	24.0	29.7	21.2
DSTC9-T1	Dial. turns	Factual FAQs (12k)	71,348	9,663	4,181	26.9	27.7	47.4
DSTC11-T5	Dial. turns	Subj. reviews (11k)	28,431	4,173	5,475	51.9	51.0	51.1

Table 1: Dataset statistics. Corpus sizes (in parentheses) are the number of indexed passages.

3.4 Bootstrapping with a Synthetic Cold-Start

Direct GRPO from an instruction-tuned base sometimes collapses to an always-<wait> policy from which on-policy exploration cannot recover. The mechanism is intuitive: at initialisation, the expected payoff of issuing a <search> and producing a grounded answer is dominated by retrieval and generation noise, while the correct-wait reward $+r_{\text{TN}}$ delivers a small but reliable positive baseline on the non-trigger majority; once GRPO concentrates probability on the trivial action, on-policy exploration cannot escape. We therefore prepend a short SFT cold-start built directly from the task labels $(x, y^\star)$ by copying them into trajectories, with no distillation teacher: a non-trigger becomes a bare <wait>; a trigger becomes $\langle q, p, \hat{y}\rangle$ with $q{=}x$ (no rephrasing), p the corpus document cited by $y^\star$'s evidence, and $\hat{y}{=}y^\star$. The two <think> spans are drawn from a small templated bank, so every cold-start token is label-copied or template-drawn. After this SFT pass, the GRPO recipe of §3.2–§3.3 runs unchanged.

4 ClaimVerif Dataset

Public benchmarks satisfy the proactive setting for knowledge-seeking dialogue (DSTC9-T1, DSTC11-T5; §5) but not for stream-level claim verification: existing corpora (FEVER (Thorne et al., 2018a), SciFact (Wadden et al., 2020)) supply pre-segmented claims, discarding the trigger decision by construction. We therefore construct *ClaimVerif*, a podcast-derived dataset that preserves the raw utterance stream, labels each utterance for check-worthiness, and grounds every check-worthy claim in web evidence with a stance verdict.

We source 40 full-length episodes from 20 topically diverse podcasts (Clifton et al., 2020) (politics, science, etc.), segmented into utterance-level units, and apply a three-stage LLM pipeline: (1) **Claim detection**: each utterance is classified as check-worthy using 10 balanced few-shot exemplars drawn from expert annotations, following a four-category taxonomy (factual, causal, numerical, attributed). (2) **Evidence harvesting**: for each detected claim, a search query is generated and executed against the DuckDuckGo API (up to 8 results); web pages are fetched and converted to clean text via trafilatura, from which the LLM extracts concise evidence snippets (up to 4 per claim). (3) **Stance classification**: each (claim, evidence) pair is classified as Supports or Refutes; ambiguous pairs are discarded. This yields 25,132 RL-ready samples, which we partition *episode-disjointly* (28/6/6 of the 40 episodes) into the 18,493 / 2,924 / 3,715 train/val/test rows as shown in Table 1.

Label quality validation: We ran two independent audits of the label creation pipeline. Firstly, we ran the pipeline on the same podcasts as Setty (2025), which include 1,960 human-annotated utterances, too small for robust end-to-end evaluation but useful as a probe of our label creation pipeline. We compare the silver labels created by the pipeline with the human-annotated gold labels, and the trigger recall/precision against gold is $1.00/0.81$, and the stance recall/precision is 0.76–0.86 on both Supports and Refutes. Secondly, we manually audited 200 utterance samples from *ClaimVerif test* episodes with mixed trigger and non-trigger utterances ($n{=}39$ stance pairs among triggered rows, judged against the linked evidence span). The trigger recall/precision against gold is $0.93/0.78$, and the stance recall/precision is 0.75–0.87 on both Supports and Refutes. These audits support the reliability of the label-creation pipeline. See App. D for more details.

5 Experiments

5.1 Experimental Setup

Tasks and datasets. We evaluate ProSearch on three proactive substrates (Table 1): silver-labeled podcast claim verification over a frozen web-evidence corpus (*ClaimVerif*, §4), and knowledge-

	ClaimVerif					DSTC9-T1					DSTC11-T5				
	Act	**Trig.**	**Retr.**	**Resp.**	**E2E**	**Act**	**Trig.**	**Retr.**	**Resp.**	**E2E**	**Act**	**Trig.**	**Retr.**	**Resp.**	**E2E**
Llama-3.2-3B-Instruct															
Base	4.05	0.35	0.53	0.49	0.08	3.53	0.64	0.75	0.70	0.33	3.75	0.67	0.27	0.54	0.10
Base (*w/* gate)	3.89	0.68	0.50	0.50	0.17	2.92	0.97	0.74	0.70	0.50	3.59	0.99	0.27	0.54	0.15
Search-R1	1.40	0.35	0.43	0.84	0.13	3.47	0.65	0.76	0.78	0.39	4.86	0.69	0.37	0.54	0.14
Search-R1 (*w/* gate)	1.37	0.70	0.47	0.82	0.27	2.57	0.97	0.75	0.76	0.56	4.85	0.99	0.37	0.54	0.20
ProSearch (SFT)	1.58	0.51	0.66	0.76	0.26	**1.14**	0.95	0.90	0.68	0.59	1.31	**1.00**	0.49	0.50	0.25
ProSearch (GRPO)	1.66	0.57	0.62	0.64	0.23	1.24	0.93	0.87	0.85	0.69	2.69	0.99	0.51	0.59	0.30
ProSearch (SFT+GRPO)	**1.20**	**0.64**	**0.77**	**0.83**	**0.40**	1.21	**0.96**	**0.93**	**0.89**	**0.79**	**1.27**	**1.00**	**0.57**	**0.75**	**0.43**
Llama-3.1-8B-Instruct															
Base	6.62	0.58	0.46	0.58	0.09	6.03	0.64	0.80	0.78	0.40	6.34	0.68	0.32	0.58	0.13
Base (*w/* gate)	5.94	0.70	0.52	0.57	0.21	4.75	0.97	0.80	0.79	0.61	6.07	1.00	0.32	0.58	0.19
Search-R1	1.39	0.36	0.50	0.84	0.15	4.90	0.64	0.82	0.83	0.43	5.85	0.68	0.37	0.55	0.14
Search-R1 (*w/* gate)	1.51	0.70	0.53	0.83	0.31	3.64	0.97	0.81	0.83	0.66	5.90	1.00	0.37	0.55	0.20
ProSearch (SFT)	1.75	0.65	0.72	0.81	0.38	**1.44**	0.96	0.92	0.79	0.69	**1.56**	**1.00**	0.49	0.50	0.24
ProSearch (GRPO)	3.82	0.59	0.66	0.67	0.26	2.99	0.95	0.90	0.83	0.71	3.88	0.98	0.57	0.58	0.32
ProSearch (SFT+GRPO)	**1.35**	**0.66**	**0.79**	**0.87**	**0.44**	1.51	**0.96**	**0.92**	**0.94**	**0.84**	1.64	**1.00**	**0.58**	**0.73**	**0.45**

Table 2: Main results across three substrates and two backbones. **Act** = *act latency* in seconds; **Trig.** = trigger F1; **Retr.** = retrieval hit; **Resp.** = response score on true positives (verdict accuracy on ClaimVerif, normalised LLM-judge score on the dialogue substrates); **E2E** = end-to-end F1 (our primary metric). Best result per (backbone, substrate) in bold (for Act, lower is better). Full columns in App. B.

seeking dialogue with human-annotated triggers on *DSTC9-T1* (Kim et al., 2020) and *DSTC11-T5* (Kim et al., 2023). They span objective vs. subjective grounding, monologue vs. dialogue, and trigger priors from $\sim$24% to $\sim$52%; DSTC conversion details are in App. C. Test rows are scored independently with a supplied context window (no carried stream state; §3).

Model and training. We validate ProSearch on two backbones of different sizes, *Llama-3.2-3B-Instruct* and *Llama-3.1-8B-Instruct*, trained with an identical recipe so that the effect of scale can be read off cleanly. To check cross-family transfer we also rerun the 8B recipe on *Ministral-8B-Instruct-2410* (App. G). The retriever $\mathcal{R}$ is a dense BGE encoder (Xiao et al., 2024) and is fixed throughout; unless stated otherwise we retrieve top-$K=3$ on *ClaimVerif* and top-$K=5$ on the DSTC substrates, with the train- and eval-time budgets matched. The eight reward hyperparameters of Eq. 2 take the values $(\tau_{\text{TN}}, \tau_{\text{FP}}, \tau_{\text{FN}}) = (0.3, 0.1, 0.0)$ for trigger shaping, $(\rho_{\text{hit}}, \rho_{\text{miss}}) = (1.0, 0.5)$ for retrieval shaping, $\beta = 0.3$ for the latency discount, $w_v = 0.5$ on *ClaimVerif* (w_v=0 on dialogue), and judge weight $w_j = 0.5$ on *ClaimVerif* / 1.0 on dialogue (App. A). GRPO uses group size $n=3$ for 400 optimisation steps. The synthetic SFT cold-start of §3.4 runs for 2 epochs. Implementation uses `verl` (Sheng et al., 2024) with FSDP and full offloading on a single 8×H100 node. Full per-substrate learning rates and corpus statistics are deferred to App. A.

Evaluation metrics. We report five metric families, computed uniformly across substrates. **Trigger:** precision, recall, and F1 of the binary act/no-act decision against g_{trig}. **Retrieval:** hit rate on true-positive trigger turns (fraction whose top-K window contains a gold-anchored passage), isolating retrieval quality from trigger calibration. **Response:** task-specific outcome quality on true-positive rows: SUPPORTS/REFUTES verdict accuracy on ClaimVerif and the LLM-judge factuality score on DSTC9-T1/DSTC11-T5. **End-to-end (E2E-F1):** our primary metric over the full test set. Per-row credit combines correct trigger, retrieval hit when gold evidence exists (retrieval_ok; App. E), and response credit (verdict-match on *ClaimVerif*; normalised LLM-judge s_j on dialogue); E2E-F1 is the harmonic mean of corpus-level precision/recall over those credits. Gold-anchored hit can zero other valid passages (e.g. Case 12). Judge robustness / rubrics: App. F, H. **Latency:** Act / Wait generation+retrieval seconds on triggered vs. non-triggered rows (no prefill/TTFT; App. E). We report Act in the main table; Wait is deferred to App. B.

Baselines. We compare ProSearch against four baselines per backbone, varying the *generator* (**Base** vs. **Search-R1**) and whether an external classifier handles triggering (**w/ trigger gate**). **Base**

	ClaimVerif					DSTC9-T1				
	Act	**Trig.**	**Retr.**	**Resp.**	**E2E**	**Act**	**Trig.**	**Retr.**	**Resp.**	**E2E**
w/o judge reward	1.49	0.65	0.66	0.83	0.35	1.52	**0.97**	0.91	0.78	0.69
w/o retrieval reward	1.51	**0.67**	0.70	0.84	0.39	1.58	0.96	0.92	0.94	0.83
w/o balanced sampling	1.49	**0.67**	0.73	0.84	0.39	**1.41**	0.96	0.91	0.93	0.81
ProSearch (full)	**1.35**	0.66	**0.79**	**0.87**	**0.44**	1.51	0.96	**0.92**	**0.94**	**0.84**

Table 3: Ablation results on ClaimVerif and DSTC9-T1. Columns follow Table 2.

is the off-the-shelf instruction-tuned backbone, evaluated zero-shot. **Search-R1** (Jin et al., 2025) finetunes the same backbone with the published think–search–answer recipe on actionable inputs (g_{trig}=1); lacking a <wait> token, it engages on most inputs and tends toward near-saturating recall with precision near the gold-trigger rate (full P/R in App. B). The *gated* variants (**Base / Search-R1 w/ trigger gate**) attach a same-size Llama-3.x classifier, finetuned on g_{trig}, that emits <wait> and skips retrieval on negatives: the strongest post-hoc cascade we evaluate.

5.2 Main Results

Table 2 reports results for both backbones across all three substrates. Three findings stand out.

(R1) Joint training outperforms two-stage cascades. ProSearch leads E2E-F1 at both scales, beating the strongest cascade (Search-R1 w/ trigger gate) by $+0.13$–$+0.25$ at 8B. On DSTC the gate already saturates Trig. F1 (0.97–1.00), so the residual lift is retrieval/response from joint query–answer training; on ClaimVerif the cascade can edge Trig. (0.70 vs. 0.66) but still trails E2E; across substrates retrieval rises $+0.11$–$+0.26$ without gated feedback into query writing.

(R2) SFT and GRPO contribute complementary signal; the joint recipe captures both. Decoupling the stages attributes the gain. SFT alone anchors a compact think–search–answer template and a calibrated trigger (compact Act latency, 1.4–1.8 s at 8B on the dialogue substrates; on ClaimVerif, SFT+GRPO is slightly faster). GRPO alone over-triggers (high recall, weak wait calibration) and inflates Act latency (3.0–3.9 s at 8B), signatures of a policy that rarely observes <wait> with non-zero gradient. SFT+GRPO wins on every substrate at both scales ($+0.06$–$+0.14$ E2E over the better single stage): the cold-start supplies a usable action distribution; GRPO then sharpens retrieval and response without disturbing trigger calibration. On *DSTC11-T5*, SFT's perfect Trig. F1 still yields only 0.24–0.25 E2E because retrieval and response are binding; GRPO lifts both, with the larger gain on response.

(R3) Latency is preserved. ProSearch is fastest on most cells (Act 1.2–1.7 s), beating the cascade by $\sim$0.2 s on *ClaimVerif* and 1.4–4.2 s on dialogue. Act is generation+retrieval under a shared stopwatch that excludes prefill/TTFT (App. E); the lead tracks compact SFT trajectories (cf. GRPO-only Act 3.0–3.9 s at 8B), not trigger frequency. Wait columns are in App. B.

5.3 Analysis

We next attribute the Table 2 gain to individual knobs on the 8B recipe.

(A1) Reward shaping: every term carries weight. Table 3 removes one recipe component at a time. The LLM-judge bonus is the largest quality lever (-0.09 / -0.15 E2E on ClaimVerif / DSTC9-T1): on short-label substrates it is what shapes the answer body. The retrieval-hit bonus helps when hit is unsaturated ($+0.05$ E2E on ClaimVerif; near-zero on DSTC9 near the $\sim$0.92 ceiling). The balanced sampler similarly costs $\leq$0.05 E2E, demonstrating the effectiveness of addressing the wait/act imbalance.

(A2) Structured rewards matter beyond matched SFT or exposing <wait>. Two natural alternatives are that Search-R1 would match ProSearch given the same SFT prior, or simply by exposing a <wait> token. Table 4 tests the first: Search-R1 inherits ProSearch's balanced SFT (trajectories truncated at </search>), then continues with *outcome-only* GRPO (no intermediate

	Search-R1		+ gate		ProSearch	
	Cold	+SFT	Cold	+SFT	Cold	+SFT
ClaimVerif	0.15	0.15	0.31	0.34	0.26	**0.44**
DSTC9-T1	0.43	0.17	0.66	0.75	0.71	**0.84**

Table 4: E2E-F1 when Search-R1 inherits ProSearch's balanced SFT.

Method	Act	Wait	Wait%	E2E
Search-R1 (no <wait>)	1.39	0.46	3.6	0.15
+ <wait>, prompt-only	6.16	1.49	29.4	0.17
+ <wait>, outcome GRPO	1.29	n/a	0.0	0.15
ProSearch	1.35	0.05	82.8	**0.44**

Table 5: <wait> in Search-R1 vs. ProSearch on ClaimVerif (Llama-8B).

Train \ eval	DSTC9	DSTC11
DSTC9 only	**0.84**	0.33
DSTC11 only	0.65	**0.45**
Joint DSTC9 & DSTC11	**0.84**	0.27

Table 7: Cross-dataset E2E-F1. Diagonal: in-domain; off-diagonal: zero-shot OOD.

Hop budget	Act	Wait	Trig	Retr	Resp	E2E	hops
τ=1 (default)	1.51	0.04	0.96	0.92	0.94	0.84	1.00
τ=5	1.62	0.04	0.96	0.94	0.93	**0.86**	**1.04**

Table 8: Multi-hop GRPO on DSTC9-T1. hops = mean retrievals / gold trigger.

rewards). Matched SFT ties or *hurts* cold Search-R1: without structured rewards, SFT priors collapse into always-trigger behavior and retrieval falls to the SFT floor. A post-hoc gate mitigates always-trigger but still trails ProSearch. Table 5 tests the second on ClaimVerif: (i) prompt-only <wait> and (ii) outcome-only GRPO with per-turn shaping zeroed. Prompt-only is $4.4\times$ slower on Act at E2E 0.17; outcome-only recovers Act only by collapsing waits to zero (<wait> gets no terminal credit under that objective). The correct-wait bonus τ_{TN}=0.3 in Eq. 3 supplies that signal (cf. ProSearch's $82.8\,\%$ wait emission rate in Table 5); stronger terminal-only alternatives are noted in App. I. Both controls stay near vanilla Search-R1, so the shaping terms (not the cold-start or the token alone) matter here. Matched-SFT breakdowns are in App. J.

(A3) Effect of retriever and top-K budget. Table 6 shows that BGE vs. BM25 lifts E2E $0.65 \rightarrow 0.84$ on *DSTC9-T1*, driven by retrieval hit ($0.76 \rightarrow 0.92$) on short, paraphrastic FAQs where lexical overlap is brittle. Matched train–eval K peaks at K=5; we adopt BGE with K=5 as the canonical configuration.

	Method		# Ret.			
	BM25	BGE	1	3	5	7
Trig.	0.96	0.96	0.97	0.96	0.96	0.96
Retr.	0.76	0.92	0.82	0.91	0.92	0.94
Resp.	0.90	0.94	0.97	0.94	0.94	0.93
E2E	0.65	0.84	0.77	0.82	0.84	0.83

Table 6: Retriever and top-K on DSTC9-T1.

(A4) Behavior under distribution and hop-budget shift. We next ask how the policy behaves when the stream or retrieval budget changes. Table 7 reports zero-shot OOD and joint training on the dialogue substrates (each cell scored against the eval substrate's native BGE index). Zero-shot OOD drops relative to in-domain; joint training preserves DSTC9-T1 (0.84) but not DSTC11-T5 (0.27 vs. single-task 0.45), consistent with a *retrieval-alignment ceiling* from overlapping-but-distinct gold-anchor regimes. Component rates are in App. J. Separately, as noted in §3.2, max_turns=1 is the latency-sensitive default while $T(\tau)$ already admits multi-hop. Injecting a failure-retry pattern into 15 % of SFT rows and training with max_turns=5 on DSTC9-T1 (Table 8) lifts E2E $0.84 \rightarrow 0.86$ while mean hops move only $1.00 \rightarrow 1.04$ ($+0.11$ s Act): extra hops are used selectively on first-retrieval misses, not indiscriminately. This multi-hop run is a *diagnostic* stress test of the turn counter, not an evaluation on a genuinely multi-hop proactive benchmark.

6 Conclusion

We introduced ProSearch, a single LLM trained end-to-end with GRPO over $\{\text{WAIT}, \text{SEARCH}\}$ that internalises abstention under sparse actionable streams. Three additive reward terms shape trigger, retrieval, and answer behavior inside one group-relative update; a teacher-free cold-start prevents the always-wait collapse. Across ClaimVerif, DSTC9-T1, and DSTC11-T5 at 3B/8B, ProSearch improves end-to-end F1 by $+0.13$–$+0.25$ over the strongest cascade while keeping Act latency competitive. The SFT+structured-reward recipe beats the gated cascades we evaluate (on DSTC mainly via retrieval/response, not trigger); we do not claim uniqueness against every matched modular alternative. We see this as a step toward agents that treat inaction as a first-class action under segment-level, latency-sensitive monitoring.

AI USE STATEMENT

In this work, we used generative AI tools for generating and cleaning synthetic training labels for *ClaimVerif* (`gpt-5.4`-series pipeline; App. D). We have not used generative AI tools for developing theoretical models, formulating or proving mathematical claims, proposing hypotheses, designing research methodology, translation, qualitative analysis, or interpreting results; these required-disclosure tasks are not applicable beyond the dataset pipeline above. Additionally, we used generative AI tools for drafting and editing prose, LaTeX assistance, and assisting with implementation of training and evaluation code. LLM judges used as rewards and metrics are part of the experimental protocol (§3.2), not authoring aids. We have reviewed all AI-assisted work: silver labels were validated against human gold probes and author audits (App. D), and AI-assisted code and prose were checked by the authors before inclusion. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Fact-checking misuse and over-trust. Proactive claim verification can be misused to surface partial or contested evidence as if it were settled, and users may over-trust an automated SUPPORTS/REFUTES label without inspecting the cited passage. We mitigate this by always returning the retrieved evidence span alongside the verdict and recommend that downstream deployments expose both rather than the label alone.

Dataset sourcing and redistribution. *ClaimVerif* is built from publicly available podcast transcripts (Clifton et al., 2020) and web pages retrieved via DuckDuckGo, with text extracted by `trafilatura`. The source podcasts and web pages remain copyrighted by their respective owners; we will release only the generated labels and source URLs (not the scraped page contents) so that downstream users can re-fetch evidence under the original licenses. Podcast transcripts may mention named individuals in factual claims; we retain these mentions as they appear in the public source audio and do not introduce any new personally identifying information.

REPRODUCIBILITY STATEMENT

We will release code, trained checkpoints, the ClaimVerif construction scripts and data splits, and configuration files needed to reproduce all tables in this paper. Training hyperparameters, reward knobs, and evaluation prompts are documented in the appendix.

REFERENCES

Siddhant Arora, Haidar Khan, Kai Sun, Xin Luna Dong, Sajal Choudhary, Seungwhan Moon, Xinyuan Zhang, Adithya Sagar, Surya Teja Appini, Kaushik Patnaik, et al. Stream RAG: Instant and accurate spoken dialogue systems with streaming tool usage. *arXiv preprint arXiv:2510.02044*, 2025.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *ICLR*, 2024.

Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, et al. 100,000 podcasts: A spoken english document corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 5903–5917, 2020.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, et al. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dongxin Guo, Jikun Wu, and Siu Ming Yiu. When to retrieve during reasoning: Adaptive retrieval for large reasoning models. *arXiv preprint arXiv:2604.26649*, 2026.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. A survey on automated fact verification. *Knowledge-Based Systems*, 2022.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *ICML*, 2020.

Helia Hashemi, Victor Rühle, and Saravan Rajmohan. Cost-aware retrieval-augmentation reasoning models with adaptive retrieval depth. *arXiv preprint arXiv:2510.15719*, 2026.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *NAACL*, 2024.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *EMNLP*, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Ö. Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tur. Beyond domain APIs: Task-oriented conversational modeling with unstructured knowledge access. In *SIGdial*, 2020.

Seokhwan Kim, Spandana Gella, Chao Zhao, Di Jin, Alexandros Papangelis, Behnam Hedayatnia, Yang Liu, and Dilek Z Hakkani-Tür. Task-oriented conversational modeling with subjective knowledge track in dstc11. In *Proceedings of The Eleventh Dialog System Technology Challenge*, pp. 274–281, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

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

Xingyu Bruce Liu, Shitao Fang, Weiyan Shi, Chien-Sheng Wu, Takeo Igarashi, and Xiang'Anthony' Chen. Proactive conversational agents with inner thoughts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–19, 2025.

Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, et al. Proactive agent: Shifting llm agents from reactive responses to active assistance. In *International Conference on Learning Representations*, volume 2025, pp. 47431–47457, 2025.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2022.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *NeurIPS*, 2024.

Vinay Setty. Fact-checking multilingual podcasts. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pp. 1100–1101, 2025.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the information needs of large language models. *arXiv preprint arXiv:2403.10081*, 2024.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*, 2018a.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: A large-scale dataset for fact extraction and verification. In *NAACL*, 2018b.

Junlong Tong, Yingqi Fan, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. StreamingThinker: Large language models can think while reading. *arXiv preprint arXiv:2510.17238*, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *ACL*, 2023.

Venktesh V and Vinay Setty. Livefc: A system for live fact-checking of audio streams. *arXiv preprint arXiv:2408.07448*, 2024.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *EMNLP*, 2020.

Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 641–649, 2024.

Bufang Yang, Lilin Xu, Liekang Zeng, Kaiwei Liu, Siyang Jiang, Wenrui Lu, Hongkai Chen, Xiaofan Jiang, Guoliang Xing, and Zhenyu Yan. Contextagent: Context-aware proactive llm agents with open-world sensory perceptions. *Advances in Neural Information Processing Systems*, 38: 167509–167543, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2023.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, Xiaojun Chang, Junge Zhang, Feng Yin, Yitao Liang, and Yaodong Yang. ProAgent: Building proactive cooperative agents with large language models. In *AAAI*, 2024a.

Xuan Zhang, Yang Deng, Zifeng Ren, See Kiong Ng, and Tat-Seng Chua. Ask-before-plan: Proactive language agents for real-world planning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10836–10863, 2024b.

Sebastian Zhao, Alan Zhu, Hussein Mozannar, David Sontag, Ameet Talwalkar, and Valerie Chen. Codinggenie: A proactive llm-powered programming assistant. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, pp. 1168–1172, 2025.

A Model and Training Details

This appendix completes the recipe of the main-text *Model and training* paragraph with the per-stage, per-backbone training hyperparameters (Table 9) and the eight reward-knob values of Eq. 2 (Table 10). All experiments run on a single 8×H100 (80 GB) node; implementation is built on `verl` (Sheng et al., 2024) with a co-located vLLM (Kwon et al., 2023) rollout engine.

Training hyperparameters. Table 9 fills in the per-cell values for the two training stages. The SFT cold-start uses a leading-token decision-boost ($\times 30$ on the first 3 response tokens) to counter the response-length asymmetry between `<wait>` and full retrieval trajectories. GRPO uses a low-variance KL penalty (`kl_loss_type` = *low_var_kl*) and a tight clipping ratio; both regularisers are required to avoid the always-wait collapse basin we observed during early development. State-masking (`state_masking` = *true*) hides the retrieved `<information>` tokens from the policy-gradient computation so the gradient flows only through the model's own emissions. Checkpoints are saved every 25 steps; the reported checkpoint is the one whose validation judge score peaks while $\|\mathrm{KL}\|$ remains below 0.5. Where the GRPO learning rate differs by substrate, it is chosen as the value that maximises that validation criterion within the KL-clean range. Cascade baselines do *not* use this sweep: Base is the off-the-shelf checkpoint; Search-R1 follows the published think–search–answer recipe with fixed learning rates on actionable-only data; the trigger gate is a separately SFT'd classifier on g_{trig} (final epoch).

Reward hyperparameters. Table 10 re-states the eight reward-knob values of Eq. 2 in a compact lookup form, together with a one-line semantic for each knob. Values are shared across substrates (and were fixed after early validation pilots, not a full multi-dimensional grid); the only per-substrate knobs are the judge weight w_j (1.0 on dialogue, 0.5 on *ClaimVerif*) and w_v (ClaimVerif only). Table 3 ablates the presence of reward terms and the sampler, not the full coefficient surface.

Datasets, corpora, and retriever. Class-balanced training triples (minority trigger class up-sampled 1:1): ∼28K (ClaimVerif), ∼104K (DSTC9-T1), ∼30K (DSTC11-T5). Indexed retrieval corpora: 20,767 ClaimVerif passages from 19,351 unique URLs, 12,039 (DSTC9-T1), 10,882 (DSTC11-T5). The dense retriever is *BAAI/bge-large-en-v1.5* (Xiao et al., 2024), served via a long-lived FAISS index. Following the main text, K=3 on *ClaimVerif* and K=5 on the DSTC substrates.

SFT			**GRPO**		
Knob	**3B**	**8B**	**Knob**	**3B**	**8B**
Learning rate	$5\cdot10^{-6}$	$2\cdot10^{-6}$	Group size n_{agent}	3	3
Epochs	2	2	Learning rate	$7\cdot10^{-7}$	$2\cdot10^{-6}$
Global / micro batch	64 / 8	64 / 8	`kl_loss_coef`	0.05	0.05
Warmup ratio	5%	5%	`clip_ratio`	0.05	0.05
Decision-boost	30	30	Rollout temperature	1.0	1.0
Max seq. length	4096	4096	State-masking	on	on
			Optimization steps	400	400

Table 9: Per-stage, per-backbone training hyperparameters. Both stages use AdamW with weight decay 0. SFT uses gradient clip 1.0; GRPO uses gradient clip 0.5. The same recipe skeleton is used across the three substrates; GRPO learning rate / checkpoint selection uses validation as described above.

B Full Per-Substrate Main Results

Table 2 reports Act latency with Trigger F1, Retrieval Hit, Resp, and E2E-F1 for compactness. Tables 11, 12, and 13 give the full per-substrate breakdown: Trigger Precision/Recall (which decompose the main-table F1), Wait latency on non-triggered segments, and Response Miss (verdict/judge score when retrieval fails).

One thing worth noting is that Wait latency is deferred from the main table because Act is the binding cost on triggered turns. Under the shared stopwatch of App. E, headline ProSearch (SFT+GRPO) Wait is 0.04–0.05 s (emitting `<wait>`), gated cascades pay ∼0.01 s (gate forward pass), and ungated Base

Component	Knob	Value	Comment
Trigger shaping	τ_{TN}	0.3	wait-correct reward
	τ_{FP}	0.1	false-trigger penalty
	τ_{FN}	0.0	missed trigger (silence floor)
Retrieval shaping	ρ_{hit}	1.0	top-K retrieval hit
	ρ_{miss}	0.5	retrieval miss baseline
Latency	β	0.3	turn-count discount on $T(\tau)$
Verdict bonus	w_v	0.5	ClaimVerif only (no-op on dialogue)
Judge bonus	w_j (*ClaimVerif*)	0.5	`factv2_justification`
	w_j (*DSTC9-T1*, *DSTC11-T5*)	1.0	`factual_on_TPs`

Table 10: Reward hyperparameters from Eq. 2, set to the canonical ProSearch values. Penalty knobs (τ_{FP}, τ_{FN}) enter Eq. 3 with a negative sign; the values listed are magnitudes. The substrate-dependent split on w_j is the only per-task knob.

	Latency		Trigger			Retrieval	Response		E2E-F1
	Act	Wait	Precision	Recall	F1		Hit	Miss	
Llama-3.2-3B-Instruct									
Base	4.05	0.72	0.21	0.94	0.35	0.53	0.49	0.36	0.08
Base (*w/* gate)	3.89	0.01	0.67	0.68	0.68	0.50	0.50	0.37	0.17
Search-R1	1.40	0.08	0.21	1.00	0.35	0.43	0.84	0.80	0.13
Search-R1 (*w/* gate)	1.37	0.02	0.67	0.73	0.70	0.47	0.82	0.78	0.27
ProSearch (SFT)	1.58	0.04	0.83	0.37	0.51	0.66	0.76	0.74	0.26
ProSearch (GRPO)	1.66	0.04	0.58	0.55	0.57	0.62	0.64	0.55	0.23
ProSearch (SFT + GRPO)	1.20	0.04	0.73	0.57	0.64	0.77	0.83	0.72	0.40
Llama-3.1-8B-Instruct									
Base	6.62	1.83	0.50	0.68	0.58	0.46	0.58	0.44	0.09
Base (*w/* gate)	5.94	0.01	0.67	0.73	0.70	0.52	0.57	0.44	0.21
Search-R1	1.39	0.46	0.22	1.00	0.36	0.50	0.84	0.79	0.15
Search-R1 (*w/* gate)	1.51	0.01	0.67	0.73	0.70	0.53	0.83	0.76	0.31
ProSearch (SFT)	1.75	0.05	0.76	0.57	0.65	0.72	0.81	0.68	0.38
ProSearch (GRPO)	3.82	0.91	0.46	0.83	0.59	0.66	0.67	0.56	0.26
ProSearch (SFT + GRPO)	1.35	0.05	0.74	0.60	0.66	0.79	0.87	0.75	0.44

Table 11: Full results on *ClaimVerif*.

is 0.7–2.7 s because it writes a free-form refusal on every no-action turn (GRPO-only Wait can be higher when wait emissions are verbose). Trigger P/R shows the ungated Base / Search-R1 signature of near-saturating recall with precision near the gold-trigger rate; gated rows recover precision at matched recall. Response Miss is typically below Response Hit, so E2E credit is not driven by answers that ignore a retrieval failure.

C DSTC9-T1 & DSTC11-T5 Data Conversion

DSTC9-T1: We convert the DSTC9-T1 dataset into the ProSearch format: The hierarchical knowledge base (4 domains, 143 entities, ∼2,900 FAQ snippets for training; 5 domains, 666 entities, ∼12,039 snippets for evaluation) is flattened into an ID-keyed corpus (`domain__entity_id__doc_id`) that we later index with BGE (and with BM25 in the retriever ablation of Table 6). Each dialogue instance becomes one training sample: the binary `target` field maps to our trigger label; knowledge references are resolved to evidence text; and the `response` field provides the ground-truth outcome. The converted dataset totals 71,348 training, 9,663 validation, and 4,181 test samples.

DSTC11-T5: Unlike DSTC9-T1 (factual FAQs only), DSTC11-T5 (Kim et al., 2023) introduces *subjective knowledge* from customer reviews alongside FAQs, requiring the agent to detect when review-grounded responses are appropriate. The knowledge base spans 2 domains (ho-

	Latency		Trigger			Retrieval	Response		E2E-F1
	Act	Wait	Precision	Recall	F1		Hit	Miss	
Llama-3.2-3B-Instruct									
Base	3.53	1.09	0.47	0.99	0.64	0.75	0.70	0.51	0.33
Base (*w/* gate)	2.92	0.01	1.00	0.94	0.97	0.74	0.70	0.49	0.50
Search-R1	3.47	0.86	0.49	1.00	0.65	0.76	0.78	0.47	0.39
Search-R1 (*w/* gate)	2.57	0.00	1.00	0.95	0.97	0.75	0.76	0.50	0.56
ProSearch (SFT)	1.14	0.04	1.00	0.91	0.95	0.90	0.68	0.44	0.59
ProSearch (GRPO)	1.24	0.77	0.94	0.91	0.93	0.87	0.85	0.37	0.69
ProSearch (SFT + GRPO)	1.21	0.04	1.00	0.93	0.96	0.93	0.89	0.46	0.79
Llama-3.1-8B-Instruct									
Base	6.03	2.74	0.48	1.00	0.64	0.80	0.78	0.57	0.40
Base (*w/* gate)	4.75	0.01	1.00	0.95	0.97	0.80	0.79	0.56	0.61
Search-R1	4.90	1.04	0.47	1.00	0.64	0.82	0.83	0.54	0.43
Search-R1 (*w/* gate)	3.64	0.00	1.00	0.95	0.97	0.81	0.83	0.54	0.66
ProSearch (SFT)	1.44	0.04	1.00	0.92	0.96	0.92	0.79	0.43	0.69
ProSearch (GRPO)	2.99	0.22	0.91	0.98	0.95	0.90	0.83	0.46	0.71
ProSearch (SFT + GRPO)	1.51	0.04	1.00	0.93	0.96	0.92	0.94	0.41	0.84

Table 12: Full results on *DSTC9-T1*.

	Latency		Trigger			Retrieval	Response		E2E-F1
	Act	Wait	Precision	Recall	F1		Hit	Miss	
Llama-3.2-3B-Instruct									
Base	3.75	0.94	0.51	0.98	0.67	0.27	0.54	0.49	0.10
Base (*w/* gate)	3.59	0.02	1.00	0.98	0.99	0.27	0.54	0.49	0.15
Search-R1	4.86	0.31	0.53	0.99	0.69	0.37	0.54	0.50	0.14
Search-R1 (*w/* gate)	4.85	0.01	1.00	0.99	0.99	0.37	0.54	0.50	0.20
ProSearch (SFT)	1.31	0.04	1.00	1.00	1.00	0.49	0.50	0.36	0.25
ProSearch (GRPO)	2.69	0.10	0.99	1.00	0.99	0.51	0.59	0.50	0.30
ProSearch (SFT + GRPO)	1.27	0.04	1.00	1.00	1.00	0.57	0.75	0.41	0.43
Llama-3.1-8B-Instruct									
Base	6.34	1.96	0.52	1.00	0.68	0.32	0.58	0.53	0.13
Base (*w/* gate)	6.07	0.01	1.00	1.00	1.00	0.32	0.58	0.53	0.19
Search-R1	5.85	0.75	0.51	1.00	0.68	0.37	0.55	0.53	0.14
Search-R1 (*w/* gate)	5.90	0.01	1.00	1.00	1.00	0.37	0.55	0.55	0.20
ProSearch (SFT)	1.56	0.04	1.00	1.00	1.00	0.49	0.50	0.33	0.24
ProSearch (GRPO)	3.88	0.10	0.99	0.97	0.98	0.57	0.58	0.49	0.32
ProSearch (SFT + GRPO)	1.64	0.04	1.00	1.00	1.00	0.58	0.73	0.43	0.45

Table 13: Full results on *DSTC11-T5*.

tel, restaurant) with 143 entities and is composed of two knowledge types: 2,869 FAQ snippets (question–answer pairs) and 8,013 review sentences (sentence-level subjective content), totaling 10,882 documents. Each snippet receives a unique hierarchical ID (e.g., `hotel__0__faq__3` or `hotel__0__review__2__1`). Each dialogue instance is converted identically to DSTC9-T1: the binary `target` field maps to our trigger label; knowledge references are resolved to evidence text (FAQ body or review sentence); and the `response` field provides the ground-truth outcome. The converted dataset totals 28,431 training, 4,173 validation, and 5,475 test samples, with a trigger rate of $\sim$51.9% (compared to 26.9% in DSTC9-T1). Among triggered samples, 93.4% reference reviews and 6.6% reference FAQs, highlighting the subjective-knowledge emphasis of this benchmark.

D ClaimVerif Dataset Creation Details

This appendix supplements §4 with reproducibility knobs and cleaning numbers; the three-stage pipeline itself is unchanged from the main text. Full scripts are released with the code.

Model tiers and generation. Claim detection and stance use `gpt-5.4-mini` (T=0.3); evidence harvesting uses `gpt-5.4` (T=0.2) for query generation and snippet extraction (fallback synthesis at T=0.5). Claim detection processes utterances in batches of 15 with the same 10 balanced few-shot exemplars as the main text. Evidence pages are fetched asynchronously (15 s timeout), truncated to ∼6k characters before extraction; null / non-200 pages are dropped, yielding avg. 2.0 snippets/claim. Stance responses that fail JSON parse fall back to keyword matching; remaining ambiguous pairs are discarded. The natural stance split (82.1% SUPPORTS / 17.9% REFUTES) is preserved without resampling.

Timestamp alignment. Annotations are keyed by utterance text; RL rollouts need Whisper word timestamps. Each claim is fuzzy-matched to the Whisper word sequence (sliding window, step 3; 80% similarity threshold, early exit above 95%). Matched span start/end times become the `(is_trigger, evidence, verdict, start, end)` tuples. Claims shorter than 5 tokens are skipped.

Cleaning, splits, and residual URL overlap. The pipeline produces 38,134 raw utterances upstream of cleaning. Two passes shrink this to the 25,132 RL-ready samples of Table 1: (i) drop incomplete stance files and trigger claims with no stance row (4,059 removed → 34,075); (ii) drop timestamp-alignment failures (8,943, mostly short ASR fragments). Episode-disjoint 70/15/15 partitioning (28/6/6 of 40 episodes; seed 42) gives the 18,493 / 2,924 / 3,715 train/val/test rows. Training is trigger-balanced 50/50 by oversampling; evaluation uses the natural distribution. Exact gold-URL overlap across splits is low (1.95% of test URLs, 2.99% of val) and does not affect episode-disjointness of the queries.

Label quality validation. Table 14 reports silver recall/precision under two probes that share the same metric schema. On the podcasts annotated by Setty (2025) (multi-annotator gold; n=1,960 utterances, n=621 unanimous stance pairs), trigger recall = 1.00 with precision 0.81 (errors are one-sided false positives) and stance recall/precision is 0.76–0.86 on both SUPPORTS and REFUTES; that gold split is too small for robust end-to-end evaluation and is used only as a pipeline probe, not for Tables 2–3. Separately, because that set does not cover our episode-disjoint ClaimVerif test podcasts, we adjudicate a stratified sample of n=200 test utterances (mix of triggered and non-triggered) and n=39 stance pairs among triggered rows covering both SUPPORTS and REFUTES, judging the silver trigger against the utterance and, when triggered, the silver stance against the linked evidence span. On this audit, trigger recall/precision is 0.93/0.78 and stance recall/precision is 0.75–0.87 (S/R in Table 14). Headline E2E tables remain on the full silver test split.

Probe	Axis	Sample size	Silver recall	Silver precision
Setty (2025) (multi-annot. gold)	Utterances	1,960	1.00	0.81
	Stance	621	0.86 / 0.79 (S/R)†	0.80 / 0.76 (S/R)†
ClaimVerif test (author audit)	Utterances	200	0.93	0.78
	Stance	39	0.84 / 0.75 (S/R)†	0.87 / 0.75 (S/R)†

Table 14: Silver pipeline vs. human judgment under two probes with a shared recall/precision schema. The Setty (2025) block uses prior multi-annotator gold; the ClaimVerif-test block is author adjudication on our episode-disjoint evaluation episodes. Utterance rows score the trigger decision; stance rows score SUPPORTS/REFUTES on (claim, evidence) pairs. †S/R = SUPPORTS / REFUTES.

E Evaluation Metrics: Full Definitions

Per row i: gold trigger $g_i \in \{0,1\}$, action $a_i \in \{\texttt{wait}, \texttt{search}\}$, gold evidence $e_i^\star$ (possibly null), retrieved set $\mathcal{R}_i$, answer $\hat{y}_i$.

$$\begin{aligned} \textsc{Trig}_i &= \mathbf{1}[a_i = \texttt{search}], \\ \textsc{Hit}_i &= \mathbf{1}[e_i^\star \in \mathcal{R}_i], \\ \textsc{Resp}_i &= \begin{cases} \mathbf{1}[\hat{y}_i = y_i^\star] & \text{(ClaimVerif)} \\ s_j(\hat{y}_i; y_i^\star)/5 & \text{(DSTC9/11)} \end{cases} \end{aligned}$$

	DSTC9-T1			*DSTC11-T5*		
Cell	**Llama/5**	**GPT/5**	Δ	**Llama/5**	**GPT/5**	Δ
Base	0.780	0.657	+0.123	0.580	0.493	+0.087
Base (*w/* gate)	0.790	0.629	+0.161	0.580	0.491	+0.089
Search-R1	0.830	0.547	+0.283	0.550	0.395	+0.155
Search-R1 (*w/* gate)	0.830	0.541	+0.289	0.550	0.389	+0.161
ProSearch (GRPO)	0.830	0.704	+0.126	0.580	0.475	+0.105
ProSearch (SFT)	0.790	0.771	+0.019	0.500	0.489	+0.011
ProSearch (SFT+GRPO, headline)	**0.940**	**0.815**	+0.125	**0.730**	**0.579**	+0.151
Agreement @ *both* ≥ 4 (n-weighted)	0.85			0.80		

Table 15: Independent-judge re-grading of the dialogue substrates of Table 2. **Llama/5** and **GPT/5** are normalised judge scores on TPs; the Llama column matches the **Resp.** column of Table 2. Absolute scores are judge-sensitive (Δ), but ProSearch remains the top cell under *both* judges and binary agreement on $[s_j \geq 4]$ is high. Full per-cell numbers (including ablations) are released with the code.

where $s_j \in \{0, \ldots, 5\}$ is the Llama-3.1-8B-Instruct `factual_on_TPs` score (normalised by 5 in Resp/E2E). Mid-trajectory `<search>` still counts as triggered even if the final token is `<wait/>`.

Trigger / retrieval / response. Trigger P/R/F1 are standard binary metrics of $\textsc{Trig}_i$ vs. g_i. Retrieval hit is the mean of $\textsc{Hit}_i$ over TPs ($g_i = \textsc{Trig}_i = 1$) with defined gold evidence (other on-topic passages do not count; cf. Case 12). Response is the mean of $\textsc{Resp}_i$ over true positives (verdict match on ClaimVerif; LLM-judge factuality on DSTC9/11).

E2E-F1. Computed over the full test set. Per-row credit

$$\textit{credit}_i \;:=\; \textsc{Trig}_i \cdot g_i \cdot \text{retrieval_ok}_i \cdot \textsc{Resp}_i,$$

with $\text{retrieval_ok}_i := \neg\, e_i^\star \vee \textsc{Hit}_i$. Precision / recall sum $\textit{credit}_i$ over model-triggered / gold-triggered rows; E2E-F1 is their harmonic mean. The **Resp.** column of Table 2 averages the same $\textsc{Resp}_i$ over TPs without the trigger/retrieval gate.

Latency. Wall-clock seconds from the first *generated* token to the terminal `</answer>` / `<wait/>`, including retrieval on triggered rows. Prompt prefill and time-to-first-token are excluded for all methods, so Act/Wait measure generation+retrieval cost rather than end-to-end user latency. We report them separately (mean ℓ_i conditioned on $\textsc{Trig}_i = 1$ / 0): averaging would favour models that trigger less often. For gated cascades, Wait includes the gate forward pass ($\sim$0.01 s); on Act rows that pass precedes decoding and is omitted from Act; adding it back ($\sim$+0.01 s) does not change ranking. Act remains the primary cross-method latency comparison under this stopwatch.

F Robustness to the Choice of LLM-as-Judge

Dialogue response scores use an LLM judge; ClaimVerif E2E does not (hard $\textit{verdict_correct}_i$; App. E). To check family overlap between the GRPO reward judge and the grader, we re-grade every DSTC9-T1 / DSTC11-T5 cell of Table 2 with an independent *gpt-5.4-mini* judge on the same true-positive rows of the full test set (Table 15). Absolute Resp scores are judge-sensitive (paired Δ in the table), but method ranking is stable: ProSearch remains the top cell under both judges and above all four cascade baselines; binary agreement on $[s_j \geq 4]$ is 0.85 / 0.80. We do not report human–judge agreement. Rubric anchors are in App. H.

G Cross-Family Transfer: Ministral-8B

To test whether the Table 2 conclusions are Llama-specific, we rerun the identical ProSearch SFT+GRPO recipe on `Ministral-8B-Instruct-2410` (Mistral family; size-matched to Llama-8B, different attention and tokenizer). Tables 16–17 report the full cascade matrix on ClaimVerif and DSTC9-T1; a Llama-8B reference column is included for calibration. The headline qualitative conclusions transfer: ProSearch (SFT+GRPO) beats Search-R1+gate (0.41 vs. 0.30, +0.11

on ClaimVerif; 0.86 vs. 0.70, +0.16 on DSTC9-T1) and beats SFT-only (0.41 vs. 0.37, +0.04; 0.86 vs. 0.76, +0.10). Absolute E2E-F1 reaches 0.41 / 0.86, or 93–102 % of the Llama-8B scores (0.44 / 0.84). Two Ministral-specific caveats: on ClaimVerif, Base+gate (0.13) underperforms ungated Base (0.23), so the full Base→Search-R1 cascade ordering is weaker than on Llama; and GRPO-only Act latency is anomalously low (0.26 s), so the "GRPO-without-SFT is slow" latency story in (R2)–(R3) should be read as Llama-primary.

Row	Act ↓	Trig-F1	Retr	Resp	E2E-F1	*Llama-8B*
Base	2.04	0.52	0.58	0.73	0.23	*0.09*
Base + gate	2.14	0.30	0.64	0.67	0.13	*0.21*
Search-R1	1.86	0.37	0.46	0.82	0.15	*0.15*
Search-R1 + gate	2.15	0.70	0.52	0.81	0.30	*0.31*
ProSearch (SFT)	0.99	0.64	0.70	0.81	0.37	*0.38*
ProSearch (GRPO)	0.26	0.62	0.70	0.81	0.36	*0.26*
ProSearch (SFT+GRPO)	1.68	**0.67**	**0.75**	0.81	**0.41**	*0.44*

Table 16: Ministral-8B results on ClaimVerif. Columns follow Table 2; the last column repeats the Llama-8B cell for calibration.

Row	Act ↓	Trig-F1	Retr	Resp	E2E-F1	*Llama-8B*
Base	1.61	0.66	0.84	0.77	0.47	*0.40*
Base + gate	1.33	0.52	0.81	0.75	0.34	*0.61*
Search-R1	1.98	0.64	0.85	0.81	0.47	*0.43*
Search-R1 + gate	1.54	0.96	0.84	0.80	0.70	*0.66*
ProSearch (SFT)	0.98	0.94	0.92	0.84	0.76	*0.69*
ProSearch (GRPO)	1.06	0.93	0.91	0.87	0.78	*0.71*
ProSearch (SFT+GRPO)	1.09	**0.98**	**0.93**	**0.90**	**0.86**	*0.84*

Table 17: Ministral-8B results on DSTC9-T1. Columns as in Table 16.

H Judge Rubrics and Prompts

Dialogue response scoring uses `factual_on_TPs`; ClaimVerif justification training uses `factv2_justification` (ClaimVerif *headline* E2E response credit is hard `verdict_match`, not this judge). Both enter Eq. 5 as $s/S_{\max}$. Prompt anchors:

- `factual_on_TPs` (DSTC9-T1, DSTC11-T5): 0 off-topic/wrong-entity; 1 contradiction; 3 safe abstention; 4 consistent-but-imprecise; 5 fully correct.
- `factv2_justification` (ClaimVerif train): 0 vacuous template; 1 argues-for-opposite-verdict; 2 ambiguous hedge; 4 strong evidence-span citation.

Both rubrics share a `JUDGE_SYSTEM` prompt that requires single-JSON-line output and forbids middle-score defaults (explicit anti-3-bias instruction). Full prompt text and metrics code are released with the codebase.

I Limitations

Single-shot retrieval as a domain choice. Under streaming latency constraints we default to `max_turns`=1: the target substrates are largely single-hop (*DSTC9-T1* resolves ∼93 % of gold anchors on the first retrieval; ClaimVerif verdicts are per-claim). This is a substrate choice, not an architectural limit: the turn counter already admits multi-hop, and analysis (A4) shows that injecting a failure-retry pattern into 15 % of SFT rows and training with `max_turns`=5 lifts retrieval and E2E-F1 selectively (0.84 → 0.86) without indiscriminate multi-hop use. Systematic characterization on genuinely multi-hop streams is future work.

Joint-corpus training and the retrieval-alignment ceiling. Zero-shot trigger calibration transfers strongly across DSTC9 and DSTC11 (analysis A4), but training one policy on the union corpus

preserves DSTC9 in-domain while capping DSTC11 below the zero-shot OOD baseline. We attribute this to overlapping-but-distinct retrieval regimes that a shared retriever cannot simultaneously optimise; domain-conditional retrieval heads are a natural next step.

Silver ClaimVerif labels and statistical uncertainty. Headline ClaimVerif metrics use silver train/test labels over a frozen harvested corpus with a natural $82.1/17.9$ SUPPORTS/REFUTES prior (majority-class shortcut risk). The Setty gold probe and our author test-episode audit (n=200 utterances; n=39 stance pairs; App. D) bound label noise but are not a full human test split with inter-annotator agreement. DSTC9/11 supply human-annotated triggers and knowledge. Dialogue Resp/E2E use an LLM judge; rankings are robust under an independent judge (App. F). We do not report multi-seed confidence intervals on the main or ablation tables.

Stronger controls left open. Our cascades include Search-R1 with matched SFT and an external gate (analysis A2), but we do not (i) train an execution-only ProSearch generator (trigger rewards zeroed) and compose it post-hoc with the same gate, or (ii) run same-SFT GRPO under a sparse terminal reward that credits both correct waiting and full trigger–retrieve–answer success without mid-term r_{ret}/r_{ans} shaping. Whether either closes the remaining gap is left open.

Clean-text input and static corpora. All three substrates feed pre-segmented text rather than raw automatic speech recognition (ASR), and retrieval corpora are frozen snapshots at evaluation. A deployed real-time agent would face boundary ambiguity, disfluencies, and corpus drift; closing these gaps is left to future work.

Scale beyond 8B. Due to resource constraints, we only validate the recipe on Llama-3.x at 3B/8B and, for cross-family transfer, on Ministral-8B (App. G). Whether the same gains persist at substantially larger scales remains open.

J Additional Analysis Tables

OOD component decomposition. Table 18 expands the E2E-only cross-dataset matrix of Table 7 into trigger rate, gold-trigger rate, Trig-F1, retrieval hit, and response. The "always-wait" failure mode is what an *always-wait prompt ablation* of Base exhibits (trigger rate 0.06–0.09 vs. GT $0.47/0.51$)—not the Table 2 Base row, and not ProSearch: OOD Trig-F1 remains 0.98 (DSTC9→DSTC11) and 0.84 (DSTC11→DSTC9). The residual OOD E2E gap sits in retrieval hit and response quality.

Model	Setting	Trig rate	GT	Trig-F1	Retr	Resp	E2E
Base 8B (prompted)	DSTC9 → DSTC9	0.09	0.47	0.09	0.98	0.80	0.07
Base 8B (prompted)	DSTC11 → DSTC11	0.06	0.51	0.06	0.55	0.57	0.02
ProSearch	DSTC9 → DSTC9	0.44	0.47	0.96	0.92	0.94	0.84
ProSearch	DSTC11 → DSTC11	0.51	0.51	1.00	0.58	0.73	0.45
ProSearch	DSTC9 → DSTC11 (OOD)	0.53	0.51	0.98	0.57	0.47	0.33
ProSearch	DSTC11 → DSTC9 (OOD)	0.42	0.47	0.84	0.57	0.83	0.65

Table 18: Component-level OOD transfer (8B). Each cell is scored against the eval substrate's native BGE index.

Matched-SFT Search-R1: full components. Table 19 expands Table 4 with Trig-F1 (rate), retrieval hit, and response. Under continued outcome-only GRPO the matched-SFT variant collapses to always-trigger (rate 1.00) while retrieval drops toward the SFT floor; ProSearch remains $+0.29$ / $+0.40$ E2E above the better Search-R1 variant.

K Prompt Templates

Both task domains share the same top-level action space (`<wait>` / `<search>`), with `<answer>` emitted inside the search branch, and task-specific instructions.

Substrate	Training source	Trig-F1	Retr	Resp	E2E
ClaimVerif	SFT-only (matched dist., no RL)	0.35 (1.00)	0.47	0.71	0.13
ClaimVerif	Search-R1 cold (paper)	0.36 (0.96)	0.50	0.84	0.15
ClaimVerif	Search-R1 + matched SFT	0.35 (1.00)	0.51	0.82	0.15
ClaimVerif	ProSearch (SFT+GRPO)	0.66	0.79	0.87	**0.44**
DSTC9-T1	SFT-only (matched dist., no RL)	0.64 (1.00)	0.35	0.57	0.17
DSTC9-T1	Search-R1 cold (paper)	0.64 (1.00)	0.82	0.83	0.43
DSTC9-T1	Search-R1 + matched SFT	0.64 (1.00)	0.42	0.60	0.17
DSTC9-T1	ProSearch (SFT+GRPO)	0.96	0.92	0.94	**0.84**

Table 19: Full component breakdown for the matched-SFT Search-R1 control (Llama-8B). Build: truncate ProSearch SFT trajectories at `</search>`, SFT on that distribution, then outcome-only GRPO (no r_{trig}, r_{ret}, or $b(\tau)$).

Claim verification

```
You are a proactive agent monitoring a live information stream.
For each segment, you must decide:
1. If this segment does NOT require action, respond with <wait/>
2. If it DOES require action, search for relevant knowledge:
  <search>your search query</search>
3. After reviewing retrieved evidence, commit to a response:
  <answer>SUPPORTS</answer> or <answer>REFUTES</answer>
Think step by step in <think>...</think> tags before acting.
```

Knowledge-grounded dialogue

```
You are a proactive agent monitoring a customer service dialogue.
For each turn, you must decide:
1. If the turn can be handled without external knowledge:  <wait/>
2. If external knowledge is needed, search the FAQ:
  <search>your search query</search>
3. After reviewing FAQ results, provide a grounded answer:
  <answer>your response based on the FAQ knowledge</answer>
Think step by step in <think>...</think> tags before acting.
```

L Case Studies

We present qualitative examples of ProSearch agent behavior across all three evaluation domains. For each domain, we show four representative cases illustrating: (A) correct proactive retrieval with a grounded response, (B) correct abstention when no external knowledge is needed, (C) a harder case requiring contextual reasoning (e.g., pronoun resolution, multi-anchor evidence), and (D) a representative *failure*, one per failure axis surfaced by the per-row F1 decomposition in App. E: a stance miscalibration (response axis) on *ClaimVerif*, a spurious trigger (false-positive trigger axis) on *DSTC9-T1*, and a polarity-mismatched retrieval (retrieval-hit axis) on *DSTC11-T5*. Failure rows are taken verbatim from the released test-split logs of the headline 8B ProSearch cells. In every trace, color-coded tags indicate the agent's reasoning pipeline: think = internal reasoning, search = query formulation, information = retrieved evidence, answer = final response, wait = correct abstention.

Case 1: Claim Verification (ClaimVerif) — Correct Proactive Retrieval

Stream context: *"Right wing forces would weaponize a ban on TikTok to try to drive a wedge between young voters Democrats need and the 80-year-old president running for reelection."*
Segment to verify: "many young people who are using the app who suddenly see it go away if a ban goes into effect have heard the reasoning for banning the app in the first place."
Ground truth: `SUPPORTS` **Score:** 4/4

think The segment asserts something factual; I need to retrieve documents on this.
search many young people who are using the app who suddenly see it go away if a ban goes into effect have heard the reasoning for banning the app in the first place
information Doc 1 ("powelltate.com"): *adults ages 18–27, who are over twice as likely to use TikTok as their counterparts in other generations, are significantly more likely to believe the ban comes from a desire to limit free speech or competition...*
answer [SUPPORTS] The retrieved evidence directly confirms the claim.

Case 2: Claim Verification (ClaimVerif) — Correct Abstention

Stream context: *"...the effort for accountability, both legal and political; even if nothing happens with a criminal case, a screw-up by officials trying to charge would be weaponized..."*
Segment to verify: "Maybe he'll be indicted for multiple crimes in multiple districts, but it seems very, very possible that there will not be a trial, a conviction or anything else."
Ground truth: No action needed (opinion/speculation) **Expected:** `<wait/>`

wait/ *Agent correctly identifies the segment as unverifiable speculation and abstains.*

Case 3: Claim Verification (ClaimVerif) — Multi-anchor Evidence

Stream context: *"...will Jack Smith now beat Alvin Bragg to get the first indictment of Donald Trump? ... special counsel Jack Smith is out there ... a federal judge has ordered Mike Pence to testify ..."*
Segment to verify: "So the January 6th investigation by the special counsel is proceeding."
Ground truth: `SUPPORTS` **Score:** 3/4

think The segment asserts something factual; I need to retrieve documents on this.
search the January 6th investigation by the special counsel is proceeding
information Doc 1 ("digital.library.unt.edu"): *On May 17, 2017, Robert S. Mueller III was appointed as Special Counsel ... the investigation's scope included allegations of coordination between Donald Trump's presidential campaign and the Russian government...*
answer [SUPPORTS] The retrieved evidence directly confirms the claim.

Case 4: Claim Verification (ClaimVerif) — Stance Miscalibration (Failure)

Segment to verify: "he was convicted of electoral interference."
Retrieved evidence (gold-anchored): *"After his success in the 2024 vote, the various legal issues he had been battling have largely evaporated. The interference case has now been dismissed."*
Ground truth: `REFUTES` **Model verdict:** `SUPPORTS`

search convicted of electoral interference 2024
answer `[SUPPORTS]` The retrieved evidence directly confirms the claim.

Diagnosis: gold evidence is on-topic (the interference case) but polarity-inverted (dismissed ≠ convicted). The policy emits a high-confidence `SUPPORTS` with a templated justification; per-row F1 credits this row as 0. ClaimVerif's upstream class skew (82.1% SUPPORTS) leaves a residual SUPPORTS basin that GRPO compresses but does not eliminate.

Figure 3: Case studies on the **ClaimVerif** claim verification task. Cases 1–3 are successes: a full proactive pipeline (think → search → answer), correct abstention on speculative content, and grounding in multi-anchor evidence. Case 4 (red border) is a failure: a polarity-inverting evidence span on which the model issues a high-confidence `SUPPORTS` (response-axis failure; trigger and retrieval are correct).

Case 5: DSTC9-T1 (Tourism Dialogue) — Correct Proactive Retrieval

Dialogue context: *User:* planning a trip to San Francisco, looking for a landmark in North Beach / Telegraph Hill. *System:* two options — Coit Tower or Pier 33.
Latest user turn: "how about coit tower is that location dog friendly"
Ground truth: Search needed **Score:** 5/5

search COIT TOWER how about coit tower is that location dog friendly
information Doc 1 (Coit Tower): *Are dogs allowed at Coit Tower? No.* | Doc 2: *Is Coit Tower wheelchair accessible? No.* | ...
answer No, dogs are not allowed at Coit Tower. Would you like me to book a ticket for you?

Case 6: DSTC9-T1 (Tourism Dialogue) — Correct Abstention

Latest user turn: "i was also looking for a place to stay in the same area, something like a regular two-star inn."
Ground truth: No search needed (booking request, not knowledge question) **Expected:** `<wait/>`

wait/ *Agent correctly identifies this as a booking/preference statement requiring no FAQ lookup.*

Case 7: DSTC9-T1 (Tourism Dialogue) — Pronoun Resolution in Search

Dialogue context: *User:* dine out in Castro, place I can book on Tuesday. *System:* It's called Kitchen Story. *User:* phone number? *System:* (415) 525...
Latest user turn: "Okay thanks. Does it have any seat available outside?"
Ground truth: Search needed **Score:** 5/5

search KITCHEN STORY Does it have any seat available outside?
information Doc 1 (Kitchen Story): *Are there outdoor seats at Kitchen Story? Yes.* | Doc 2 (Kitchen Istanbul): *outdoor seating available.* | ...
answer Yes, Kitchen Story offers outdoor seating. Anything else I can do for you?

Note: The agent resolves the pronoun "it" to "Kitchen Story" from dialogue context, constructing an entity-grounded query.

Case 8: DSTC9-T1 (Tourism Dialogue) — Spurious Trigger (Failure)

Latest user turn: "Can I get a confirmation number of the booking emailed to me?"
Ground truth: `<wait/>` (procedural; no FAQ lookup needed) **Outcome:** False trigger

think The user is asking about getting a confirmation number; let me check the FAQ.
search Can I get a confirmation number of the booking emailed to me?
information *(no FAQ match; the corpus contains no email-confirmation entry)*
answer You will receive an email confirmation once booking is complete.

Diagnosis: the policy ungrounded its own (correct) procedural reply with an unnecessary retrieval round (+0.6 s latency). The trigger-axis failure: per-row F1 credits this row as 0 *despite a serviceable surface answer.*

Figure 4: Case studies on **DSTC9-T1** (San Francisco tourism FAQ). Cases 5–7 are successes: a standard retrieval-grounded response, correct abstention on a booking turn, and deictic pronoun resolution ("it" → "Kitchen Story"). Case 8 (red border) is a failure: a spurious trigger on a procedural confirmation-number turn (false-positive trigger axis).

Case 9: DSTC11-T5 (Multi-domain Dialogue) — Correct Proactive Retrieval

Dialogue context: *User:* hotel, moderately priced, free parking. *System:* Acorn Guest House. *User:* address? *System:* 154 Chesterton Road, cb41da. *User:* cat-friendly? *System:* No pets allowed.
Latest user turn: "Is there a charge for WiFi?"
Ground truth: Search needed **Score:** 5/5

search ACORN GUEST HOUSE Is there a charge for WiFi?
information Doc 1 (Acorn Guest House): *The wifi is free.* | Doc 2: *Free WiFi.* | ...
answer According to the reviews I have on hand, the WiFi is free for guests to use during their stay.

Case 10: DSTC11-T5 (Multi-domain Dialogue) — Correct Abstention

Latest user turn: "any food will be just fine. Once you find the restaurant you want to book a table for 7 people at 13:30 on friday."
Ground truth: No search needed (booking action, not knowledge question) **Expected:** `<wait/>`

wait/ *Agent correctly identifies this as an API-action request (table booking) requiring no FAQ retrieval.*

Case 11: DSTC11-T5 (Multi-domain Dialogue) — Pronoun Resolution in Search

Dialogue context: *User:* four-star room, free wifi/parking, cheap. *System:* The Autumn House would be perfect — a 4-star guesthouse on the east end of town.
Latest user turn: "Are they located in a quiet area?"
Ground truth: Search needed **Score:** 5/5

search AUTUMN HOUSE Are they located in a quiet area?
information Doc 1 (Autumn House): *Outside sounds were extremely limited; very comfortable with the quiet atmosphere.* | Doc 2: *The room is big, nice and quiet.* | ...
answer According to the reviews, the Autumn House is in a quiet area; outside sounds are limited.

Note: The agent resolves "they" to "Autumn House" from context, constructing an entity-grounded query.

Case 12: DSTC11-T5 (Multi-domain Dialogue) — Polarity-Mismatched Retrieval (Failure)

Latest user turn: "Does it have a nice view?" *(context: BACKSTREET BISTRO)*
Gold-keyed review: *"...outdoor seating that overlooks a beautiful lake."*

search BACKSTREET BISTRO Does it have a nice view?
information Doc 1: *"We wished it had a better view."* | Doc 2: *"the view was just okay."* | Doc 4: *"The only downside ... was the poor view ... of an alleyway."* | ...
answer According to the reviews I have on hand, the view is not very good — described as an alleyway in the city.

Diagnosis: entity binding is correct but BGE surfaces five negative-polarity reviews; the positive-polarity gold review is absent from the top-K=5. The answer is faithfully grounded in the retrieved evidence yet polarity-inverted relative to the gold response (retrieval-axis failure; per-row F1 = 0). Conflicting user reviews for the same entity — the canonical DSTC11-T5 retrieval failure mode.

Figure 5: Case studies on **DSTC11-T5** (multi-domain task-oriented dialogue with subjective knowledge). Cases 9–11 are successes: standard retrieval, abstention on a booking-action turn, and pronoun resolution ("they" → "Autumn House"). Case 12 (red border) is a failure: a polarity-mismatched retrieval on a multi-review corpus where BGE surfaces only negative reviews of the same entity (retrieval-hit axis).