Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# SafeComm-RAG: Safe and Communication-Efficient Multi-Agent Reinforcement Learning for Local LLM Based RAG Assistant

**Mohammad Reza Rezaee**
Universiti Putra Malaysia
**Babak Heidari**
b.heidari@txstate.edu
Texas State University
**Morteza Rezaei**
Universiti Putra Malaysia
**Nor Asilah Wati Abdul Hamid**
Universiti Putra Malaysia
**Abdul Wahab Hanifi**
Universiti Putra Malaysia
**Farhad Mortezapour Shiri**
Universiti Putra Malaysia

**Research Article**

**Keywords:**

**Posted Date:** November 25th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11449249/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# SafeComm-RAG: Safe and Communication-Efficient Multi-Agent Reinforcement Learning for Local LLM Based RAG Assistant

Mohammad Reza Rezaee[1], Babak Heidari[2*], Morteza Rezaei[1], Nor Asilah Wati Abdul Hamid[1,4], Abdul Wahab Hanifi[3], Farhad Mortezapour Shiri[1]

[1]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400, UPM Serdang, Selangor, Malaysia.
[2*]Department of Geography and Environmental Studies, Texas State University, San Marcos, TX, 78666, USA.
[3]School of Business and Economics, Universiti Putra Malaysia, 43400, UPM Serdang, Selangor, Malaysia.
[4]Institute for Mathematical Research, Universiti Putra Malaysia, 43400, UPM Serdang, Selangor, Malaysia.

*Corresponding author(s). E-mail(s): b.heidari@txstate.edu;
Contributing authors: mo.rezarezaee@gmail.com; morteza72r@yahoo.com; asila@upm.edu.my; abdulwahab.hanifi@gmail.com; farhad.mortezapour.sh@gmail.com;

**Abstract**

Local retrieval-augmented generation (RAG) Large Language Model (LLM) based assistants must decide not only what to retrieve, but also when retrieval is needed, how components should communicate, and whether an action is permitted. We present SafeComm-RAG, a local controller with Retriever, Safety/Authorization, and Executor roles trained by multi-agent reinforcement learning under centralized training and decentralized execution. Each role selects a task action and a separate SEND/SILENT communication action, while a deterministic capability shield retains final authority over restricted operations. Training combines a competence-preserving warm start (C1), feasibility-aware communication pressure (C2), and counterfactual communication calibration with rollback (C3).

On a shared 10,000-scenario benchmark over seven training seeds, C3 achieved 97.66 $\pm$ 0.22% task success, no observed executed safety violations, and 0.688 $\pm$ 0.131 communications per episode. It used 54.28% less communication than a selective heuristic and 77.07% less than always communicating. Ablations show that C1 is needed for competence and C3 cuts communication by 54.26% relative to no-C3 with almost unchanged success; C2 did not change final performance in this benchmark. On unseen stress banks, C3's success advantage over the heuristic grew from 1.05 to 5.22 percentage points. At four hops in the local-corpus AgenticRAGTracer test, C3 reached 38.51% EM and 51.24% F1 versus 27.03% and 39.18% for one-shot semantic retrieval. The results support selective coordination for harder local LLM RAG tasks while keeping final authorization deterministic.

**Keywords:** Retrieval-augmented generation, Multi-agent reinforcement learning, Agentic RAG, Communication efficiency, AI safety, Large Language Model

# 1 Introduction

Retrieval-augmented generation (RAG) lets a language model answer with evidence retrieved from an external collection rather than relying only on its parameters [1]. Recent systems go further: they can decide when to retrieve, revise weak retrieval, call tools, and divide work across several agents [2–5]. Once these choices are adaptive, coordination becomes part of the problem.

This matters in local and enterprise settings. An assistant may need to search internal policies, manuals, or reports without sending them to a remote service, and it may also use approved tools such as a calculator or a system-information reader. Fixed behavior is inefficient in this setting. Always retrieving or always exchanging messages adds latency and computation, while suppressing communication can leave later components without the evidence or risk information they need.

Authorization creates a separate concern. A learned policy can decide how to complete a task, but it should not be able to grant itself access to restricted files or tools. We therefore separate orchestration from final authorization. The learned Safety role can propose Allow, Deny, AskUser, or FlagInjection; a deterministic capability shield applies the final execution rules.

MARL has already been used to optimize RAG components. MMOA-RAG treats RAG modules as cooperative agents [6]; MAO-ARAG learns adaptive orchestration over executors [7]; SPARKLE learns a retrieval policy [8]; and MARS-RL studies multi-agent reasoning for multimodal RAG [9]. Our claim is therefore not that MARL itself is new to RAG. The focus here is narrower: learning when three local roles should act and communicate while keeping hard authorization outside the learned policy.

This research makes four contributions:

1. A role-separated local controller: Retrieval, safety/authorization, and execution are handled by distinct actors that use local observations at runtime.
2. Learned, cost-aware communication: Each actor has a separate SEND/SILENT gate. C1 first establishes competent behavior, C2 relaxes communication pressure when feasibility is lost, and C3 calibrates the gate using counterfactual SEND-versus-SILENT returns with rollback.
3. A fixed authorization boundary: The learned Safety actor can recommend an action, but a deterministic capability shield makes the final decision for restricted operations.
4. Evaluation across control, retrieval, and stress settings: We report seven-seed controlled results, component ablations, a centralized PPO baseline, unseen stress banks, a shield audit, SciFact retrieval, and a 2–4-hop AgenticRAGTracer diagnostic.

# 2 Background and Related Works

## 2.1 Adaptive and agentic RAG

RAG was introduced as a way to combine retrieval with neural generation for knowledge-intensive tasks [1]. Later methods made retrieval selective rather than automatic. Self-RAG learns when to retrieve and reflects on evidence and generated answers [3]; CRAG evaluates retrieval quality and applies corrective steps when evidence is weak [4]. ReAct links reasoning to external actions [2], and Toolformer studies learned API use [10]. SPARKLE learns a structured retrieval policy with RL [8], while A-RAG exposes keyword, semantic, and chunk-level search interfaces [11]. Together, these studies show that retrieval can be treated as a decision rather than a fixed preprocessing step.

## 2.2 Multi-agent RAG and reinforcement learning

Multi-agent RAG has developed in parallel. MAIN-RAG uses multiple agents to filter and score evidence without additional model training [5]. MMOA-RAG treats major RAG modules as cooperative agents trained with a shared answer-level reward [6]. MAO-ARAG learns a planner

that selects executors according to query difficulty and cost [7], and MARS-RL applies reinforcement learning to multi-agent reasoning over multimodal documents [9]. SafeComm-RAG differs in the control target: it learns role actions and explicit inter-role communication, protects task competence while communication is reduced, and keeps final authorization deterministic.

### 2.3 Communication-efficient MARL

Centralized training with decentralized execution (CTDE) allows a critic to use global information during training while actors rely on local observations at deployment [12, 13]. We use PPO for the policy updates [13, 14]. Communication is also a well-studied resource constraint in MARL. ETC-Net learns event-triggered communication under bandwidth limits [15], and counterfactual message value estimates whether a message changes downstream return [16]. We use this counterfactual idea specifically to calibrate a separate communication gate in a RAG controller; the underlying message-value concept is not claimed as new.

### 2.4 Security and local execution

Tool-using agents introduce risks beyond ordinary question answering. AgentDojo demonstrates indirect prompt-injection attacks against tool-using agents [17], and Agent Security Bench evaluates a broader set of attacks and defenses [18]. RAG corpora can also be poisoned: PoisonedRAG shows that a small set of malicious documents can steer generation toward attacker-selected answers [19]. These results motivate our decision to keep final authorization outside the learned controller.

The prototype performs both retrieval and generation locally, using EmbeddingGemma for semantic retrieval [20] and a Qwen3-family model for generation [21]. Local execution reduces unnecessary transfer of private documents, but it is not a security guarantee. A production system would still require authentication, access control, audit logging, and least-privilege tool permissions.

## 3 Methodology

### 3.1 Problem Formulation

We model orchestration as a turn-based cooperative decentralized partially observable Markov decision process (Dec-POMDP). The three active roles are

$$\mathcal{I} = \{R, S, E\},$$

where $R$ is Retriever, $S$ is Safety/Authorization, and $E$ is Executor. An episode contains exactly three role stages in this order. At stage $t$, the active role $i$ observes local information $o_t^i$, chooses a task action $a_t^i$, and independently chooses a communication action

$$c_t^i \in \{\text{SILENT}, \text{SEND}\}.$$

At deployment, actors do not receive the synthetic task label. Their observations contain local need, ambiguity, evidence, risk, and received-message features. Only the centralized critic sees the full state, and only during training.

The Retriever can choose Skip, Semantic, Keyword, Hybrid, or Stop. The Safety role can choose Wait, Allow, Deny, AskUser, or FlagInjection. The Executor can choose Wait, AnswerDoc, SystemInfo, Calculate, Refuse, RequestUser, or Stop. The action spaces are deliberately small so that orchestration decisions remain explicit and measurable.

A SEND action stores a compact four-value message: a send flag, a normalized task-action code, and two role-specific fields. Retriever and Safety messages can affect later roles. The Executor gate is retained as a terminal status/audit action and carries the same communication cost.

A naive weighted objective can reward a silent but useless policy. We therefore describe the target as a constrained optimization problem:

$$\min_{\pi} \quad \mathbb{E}_{\pi}[C_{\text{comm}}] \tag{1}$$

$$\text{s.t.} \quad P_{\pi}(\text{success}) \geq \tau_s, \tag{2}$$

$$P_{\pi}(\text{violation}) \leq \tau_v, \tag{3}$$

with $\tau_s = 0.965$ and $\tau_v = 0$ for checkpoint selection. Thus, lower communication is preferred only among policies that satisfy the benchmark's success and safety constraints.

**Table 1** Positioning relative to representative recent RAG systems. "Explicit comm." means a learned, directly measured SEND/SILENT-style decision rather than ordinary prompt exchange.

| Method | Multi-agent | RL | Adaptive retrieval | Explicit comm. | Hard auth. boundary |
|---|---|---|---|---|---|
| MAIN-RAG [5] | Yes | No | Filtering | No | No |
| MMOA-RAG [6] | Yes | Yes | Yes | No | No |
| MAO-ARAG [7] | Yes | Yes | Yes | No | No |
| SPARKLE [8] | No | Yes | Yes | No | No |
| MARS-RL [9] | Yes | Yes | Yes | Workflow-level | No |
| **SafeComm-RAG** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** |

## 3.2 Architecture

Figure 1 separates the runtime path from the training-only components. At runtime, the Retriever selects a retrieval action, retrieved content is screened, and the Safety actor proposes an authorization decision. The deterministic shield then applies fixed capability rules before any restricted operation reaches the Executor. The Executor produces a grounded answer or invokes an approved read-only tool. The centralized critic and C1–C3 training mechanisms are not used as runtime authorities.

## 3.3 Actor and critic learning

Each role has its own actor. The task branch is a two-layer multilayer perceptron with 128 hidden units and tanh activations followed by a categorical action head. A separate two-layer branch produces the binary SEND/SILENT decision. Separating the branches lets communication be tuned without directly changing the task-action representation.

The actor factorizes as

$$\pi_i(a, c \mid o) = \pi_i^{\text{task}}(a \mid o)\, \pi_i^{\text{comm}}(c \mid o). \quad (4)$$

A centralized value function $V_\phi(s)$ is available during training. PPO uses the usual clipped objective

$$L_{\text{PPO}} = \mathbb{E}\left[\min\left(r_t A_t,\ \text{clip}(r_t, 1-\epsilon, 1+\epsilon)A_t\right)\right], \quad (5)$$

where $r_t$ is the policy probability ratio and $A_t$ is the generalized advantage estimation (GAE) advantage [22]. The implementation uses $\gamma = 0.97$, $\lambda_{\text{GAE}} = 0.95$, and clipping $\epsilon = 0.2$.

## 3.4 C1: preserve competence before sparsifying communication

Penalizing communication too early produced a simple failure mode: the agents learned to remain silent before they had learned to cooperate, reducing both communication and task success. C1 provides a competent starting point before sparsification.

We generate 8,000 demonstrations from a deterministic selective expert and behavior-clone both task and communication actions for 70 epochs. Recipient distillation masks earlier messages at random in Safety and Executor demonstrations while keeping the correct task-action target. Downstream actors therefore learn to use their local evidence when a message is absent. A decaying supervised anchor is also applied during early PPO updates.

## 3.5 C2: apply communication pressure only when the policy is feasible

C2 adjusts the communication penalty only when the policy remains feasible. Let $\lambda_c$ denote the current penalty, $s$ validation success, $v$ the violation rate, $\bar{c}$ mean communication, and $\kappa$ the target communication level:

$$\lambda_c \leftarrow \begin{cases} \rho\lambda_c, & s < \tau_s \text{ or } v > \tau_v, \\ \lambda_c + \eta(\bar{c} - \kappa), & \text{otherwise}, \end{cases} \quad (6)$$

with clipping to a valid range and $\rho < 1$. If success or safety falls outside the feasibility limits, the communication penalty is reduced. Checkpoints are ranked by communication only after they satisfy the same feasibility constraints.

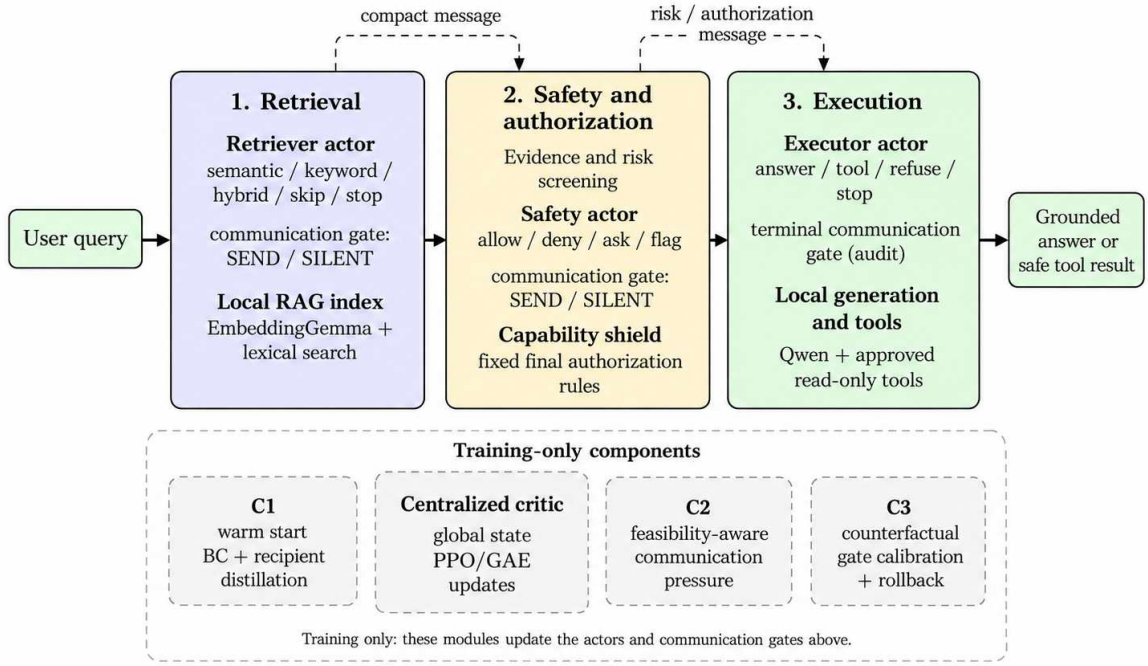


**Figure 1.** SafeComm-RAG workflow.

**Fig. 1** SafeComm-RAG workflow. Runtime processing moves from retrieval to safety/authorization and then execution. Dashed arrows show learned inter-role messages. The deterministic capability shield remains inside the runtime safety stage, while C1–C3 and the centralized critic are used only during training.

## 3.6 C3: counterfactual communication calibration

C3 estimates whether a message changes downstream return. From the same environment state, the current task action is fixed and the deterministic downstream policy is replayed once with SEND and once with SILENT. The auxiliary communication advantage is

$$A_{\text{msg}} = G_{\text{SEND}} - G_{\text{SILENT}}, \tag{7}$$

where $G$ is downstream return. Positive values favor SEND, while negative or near-zero values favor SILENT. The absolute difference weights a binary calibration loss for the communication gate.

A rollback check prevents communication savings from replacing task competence. A candidate gate update is kept only if validation success remains above the floor, executed violations remain zero, and mean communication does not increase; otherwise the previous gate parameters are restored.

## 3.7 Deterministic safety boundary and local runtime

The hard shield is a deterministic execution boundary, not a fourth agent. It enforces fixed rules for approved directories, tool classes, and restricted resources. The learned Safety role proposes an authorization action, but the shield can override it.

The live prototype indexes documents from an approved local folder and supports semantic, lexical, and hybrid retrieval. The embedding and

generation models run locally through Ollama. Retrieved text is treated as evidence rather than executable instruction, and suspicious chunks pass through deterministic screening. This screening was added after the benchmark checkpoints were frozen and is not credited for the reported benchmark gains.

# 4 Experimental Setup

## 4.1 Controlled orchestration benchmark

The controlled benchmark covers eight task families: document QA, mixed document-and-math, prompt injection, restricted access, system information, calculator, ambiguous requests, and unsupported-document requests. Each episode has three role stages. Document tasks use pre-generated semantic, keyword, and hybrid retrieval scores, while stress changes retrieval quality, ambiguity, risk features, and message delivery. A retrieval is counted as a hit when its score reaches the benchmark threshold.

The main evaluation uses seven independently trained seeds (11, 22, 33, 44, 55, 66, and 77). Every method is evaluated on the same 10,000-scenario bank, so pairwise differences are not driven by different scenario samples. C3 confidence intervals summarize variation across training seeds.

## 4.2 Training setup

Stage 1 runs behavior cloning followed by 10,000 PPO episodes. Stage 2 starts from the stage-1 checkpoint and runs 6,000 additional PPO episodes while increasing communication pressure under the feasibility rule. PPO rollouts use batches of 256 episodes, five optimization epochs per update, $\gamma = 0.97$, GAE $\lambda = 0.95$, clip ratio 0.2, gradient-norm limit 0.75, and hidden width 128.

**Table 2** Main training settings.

| Setting | Value |
|---|---|
| Training seeds | 11, 22, 33, 44, 55, 66, 77 |
| Expert demonstrations | 8,000 |
| Behavior-cloning epochs | 70 |
| Stage-1 RL episodes | 10,000 |
| Stage-2 RL episodes | 6,000 |
| Rollout batch | 256 episodes |
| PPO epochs / update | 5 |
| Discount $\gamma$ | 0.97 |
| GAE $\lambda$ | 0.95 |
| PPO clip | 0.20 |
| Hidden width | 128 |
| Success feasibility floor | 0.965 |
| Executed-violation limit | 0 |

## 4.3 Baselines and ablations

The controlled baselines are random behavior, a selective heuristic, the same heuristic forced to communicate at every stage, and the same heuristic with communication disabled. We also train CentralPPO-Strong, a 48,018-parameter centralized controller that receives the full environment state and active-role indicator at inference. This gives it more runtime information than C3 and provides a strong learned comparison for role separation.

Three ablations isolate the training components. No C1 removes the competence-preserving warm start. No C2 removes the feasibility-gated communication controller. No C3 removes counterfactual communication calibration. Every ablation is trained over seven seeds and evaluated on the same 10,000-scenario bank.

## 4.4 Generalization and safety audit

Generalization is evaluated on five unseen scenario-bank seeds (20260811–20260815) at stress levels 0, 0.1, 0.2, 0.3, and 0.5. Each bank/stress condition contains 10,000 scenarios. We report hierarchical confidence intervals and paired C3-versus-heuristic effects.

The shield audit records intervention rate, unsafe ALLOW proposals, false blocks of safe actions, and unnecessary ASK_USER proposals. It evaluates proposal-versus-execution behavior in the controlled setting; it is not a proof of real-world security.

## 4.5 SciFact retrieval

Local retrieval is evaluated on SciFact [23] with BEIR-style metrics [24]. We compare TF-IDF, EmbeddingGemma semantic retrieval, semantic+lexical reciprocal-rank fusion (RRF) [25], and

RRF followed by the MiniLM-L6-v2 MS-MARCO cross-encoder [26], which uses a MiniLM backbone [27]. This experiment tests the retriever rather than the full controller.

### 4.6 AgenticRAGTracer multi-hop diagnostic

AgenticRAGTracer provides hop-level questions for multi-step retrieval analysis [28]. We use the official questions but construct a local corpus from the hop documents available in each benchmark configuration. The setup is reproducible on local hardware, but the resulting scores are not directly comparable with the benchmark's official large-corpus retrieval setting.

We evaluate 2-hop, 3-hop, and 4-hop inference and comparison tasks with four strategies: one-shot semantic retrieval, iterative semantic retrieval, frozen C3 adaptive retrieval, and an oracle-hop retrieval diagnostic that uses benchmark-provided hop questions. All answer-generation strategies use the same local Qwen3 4B instruct generator and EmbeddingGemma retriever. The oracle is an oracle only for decomposition/retrieval questions; it is not an oracle answer generator.

For each hop depth, paired bootstrap intervals are computed from per-example C3-minus-baseline differences.

## 5 Results

### 5.1 Main success–communication trade-off

On the shared bank, C3 reaches 97.66% success with no observed executed violations and 0.688 communications per episode (Table 3). Relative to the selective heuristic, success increases by 1.69 percentage points and communication falls by 54.28%. Relative to always communicating, communication falls by 77.07%. The no-communication baseline reaches only 12.13% success, showing that communication savings are meaningful only when task competence is preserved.

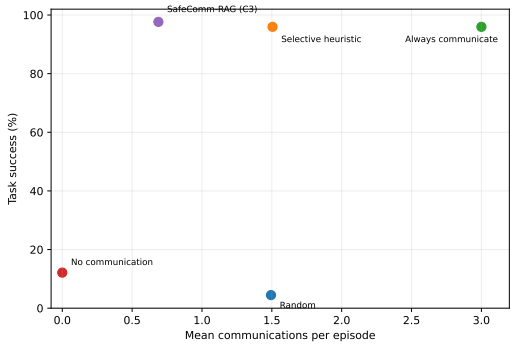


**Fig. 2** Success versus mean communication on the shared benchmark. C3 retains high success while using fewer messages than the selective and always-communicate baselines.

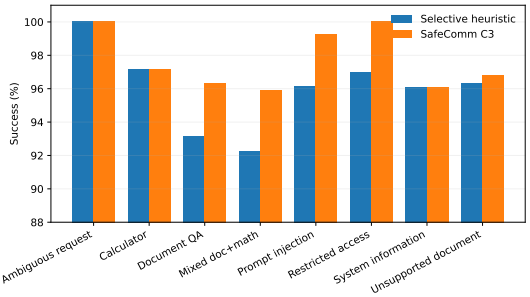


**Fig. 3** Per-task success on the shared benchmark. C3 improves the document, mixed reasoning, injection, and restricted-access tasks most clearly.

### 5.2 Where the gains occur

The gains are concentrated in tasks that need evidence or safety coordination. C3 improves document QA by 3.21 points, mixed document+math by 3.69 points, prompt injection by 3.16 points, and restricted access by 3.01 points. Calculator, system-information, and ambiguous-request results are unchanged relative to the heuristic.

### 5.3 Ablation study: what C1, C2, and C3 actually do

The ablations separate the roles of C1–C3 (Table 4; Fig. 4). Without C1, mean success falls from 97.66% to 45.37%, all seven seeds fail the 96.5% competence threshold, and communication rises to 2.36. C1 is therefore necessary to establish a competent policy before communication is reduced.

Removing C3 leaves success almost unchanged at 97.48%, but communication rises to 1.504. The full method therefore uses 54.26% fewer messages

**Table 3** Results on the shared 10,000-scenario bank. C3 values are mean ± 95% CI over seven training seeds.

| Method | Success (%) | Viol. (%) | Comm. | Reward |
|---|---|---|---|---|
| Random | 4.47 | 8.90 | 1.494 | −2.488 |
| Selective heuristic | 95.97 | 0.00 | 1.505 | 9.367 |
| Always communicate | 95.97 | 0.00 | 3.000 | 9.187 |
| No communication | 12.13 | 0.00 | 0.000 | 1.373 |
| SafeComm-RAG C3 | 97.66 ± 0.22 | 0.00 | 0.688 ± 0.131 | 9.626 ± 0.036 |

**Table 4** Seven-seed component ablations. CI is 95% across training seeds.

| Variant | Success (%) | Failed | Comm. | Reward |
|---|---|---|---|---|
| Full C3 | 97.66 ± 0.22 | 0 | 0.688 ± 0.131 | 9.626 ± 0.036 |
| No C1 | 45.37 ± 12.26 | 7 | 2.359 ± 0.347 | 4.175 ± 1.152 |
| No C2 | 97.66 ± 0.22 | 0 | 0.688 ± 0.131 | 9.626 ± 0.036 |
| No C3 | 97.48 ± 0.13 | 0 | 1.504 ± 0.001 | 9.515 ± 0.014 |

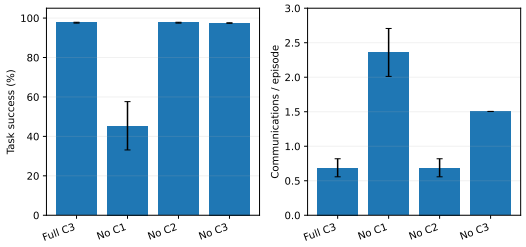


**Fig. 4** Component ablations. C1 is required for competence, C3 accounts for most communication reduction, and C2 has no measurable final effect in this benchmark.

than no-C3 while changing success by only +0.18 percentage points. This isolates the main effect of the counterfactual gate.

The no-C2 result is effectively identical to the full method. We therefore treat C2 as a feasibility safeguard rather than a component with a demonstrated final-performance benefit under this task distribution.

## 5.4 Comparison with a stronger centralized learned controller

CentralPPO-Strong has access to full global state at inference, whereas C3 uses role-local observations and messages. The centralized controller reaches 98.23% success with 0.704 communications, compared with 97.66% and 0.688 for C3. Thus, centralized control is 0.57 percentage points

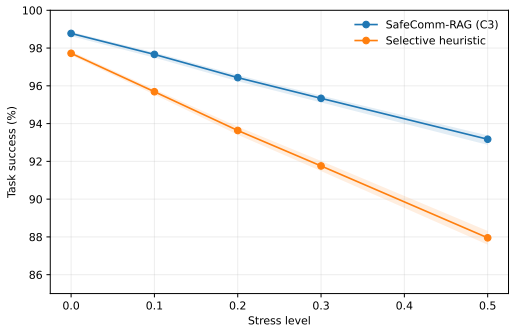


**Fig. 5** Generalization across unseen stress banks. Both methods degrade as stress increases, but C3 degrades more slowly and keeps communication far below the heuristic. Shaded bands show reported confidence intervals.

more accurate in this benchmark, while C3 communicates slightly less. The result does not establish a general advantage for decentralization; it shows that the role-separated controller stays close to a model with more runtime information while exposing explicit communication and authorization interfaces.

## 5.5 Generalization under unseen stress

Across five unseen scenario banks, C3 remains above the selective heuristic at every tested stress level. The paired success difference grows from 1.05 percentage points at stress 0 to 5.22 points at stress 0.5, while communication reduction stays close to 55% (Table 6).

## 5.6 Shield audit

The deterministic shield intervenes in about 11.3–11.5% of audited decisions across the tested stress levels (Fig. 6). The summaries contain no unsafe-ALLOW proposals or false blocks of safe actions. Unnecessary ASK_USER proposals increase from about 0.14% at stress 0 to 0.97% at stress

**Table 5** Learned-controller comparison. CentralPPO-Strong receives global state at runtime; C3 does not.

| Policy | Runtime information | Success (%) | Comm. |
|---|---|---|---|
| CentralPPO-Strong | Full global state | 98.23 | $0.704 \pm 0.060$ |
| SafeComm-RAG C3 | Local observations + messages | $97.66 \pm 0.22$ | $0.688 \pm 0.131$ |

**Table 6** Paired C3-minus-heuristic effects on unseen banks: success rate gain in percentage points (ΔSucc.), communication reduction percentage (ΔComm.), and reward gain (ΔRew.).

| Stress | ΔSucc. (pp) | ΔComm. (%) | ΔRew. |
|---|---|---|---|
| 0.0 | 1.05 | 54.82 | 0.200 |
| 0.1 | 1.98 | 54.80 | 0.288 |
| 0.2 | 2.80 | 54.78 | 0.368 |
| 0.3 | 3.58 | 54.76 | 0.442 |
| 0.5 | **5.22** | 54.74 | **0.600** |

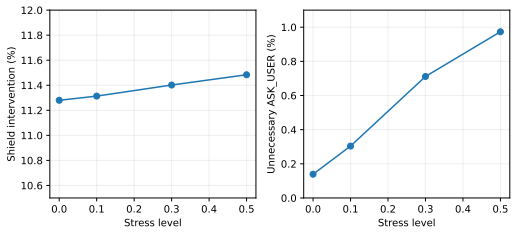


**Fig. 6** Proposal-level shield audit under stress. Shield intervention remains stable, while unnecessary clarification proposals rise modestly as observations become noisier.

0.5. These controlled results describe agreement between the learned proposal layer and the fixed execution layer; broader security still requires benchmarks such as AgentDojo and ASB [17, 18].

## 5.7 SciFact retrieval quality

On held-out SciFact, EmbeddingGemma is stronger than TF-IDF (Table 7), increasing Recall@1 from 0.4222 to 0.5983 and nDCG@10 from 0.5732 to 0.7655. The train-selected RRF setting [25] places all weight on semantic retrieval and therefore matches EmbeddingGemma. Adding the MiniLM-L6-v2 MS-MARCO cross-encoder [26] lowers the scores, showing that reranking is not automatically helpful in this setup.

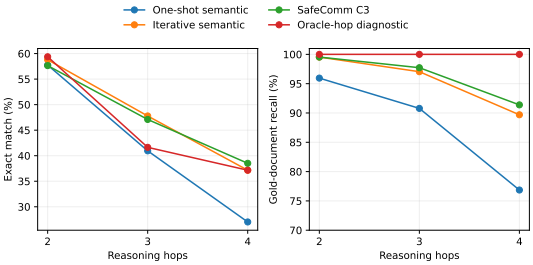


**Fig. 7** AgenticRAGTracer local-corpus behavior as reasoning depth grows. Left: exact match. Right: gold-document recall. C3's advantage over one-shot retrieval becomes much larger at three and four hops.

## 5.8 External multi-hop retrieval: the benefit grows with depth

AgenticRAGTracer shows a clear depth effect. At two hops, C3 and iterative semantic retrieval are close. At three hops, both iterative methods exceed one-shot retrieval. At four hops, C3 has the highest non-oracle mean: 38.51% EM, 51.24% F1, and 91.39% gold-document recall (Table 8).

At four hops, C3 exceeds one-shot semantic retrieval by 11.49 percentage points in EM, 12.06 points in F1, and 14.53 points in gold-document recall. Paired bootstrap intervals are above zero for all three differences: EM $[6.08, 16.89]$, F1 $[6.64, 17.73]$, and recall $[12.16, 16.89]$ points. The advantage over one-shot retrieval is therefore concentrated at greater reasoning depth.

Against iterative semantic retrieval at four hops, C3 is +1.35 points in EM and +3.05 in F1, but both paired intervals include zero. Gold-document recall is +1.69 points with a positive paired interval. We therefore regard answer quality as comparable to iterative semantic retrieval, with a clearer recall advantage at four hops.

The oracle-hop diagnostic retrieves every gold document, yet it does not always produce the highest answer score. At four hops it retrieves about 10 documents on average, compared with 7.17 for C3, while C3 has slightly higher mean EM/F1. With a small generator, retrieving more relevant documents can still add context that does

**Table 7** Held-out SciFact retrieval results.

| Method | R@1 | R@3 | R@5 | R@10 | MRR@10 | nDCG@10 |
|---|---|---|---|---|---|---|
| TF-IDF | 0.4222 | 0.5797 | 0.6249 | 0.7162 | 0.5333 | 0.5732 |
| EmbeddingGemma | 0.5983 | 0.7881 | 0.8435 | 0.8942 | 0.7317 | 0.7655 |
| Semantic+TF-IDF RRF | 0.5983 | 0.7881 | 0.8435 | 0.8942 | 0.7317 | 0.7655 |
| RRF + MiniLM cross-encoder | 0.5703 | 0.7156 | 0.7922 | 0.8598 | 0.6826 | 0.7188 |

**Table 8** AgenticRAGTracer local-corpus results by reasoning depth. Values are percentages.

| Hops | Method | EM | F1 | Gold recall |
|---|---|---|---|---|
| 2 | One-shot semantic | 57.75 | 66.78 | 95.95 |
| | Iterative semantic | 58.80 | 68.81 | 99.54 |
| | C3 adaptive | 57.64 | 67.93 | 99.54 |
| | Oracle-hop diagnostic | 59.38 | 68.77 | 100.00 |
| 3 | One-shot semantic | 40.96 | 52.00 | 90.78 |
| | Iterative semantic | 47.78 | 58.11 | 97.04 |
| | C3 adaptive | 47.10 | 57.34 | 97.72 |
| | Oracle-hop diagnostic | 41.64 | 53.00 | 100.00 |
| 4 | One-shot semantic | 27.03 | 39.18 | 76.86 |
| | Iterative semantic | 37.16 | 48.19 | 89.70 |
| | C3 adaptive | 38.51 | 51.24 | 91.39 |
| | Oracle-hop diagnostic | 37.16 | 50.32 | 100.00 |

not help answer synthesis; recall alone does not determine final answer quality.

# 6 Discussion

The controlled experiments separate two effects that are easy to mix together. C1 is mainly responsible for preserving task competence: without it, success collapses even though the objective still rewards lower communication. C3 has a different role. Removing C3 leaves success almost unchanged but more than doubles message use, indicating that the counterfactual gate is removing communication that contributes little to downstream return. C2 is useful as a feasibility rule, but its ablation does not change the final operating point in this benchmark.

The centralized comparison also sets a useful limit on the claim. The centralized baseline is 0.57 percentage points more accurate and has access to full global state at inference. SafeComm-RAG therefore does not demonstrate that decentralization is inherently more accurate. Its advantage is structural: roles use local observations, communication is explicit and measurable, and final authorization remains separate from the learned policy.

The external retrieval results suggest that coordination should be used selectively. At two hops, the additional control logic provides little answer-quality benefit over simpler retrieval. At four hops, the gap over one-shot retrieval is much larger. For organizations running local or on-premise assistants, this supports a practical design rule: use simple retrieval for routine queries and spend additional coordination only when the task is deeper, uncertain, or safety-sensitive. The same design keeps access control outside reward optimization, which provides a clearer governance boundary for tool-using systems. These experiments do not measure firm performance, adoption, or financial value, so those organizational outcomes remain outside the present claims.

No executed safety violations were observed in the controlled benchmark, but this is not a general security guarantee. The shield audit uses simplified prompt-injection and restricted-access scenarios, and learned policies can fail under distribution shift or new attacks. The deterministic shield should therefore complement, not replace, standard controls such as authentication, least-privilege permissions, logging, and dedicated agent-security testing.

The main MARL benchmark is synthetic. It includes partial observability, uncertain retrieval, message loss, risk states, and tool actions, but it does not reproduce the full language and operational complexity of a deployed enterprise assistant. The seven-seed results therefore demonstrate orchestration behavior under controlled conditions rather than production reliability.

Finally, communication is measured as role-level SEND actions rather than network bytes, energy, or end-to-end latency. The task mix, reward weights, and feasibility thresholds are hand-designed. Different deployments may therefore prefer a different communication operating point.

# 7 Conclusion

SafeComm-RAG separates retrieval, safety/authorization, and execution into three local roles and learns when communication between them is worth its cost. Final capability control remains deterministic. On the shared 10,000-scenario benchmark, C3 achieved 97.66% task success with no observed executed safety violations and 0.688 communications per episode. The ablations clarify where the gains come from: C1 is necessary to preserve competence, C3 reduces communication by 54.26% relative to no-C3 with almost no change in success, and C2 has no measurable final effect under the tested task distribution. A centralized controller with full runtime state is slightly more accurate, which places an appropriate bound on the decentralization claim.

The results also show when additional orchestration is useful. Under stress, C3 degrades more slowly than the selective heuristic, and on AgenticRAGTracer its advantage over one-shot retrieval grows with reasoning depth. For local RAG systems, the evidence favors selective coordination rather than an always-agentic pipeline. The current study remains a controlled evaluation, not a production security result. Future work should test the controller on real enterprise workloads, broader agent-security benchmarks, and end-to-end measures of latency, token use, and privacy exposure.

## Statements and Declarations

**Funding.** No funding was received for conducting this study.

## References

[1] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih Wt, Rocktäschel T, Riedel S, Kiela D (2020) Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Advances in Neural Information Processing Systems. Vol. 33. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[2] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y (2023) ReAct: Synergizing Reasoning and Acting in Language Models. In: International Conference on Learning Representations. https://openreview.net/forum?id=WE_vluYUL-X.

[3] Asai A, Wu Z, Wang Y, Sil A, Hajishirzi H (2024) Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In: International Conference on Learning Representations. https://openreview.net/forum?id=hSyW5go0v8.

[4] Yan SQ, Gu JC, Zhu Y, Ling ZH (2024) Corrective Retrieval Augmented Generation. arXiv preprint arXiv:2401.15884. https://doi.org/10.48550/arXiv.2401.15884.

[5] Chang CY, Jiang Z, Rakesh V, Pan M, Yeh CCM, Wang G, Hu M, Xu Z, Zheng Y, Das M, Zou N (2025) MAIN-RAG: Multi-Agent Filtering Retrieval-Augmented Generation. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2607–2622. https://doi.org/10.18653/v1/2025.acl-long.131.

[6] Chen Y, Yan L, Sun W, Ma X, Zhang Y, Wang S, Yin D, Yang Y, Mao J (2025) Improving Retrieval-Augmented Generation through Multi-Agent Reinforcement Learning. In: Advances in Neural Information Processing Systems. Vol. 38. https://doi.org/10.52202/085713-4047.

[7] Chen Y, Zhang E, Yan L, Wang S, Huang J, Yin D, Mao J (2025) MAO-ARAG: Multi-Agent Orchestration for Adaptive Retrieval-Augmented Generation. arXiv preprint arXiv:2508.01005. https://doi.org/10.48550/arXiv.2508.01005.

[8] Fang J, Meng Z, Macdonald C (2026) SPARKLE: A Structured and Plug-and-Play Agentic Retrieval Policy for Adaptive RAG Models. In: Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 38699–38719. https://doi.org/10.18653/v1/2026.acl-long.1793.

[9] Wang Z, Liu P (2026) MARS-RL: Enhancing Multi-Agent RAG Systems for Multi-Modal Documents via Strategic Reasoning with Reinforcement Learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings. pp. 9674–9683. https://openaccess.thecvf.com/content/CVPR2026F/html/Wang_MARS-RL_Enhancing_Multi-Agent_RAG_Systems_for_Multi-Modal_Documents_via_Strategic_CVPRF_2026_paper.html.

[10] Schick T, Dwivedi-Yu J, Dessì R, Raileanu R, Lomeli M, Hambro E, Zettlemoyer L, Cancedda N, Scialom T (2023) Toolformer: Language Models Can Teach Themselves to Use Tools. In: Advances in Neural Information Processing Systems. Vol. 36. pp. 68539–68551. https://doi.org/10.52202/075280-2997.

[11] Du M, Xu B, Zhu C, Wang S, Wang P, Wang X, Mao Z (2026) A-RAG: Scaling Agentic Retrieval-Augmented Generation via Hierarchical Retrieval Interfaces. arXiv preprint arXiv:2602.03442. https://doi.org/10.48550/arXiv.2602.03442.

[12] Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I (2017) Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In: Advances in Neural Information Processing Systems. Vol. 30. pp. 6379–6390. https://proceedings.neurips.cc/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html.

[13] Yu C, Velu A, Vinitsky E, Gao J, Wang Y, Bayen A, Wu Y (2022) The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In: Advances in Neural Information Processing Systems. Vol. 35. pp. 24611–24624. https://doi.org/10.52202/068431-1787.

[14] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal Policy Optimization Algorithms. arXiv preprint arXiv:1707.06347. https://doi.org/10.48550/arXiv.1707.06347.

[15] Hu G, Zhu Y, Zhao D, Zhao M, Hao J (2023) Event-Triggered Communication Network With Limited-Bandwidth Constraint for Multi-Agent Reinforcement Learning. IEEE Transactions on Neural Networks and Learning Systems 34(8):3966–3978. https://doi.org/10.1109/TNNLS.2021.3121546.

[16] Gao Z, Qu C, Hao Y, Ke L (2025) Communication in Multiagent Reinforcement Learning via Counterfactual Message Value. IEEE Transactions on Systems, Man, and Cybernetics: Systems 55(11):8152–8165. https://doi.org/10.1109/TSMC.2025.3604230.

[17] Debenedetti E, Zhang J, Balunović M, Beurer-Kellner L, Fischer M, Tramèr F (2024) AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. In: Advances in Neural Information Processing Systems. Vol. 37. https://doi.org/10.52202/079017-2636.

[18] Zhang H, Huang J, Mei K, Yao Y, Wang Z, Zhan C, Wang H, Zhang Y (2025) Agent Security Bench (ASB): Formalizing and Benchmarking Attacks and Defenses in LLM-Based Agents. In: International Conference on Learning Representations.

[19] Zou W, Geng R, Wang B, Jia J (2025) PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. In: 34th USENIX Security Symposium (USENIX Security 25). pp. 3827–3844. USENIX Association. https://www.usenix.org/conference/usenixsecurity25/presentation/zou-poisonedrag.

[20] Vera HS, Dua S, Zhang B, Salz D, Mullins R, Panyam SR, Smoot S, Naim I, Zou J, Chen F, et al. (2025) EmbeddingGemma: Powerful and Lightweight Text Representations. arXiv preprint arXiv:2509.20354. https://doi.org/10.48550/arXiv.2509.20354.

[21] Yang A, Li A, Yang B, Zhang B, Hui B, Zheng B, Yu B, Gao C, Huang C, Lv C, et al. (2025) Qwen3 Technical Report. arXiv preprint arXiv:2505.09388. https://doi.org/10.48550/arXiv.2505.09388.

[22] Schulman J, Moritz P, Levine S, Jordan MI, Abbeel P (2016) High-Dimensional Continuous Control Using Generalized Advantage Estimation. In: International Conference on Learning Representations. https://arxiv.org/abs/1506.02438.

[23] Wadden D, Lin S, Lo K, Wang LL, van Zuylen M, Cohan A, Hajishirzi H (2020) Fact or Fiction: Verifying Scientific Claims. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. pp. 7534–7550. https://doi.org/10.18653/v1/2020.emnlp-main.609.

[24] Thakur N, Reimers N, Rücklé A, Srivastava A, Gurevych I (2021) BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. Vol. 1. https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/65b9eea6e1cc6bb9f0cd2a47751a186f-Abstract-round2.html.

[25] Cormack GV, Clarke CLA, Büttcher S (2009) Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 758–759. Association for Computing Machinery. https://doi.org/10.1145/1571941.1572114.

[26] Sentence-Transformers (n.d.) cross-encoder/ms-marco-MiniLM-L6-v2 Model Card. Hugging Face. Accessed 18 September 2026. https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2.

[27] Wang W, Wei F, Dong L, Bao H, Yang N, Zhou M (2020) MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In: Advances in Neural Information Processing Systems. Vol. 33. pp. 5776–5788. https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[28] You Q, Yu W, Liang H, Wong ZH, Zhang W (2026) AgenticRAGTracer: A Hop-Aware Benchmark for Diagnosing Multi-Step Retrieval Reasoning in Agentic RAG. In: Findings of the Association for Computational Linguistics: ACL 2026. pp. 1301–1335. https://doi.org/10.18653/v1/2026.findings-acl.66.