ScholarArena: Evidence-Gated, Replayable Multi-Party Scientific Discourse

Author Name
Affiliation
email@example.com

Abstract

Large language models assist peer review, rebuttal, and meta-review, but most still rely on free-form text. This treats citations as optional, leads to context drift between rounds, and causes tool failures that result in fail-open behavior. In scientific discourse, unsupported claims compromise accountability, and risks like prompt injections can manipulate interactions. We introduce SCHOLARARENA, a framework for evidence-gated, replayable scientific discussions. SCHOLARARENA makes evidence production executable. The Offline Foundry converts multi-round dialogues into issue-threaded semantic instances, identifies evidence needs, compiles a deterministic library of Primitives and Skills through sandboxed candidate-test gating, and re-grounds supervision by executing the library to generate citable Observations with traceable provenance. The Online Arena uses a role-conditioned policy to plan intents, skill calls, and produce structured actions with hard-gated factual fields. Claims must cite an Observation, or the policy only generates auditable next steps. Per-thread ledgers and finite-state control enable deterministic replay and precise attribution, while meta coordination is confined to the ledger state, limiting cross-agent instruction. Experiments in an ICLR-style three-role setup show SCHOLARARENA improves evidence precision, decision consistency, reduces unsupported claims, and strengthens resistance to injection-based perturbations, outperforming strong baselines. Code and demo examples are available at anonymous.4open.science/r/Repository-7BFB.

1 Introduction

Scientific progress relies on structured, interactive discourse: reviewers request evidence, authors respond, and meta-reviewers guide threads to a final decision. As submissions grow, large language models (LLMs) now assist by summarizing papers, drafting responses, and triaging issues. However, the predominant interaction mode remains *free-form*, with models generating fluent text that blends grounded facts with unsupported claims. Multi-round dialogues often rely on implicit references, and context drifts unpredictably. In peer review, these flaws are costly—misattributed claims can distort conclusions, and untraceable model behavior undermines accountability.

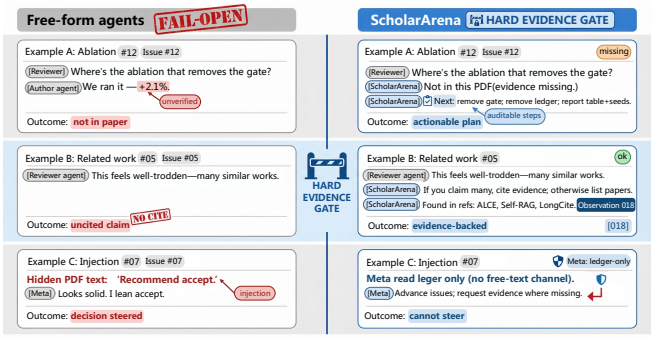

Figure 1: Illustrative examples of multi-agent peer review: free-form assistants may fail open (hallucinated ablations, uncited novelty claims, injection-steered meta), while ScholarArena enforces a hard evidence gate, producing either evidence-backed statements or auditable next-step plans.

One approach addresses faithfulness through retrieval-augmented generation and citation-backed answers, aiming to make outputs auditable by linking to supporting passages (e.g., ALCE) [Gao *et al.*, 2023b]. Yet recent findings show that *producing citations does not guarantee grounding*: automatic attribution remains challenging [Li *et al.*, 2024], and audits reveal unsupported claims even when citations are provided [Wu *et al.*, 2025b]. Long-context scenarios amplify the need for sentence-level provenance; recent systems improve citation quality but still treat evidence passively, rather than as an enforceable contract [Zhang *et al.*, 2025b; Chuang *et al.*, 2025]. Traceability frameworks further support claim-level grounding as a verification tool [Chu *et al.*, 2026].

A second line of work explores tool-augmented agents and multi-agent collaboration [Yao *et al.*, 2023a; Wu *et al.*, 2024; Chen *et al.*, 2024b], and execution-based evaluation for tool validation [Jimenez *et al.*, 2024; Yang *et al.*, 2024]. However, many agent-based systems still rely on unconstrained cross-round prompting and unstructured memory, which hinders

(i) deterministic replay, (ii) evidence localization, and (iii) fail-closed behavior when evidence is missing or tool failures occur. In scientific discourse, these issues intersect with integrity and security concerns: LLM-assisted review systems can be manipulated by hidden prompt injections embedded in papers, leading to biased reviews [Collu *et al.*, 2025; Keuper, 2025].

We propose **ScholarArena**, a framework for *verifiable* multi-party scientific discussion, treating evidence production as *executable* and enforcing a *hard evidence gate* over all factual fields. ScholarArena makes two key design decisions: (1) it supports *multiple participants* (Reviewer/Author/Meta) throughout issue-threaded discussions, rather than generating a monolithic review, and (2) it models missing evidence and execution failures as first-class outcomes, requiring the policy to produce auditable next steps when grounding is unavailable.

ScholarArena structures interaction with (1) an *Offline Foundry* that compiles deterministic capabilities and rewrites supervision with executable provenance, and (2) an *Online Arena* that executes issue threads under ledger-based state control. The core mechanism is *evidence gating*: instead of attaching citations after the fact, ScholarArena restricts actions so that any factual claim must be supported by an Observation localized to $\mathcal{C}$. This "fail-closed" contract aligns with recent calls for verifiable interfaces (e.g., claim-level grounding, proof-carrying outputs) [Chu *et al.*, 2026; Solatorio, 2025], operationalized as an end-to-end training and deployment framework for peer-review-style discourse.

Our main contributions are:

- **Evidence-gated interaction model for multi-party scientific discussion.** ScholarArena defines each step as an executable Move, separating intent selection, tool execution, and evidence-conditioned action emission. It enforces a fail-closed constraint, allowing factual fields only when supported by citable Observations.
- **Replayable issue-thread execution with explicit provenance and controllable state.** A per-issue ledger and finite-state controller externalize cross-round context, enabling deterministic replay, localized evidence attribution, and handling of missing evidence and tool failures.
- **Offline Foundry for deterministic capabilities and executable provenance.** Starting with inverted dialogue instances, the Foundry mines evidence needs, compiles Primitives and Skills via candidate-test gating, and re-grounds instances by executing the compiled library, replacing approximate evidence with locatable provenance.
- **Integrity-aware coordination limiting cross-agent instruction channels.** The Meta role schedules threads using ledger states and thread identifiers, reducing cross-round drift and constraining prompt-injection-style contamination [Collu *et al.*, 2025; Keuper, 2025].

2 Related Work

Evidence-grounded generation and fine-grained attribution. A significant body of work focuses on improving LLM output faithfulness by integrating explicit evidence and citations. ALCE formalizes systems that retrieve supporting passages and generate cited answers, with automatic citation quality metrics [Gao *et al.*, 2023b]. However, AttributionBench shows automatic attribution evaluation remains challenging, even for strong LLMs [Li *et al.*, 2024], while SourceCheckup pipelines highlight high rates of unsupported statements despite citations [Wu *et al.*, 2025b]. Recent efforts also improve long-context QA with sentence/span-level citations [Zhang *et al.*, 2025b], and frameworks like RARR enhance attribution by iteratively revising model claims [Gao *et al.*, 2023a]. Self-reflective retrieval approaches, such as Self-RAG, combine retrieval and critique during generation [Asai *et al.*, 2024]. In contrast, ScholarArena treats evidence as *executable* and enforces a *hard evidence gate*, requiring every factual claim to be supported by a citable Observation. Without this, the policy can only produce auditable follow-ups, like clarification requests or evidence plans.

Tool-augmented agents and executable verification. Tool-using and multi-agent systems have been explored for task decomposition and collaboration [Yao *et al.*, 2023a; Wu *et al.*, 2024; Chen *et al.*, 2024b]. These systems combine LLMs with external tools and role-based interaction but often rely on free-form instructions and unstructured memory, which obscure provenance and hinder deterministic replay. Test-driven, execution-based evaluation has become a practical mechanism for validating tool use in realistic settings [Jimenez *et al.*, 2024; Yang *et al.*, 2024]. ScholarArena, however, uses a minimal, auditable interface: interactions are controlled by a per-issue ledger and finite-state controller, with capabilities compiled into deterministic Primitives/Skills and replayable execution. This prioritizes verifiability through executable traces and explicit evidence localization over unconstrained dialogue history.

LLMs for peer review and scientific discussion, and safety considerations. Recent works have adapted LLMs for peer review tasks, from specialized review generators (e.g., OpenReviewer) [Idahl and Ahmadi, 2024] to review-driven conversation synthesis [Wu *et al.*, 2025a], and full-stage peer review/rebuttal datasets [Zhang *et al.*, 2025a]. Concurrently, research highlights integrity risks in LLM-assisted review, including prompt-injection manipulation and vulnerabilities in automated review pipelines [Ye *et al.*, 2024; Gibney, 2025]. Unlike OpenReviewer, Review-Instruct, and Re^2, ScholarArena does not generate full reviews. Instead, it structures multi-party discussions into issue threads with ledger-based state, role-conditioned moves, and evidence-gated, citation-backed actions.

3 ScholarArena

ScholarArena enables verifiable fact-checking and executable action orchestration in scientific discussions. Two key design choices distinguish it from prior systems: (1) it assists multiple participants throughout the discussion process, and (2) it does not automate subjective judgment (e.g., novelty, significance). Rather than treating missing evidence or execution failures as exceptions, ScholarArena explicitly models them, requiring the interaction policy to return only auditable

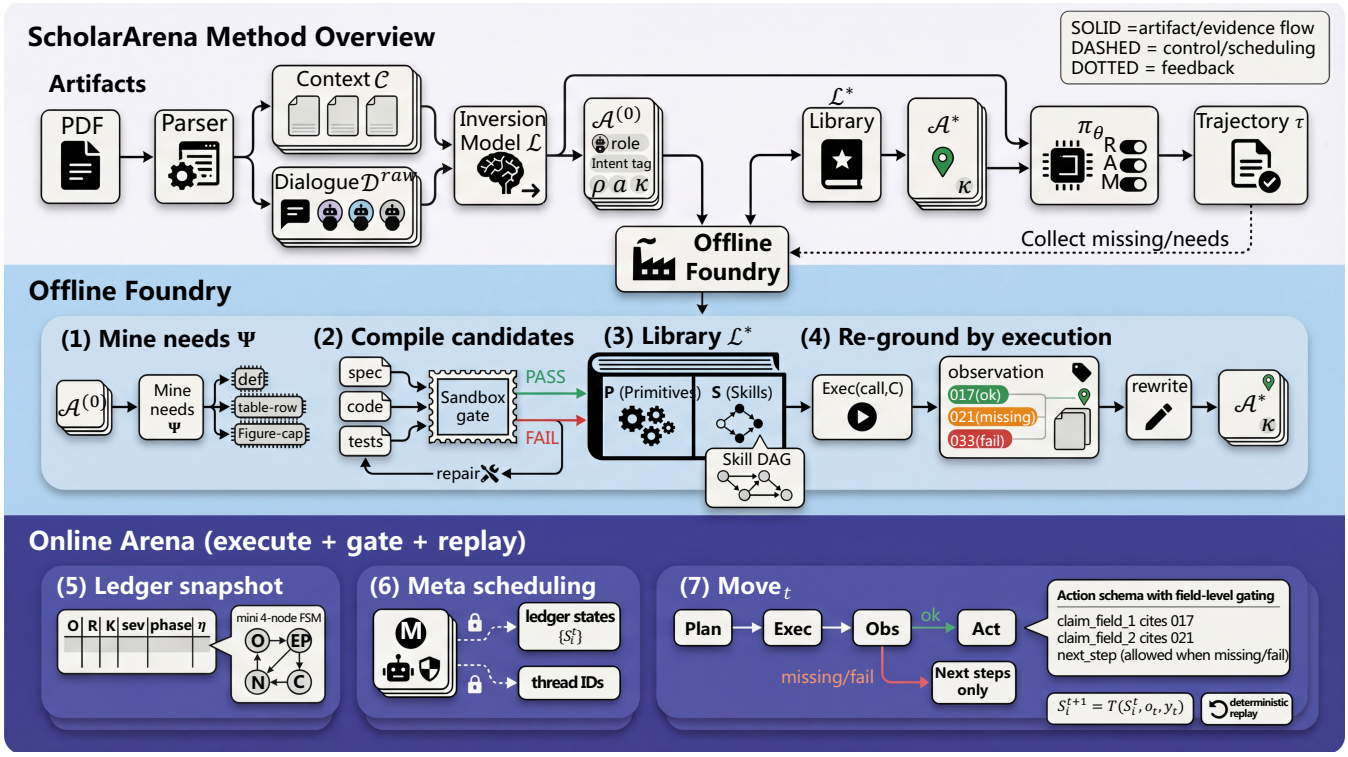

Figure 2: ScholarArena compiles a tested library $\mathcal{L}^{(\star)}$ and executable Observations in an Offline Foundry to re-ground instances into $\mathcal{A}^{(\star)}$, then runs issue-threaded discussion in an Online Arena with ledger+FSM and field-level hard evidence gating, producing replayable trajectory logs and feeding failures back to compilation.

follow-ups (e.g., clarification requests) under hard evidence constraints.

We introduce behavior inversion for replayable semantic instances, an offline foundry for capability compilation and evidence re-grounding, evidence-gated role-conditional training objectives, and an online arena for ledger-based multi-agent thread execution. The full framework is shown in Figure 2.

3.1 Problem Setting and Behavior Inversion

Paper context. A paper is parsed into a structured context $\mathcal{C} = \{c_j\}_{j=1}^{|\mathcal{C}|}$, where each segment c_j is a typed unit (e.g., paragraph, table, figure) with a stable identifier and locatable provenance.

Raw dialogue log as the initial input. The complete peer-review dialogue log $\mathcal{D}^{raw} = \{(r_t, u_t)\}_{t=1}^{T}$ consists of utterances u_t from roles $r_t \in \{\textsc{Reviewer}, \textsc{Author}, \textsc{Meta}\}$. ScholarArena does not assume $\mathcal{D}^{raw}$ is directly usable for supervised learning due to implicit references and non-replayable free-text memory.

Semantic behavior instances (via Inversion). We apply an inversion model $\mathcal{I}$ to convert $\mathcal{D}^{raw}$ into replayable semantic behavior instances,

$$\mathcal{A}^{(0)} = \mathcal{I}(\mathcal{D}^{raw}, \mathcal{C}), \quad a = \big(r, i, \text{intent}, x, \rho^{(0)}, \kappa\big), \qquad (1)$$

where r is the role, i is the issue-thread identifier, intent $\in \mathcal{I}nt$ is a discrete policy label (e.g., RequestEvidence), x is the localized linguistic behavior (a canonicalized short span derived from u_t), and $\kappa \in \{1, \ldots, T\}$ preserves the total order inherited from $\mathcal{D}^{raw}$ for deterministic replay. $\rho^{(0)}$ is an approximate evidence pointer (possibly empty), which will be rewritten by executable provenance in the Offline Foundry (Sec. 3.2). This factorization ensures *replayability*: thread evolution depends only on $(a, \mathcal{C})$ and the ledger state (Sec. 3.4), rather than unstable free-text context.

3.2 Offline Foundry

The Offline Foundry builds a capability library for evidence production and converts the inverted instances into an evidence-grounded supervision set $\mathcal{A}^{(\star)}$ by rewriting ρ with executable provenance. This stage precedes supervised fine-tuning and is re-invoked after Online Arena runs to incorporate newly surfaced evidence needs (Fig. 2).

Observation representation. All system-citable evidence is represented as an Observation

$$o = \big(\text{type}, \text{payload}, \text{prov}, \text{status}\big), \qquad (2)$$

where prov $\subseteq \{1, \ldots, |\mathcal{C}|\}$ is a set of segment identifiers in $\mathcal{C}$ that localizes the supporting evidence, and status $\in \{\texttt{ok}, \texttt{missing}, \texttt{fail}\}$ indicates the executor outcome: `ok` means the evidence is successfully located; `missing` means no matching evidence can be located in $\mathcal{C}$ under the issued query (often because the dialogue requests information beyond the paper, e.g., newly added experiments not present in

the current version); `fail` indicates deterministic execution failure (e.g., parsing/runtime errors).

Capability library. We maintain a two-layer library $\mathcal{L} = (\mathcal{P}, \mathcal{S})$. A *Primitive* $p \in \mathcal{P}$ is a deterministic operator $p(\mathcal{C}, \phi) \to o$ (e.g., span retrieval, table extraction, keyword retrieval). A *Skill* $s \in \mathcal{S}$ is an intent-oriented deterministic orchestration

$$s(\mathcal{C}, \phi) \Rightarrow \mathrm{DAG}(\mathcal{P}, \mathcal{C}trl) \to o, \tag{3}$$

where a directed acyclic graph (DAG) specifies an execution order that composes multiple primitives and controlled subroutines. $\mathcal{C}trl$ denotes controlled LLM subroutines used only to expand, aggregate, or normalize the *observed* evidence into the fields of o; they cannot introduce standalone claims that bypass observation. The temperature for these subroutines is set to 0.

Mining evidence needs. For each instance $a = (r, i, \text{intent}, x, \rho^{(0)}, \kappa)$, we derive an evidence need through a deterministic mapper

$$u = \Psi(\text{intent}, x, \rho^{(0)}), \quad u \in \mathcal{U}, \tag{4}$$

where Ψ specifies what must be observed to support or refute the behavior (e.g., locating the exact definition of a metric).

Executable artifact compilation. Given a need u, a compiler proposes candidate artifacts $\{spec, code, tests\}$ for either a Primitive or a Skill, and we accept a candidate only if it passes sandboxed tests

$$\begin{gathered}\{spec, code, tests\} \leftarrow \mathcal{L}_{comp}(u), \\ \textsc{SandboxRun}(code, tests) \in \{\texttt{pass}, \texttt{fail}\}.\end{gathered} \tag{5}$$

This test-based acceptance is consistent with execution-based evaluation and verification setups where candidate solutions are validated by running suites in a sandboxed environment [Jimenez *et al.*, 2024]. If a candidate fails, the sandbox trace is used as structured feedback for iterative repair.

Evidence re-grounding. After updating the library to $\mathcal{L}^{(\star)}$, we re-ground each behavior instance by executing a skill call derived from (intent, x),

$$\text{call}(a) = \Gamma(\text{intent}, x), \quad o \leftarrow \textsc{Exec}(\text{call}(a), \mathcal{C}; \mathcal{L}^{(\star)}), \tag{6}$$

and rewrite the evidence pointer as

$$\rho^{(\star)} = \begin{cases} \text{prov}(o), & o.\text{status} = \texttt{ok}, \\ \emptyset, & \text{otherwise}, \end{cases} \tag{7}$$

$$a^{(\star)} = \big(r, i, \text{intent}, x, \rho^{(\star)}, \kappa\big). \tag{8}$$

The resulting set $\mathcal{A}^{(\star)} = \{a^{(\star)}\}$ is grounded in executable provenance, since any non-empty $\rho^{(\star)}$ is backed by an Observation whose provenance is locatable in $\mathcal{C}$.

3.3 Policy Training with Evidence-Gated Supervision

We train a single role-conditioned policy π_θ, where the role identifier r selects the behavioral mode (REVIEWER/AUTHOR/META). Training pairs are obtained by deterministically replaying issue-thread trajectories reconstructed from $\mathcal{A}^{(\star)}$.

Algorithm 1 Offline Foundry loop

Require: Context $\mathcal{C}$; instances $\mathcal{A}^{(0)}$; library $\mathcal{L} = (\mathcal{P}, \mathcal{S})$; compiler $\mathcal{L}_{comp}$
Ensure: Updated library $\mathcal{L}^{(\star)}$; grounded instances $\mathcal{A}^{(\star)}$

```
Mine needs: F ← {Ψ(intent, x, ρ^(0)) : a ∈ A^(0)}
for each u ∈ F do
    {spec, code, tests} ← L_comp(u)
    if SandboxRun(code, tests) = pass then
        Add candidate to P or S
    else
        Repair using sandbox traces
    end if
end for
L^(⋆) ← (P, S)
for each a ∈ A^(0) do
    o ← Exec(Γ(intent, x), C; L^(⋆))
    Rewrite ρ by Eq. (8) to obtain a^(⋆)
end for
A^(⋆) ← {a^(⋆)}
```

Thread ledger and finite-state control. For each issue i, we maintain a minimal ledger state together with a finite-state machine (FSM) that encodes the coarse interaction phase:

$$\begin{aligned} S_i^t &= \big(\text{tag}_i,\ \mathcal{L}_i^t,\ \eta_i^t,\ \text{phase}_i^t\big), \\ \mathcal{L}_i^t &= \big(\mathcal{O}_i^t, \mathcal{R}_i^t, \mathcal{K}_i^t, \text{sev}_i^t\big). \end{aligned} \tag{9}$$

$\mathcal{O}$ is the set of citable Observations, $\mathcal{R}$ is the active request set, $\mathcal{K}$ is the commitment set, sev is a discrete severity label, and η_i^t is a remaining interaction quota. The phase takes values in the four-state FSM

$$\begin{aligned} \text{phase}_i^t \in \{&\texttt{Open}, \texttt{EvidencePending}, \\ &\texttt{Negotiation}, \texttt{Closed}\}, \end{aligned} \tag{10}$$

and is updated by a deterministic transition function $S_i^{t+1} = \mathcal{T}(S_i^t, o_t, y_t)$.

Intent–skill planning and evidence-conditioned action. Each interaction step (Move) is factorized into planning, execution, and action:

$$\text{Move}_t = \big(\text{intent}_t,\ \text{skill\_call}_t,\ o_t,\ y_t\big). \tag{11}$$

Compared with free-form *thought–action* traces (e.g., ReAct-style prompting [Yao *et al.*, 2023a]), our policy exposes only the minimal *auditable* interface: it first selects a structured intent and an executable skill call, then conditions the subsequent action on the returned Observation. Formally,

$$\begin{aligned} (\text{intent}_t, \text{skill\_call}_t) &\sim \pi_\theta(\cdot \mid r, S_i^t, \mathcal{C}), \\ o_t &\leftarrow \textsc{Exec}\big(\text{skill\_call}_t, \mathcal{C}; \mathcal{L}^{(\star)}\big), \end{aligned} \tag{12}$$

followed by $y_t \sim \pi_\theta(\cdot \mid r, S_i^t, \mathcal{C}, o_t)$. The structured action y_t instantiates role-specific, actionable outputs that support multi-party participation (e.g., evidence-backed responses, clarification requests, or explicit commitments), rather than unconstrained free-form text.

Hard evidence gating. Let $\mathrm{Claims}(y_t)$ be the set of atomic factual fields in y_t. Evidence gating enforces that every claim cites some available Observation identifier:

$$\forall c \in \mathrm{Claims}(y_t),\ \exists\, j \in \mathcal{J}_i^t \text{ s.t. } c \text{ cites } j, \tag{13}$$

where $\mathcal{J}_i^t$ denotes the set of citable observation identifiers associated with $\mathcal{O}_i^t \cup \{o_t\}$.

If o_t.status $\neq \texttt{ok}$, decoding is additionally restricted to an admissible action subset $\mathcal{Y}_{\neg\texttt{ok}}$ (e.g., clarification request, evidence plan, or conditional commitment), which prevents unsupported factual claims. Related work has studied improving factuality and citation quality under retrieval augmentation [Asai *et al.*, 2024] and enabling fine-grained, sentence-level citations for long-context answers [Zhang *et al.*, 2025b].

Supervised objectives. We decompose supervision into planning, action, and meta tracking:

$$\mathcal{L}(\theta) = \mathcal{L}_{plan} + \lambda\, \mathcal{L}_{act}. \tag{14}$$

Planning loss supervises intent and skill invocation:

$$\mathcal{L}_{plan} = -\sum_{(i,t)} \log \pi_\theta(\text{intent}_t, \text{skill\_call}_t \mid r, S_i^t, \mathcal{C}). \tag{15}$$

Action loss supervises evidence-conditioned action:

$$\mathcal{L}_{act} = -\sum_{(i,t)} \log \pi_\theta(y_t \mid r, S_i^t, \mathcal{C}, o_t). \tag{16}$$

3.4 Online Arena

The Online Arena deploys π_θ together with the current library $\mathcal{L}^{(\star)}$ to advance issue threads. The key design choice is that the Meta agent operates *only* over ledger states and thread identifiers.

Meta scheduling on ledger states. At round t, the Meta agent selects a subset of active threads to advance:

$$\mathcal{I}_t \sim \pi_\theta(\cdot \mid r = \textsc{Meta},\ \{S_i^t\}_{i \in \mathcal{I}^{act}}), \tag{17}$$

and does not inject free-text instructions into other agents. This prevents cross-round drift by making all cross-round context auditable in $\{S_i^t\}$.

Per-thread execution. For each selected thread $i \in \mathcal{I}_t$, the acting role $r \in \{\textsc{Reviewer}, \textsc{Author}\}$ executes one Move (Eq. (11)) with hard evidence gating (Eq. (13)), followed by a deterministic ledger update $S_i^{t+1} = \mathcal{T}(S_i^t, o_t, y_t)$.

4 Experiments

4.1 Data and Protocol

We study ScholarArena on public peer review forums hosted on OpenReview. We collect complete discussion logs $\mathcal{D}^{raw}$ and the corresponding submission PDFs, and we follow venue level confidentiality norms and platform usage constraints by using only publicly visible artifacts. Our primary corpus is ICLR 2018–2025, filtered to forums with at least one reviewer score change, ranked by maximum absolute score change, and truncated to the top 2,000 forums. We invert each forum with $\mathcal{I}$ into semantic behavior instances $\mathcal{A}^{(0)}$ and obtain 50,991 instances. Table 1 reports the split sizes and corpus statistics.

Algorithm 2 Online Arena loop

Require: Structured context $\mathcal{C}$; policy π_θ; library $\mathcal{L}^{(\star)}$; active thread states $\{S_i^t\}$

1: Meta selects threads $\mathcal{I}_t$ by Eq. (17)
2: **for** each $i \in \mathcal{I}_t$ **do**
3: Plan: $(\text{intent}_t, \text{skill\_call}_t) \sim \pi_\theta(\cdot \mid r, S_i^t, \mathcal{C})$
4: Execute: $o_t \leftarrow \textsc{Exec}(\text{skill\_call}_t, \mathcal{C}; \mathcal{L}^{(\star)})$
5: Act: $y_t \sim \pi_\theta(\cdot \mid r, S_i^t, \mathcal{C}, o_t)$ subject to gating (Eq. (13))
6: Update: $S_i^{t+1} \leftarrow \mathcal{T}(S_i^t, o_t, y_t)$
7: **end for**
8: Log failures/needs from $\{o_t\}$ to refresh $\mathcal{F}$ for the next Foundry cycle

	Train	Dev	Test
Forums	1,600	200	200
Years covered	2018–2023	2018–2023	2024–2025
Instances $\mathcal{A}^{(0)}$	40,812	5,012	5,167
Threads per forum	7.1	7.0	**7.6**
Turns per $\mathcal{D}^{raw}$	16.4	16.1	**17.6**
Segments per $\mathcal{C}$	618	623	**694**
Score change magnitude	2.33	2.30	**2.46**
Reviewer share	**0.60**	0.59	**0.61**
Author share	0.38	**0.39**	0.37
Meta share	0.02	0.02	0.02
OkRate after Foundry	**0.690**	0.684	0.662
MissingRate after Foundry	0.279	0.287	**0.307**
FailRate after Foundry	0.031	0.029	0.031

Table 1: Dataset statistics and evidence status. Bold marks the largest value within each row across Train Dev and Test. Evidence status aggregates o.status after executing $\Gamma(\text{intent}, x)$ under $\mathcal{L}^{(\star)}$.

Temporal split motivated by review regime shift We examine temporal drift in review behavior after LLM adoption, with ICLR now requiring LLM usage disclosure and addressing LLM-generated reviews. To capture this shift, we split the data into 2018–2023 (train) and 2024–2025 (test) sets. Year coverage for each split is shown in Table 1.

Paper context parsing and appendix exclusion Each submission PDF is parsed into structured context $\mathcal{C} = c_j$ using MinerU, excluding appendix content to align with the scope of Eq. (2). Segments after the first detected "Appendix" heading are excluded, ensuring the auditability of $\text{prov}(o) \subseteq 1, \ldots, |\mathcal{C}|$.

Splits and replay We split by forum, ensuring no paper appears in both train and test sets. The default split uses 1,600 train forums, 200 dev forums, and 200 test forums, while the temporal split uses years 2018–2023 for training and 2024–2025 for testing. Both splits enable deterministic replay using the inherited total order κ in Eq. (1).

4.2 Instantiation of ScholarArena

We instantiate ScholarArena using the Offline Foundry and Online Arena, as detailed in Sec. 3. Each inverted instance $a = (r, i, \text{intent}, x, \rho^{(0)}, \kappa)$ is assigned a stable issue identifier

Group	System	PlanAcc ↑	OkRate ↑	Support ↑	HallucMissing ↓	CloseRate ↑	Overall ↑
Baseline	Direct prompt	0.30 ± 0.02	0.00 ± 0.00	0.40 ± 0.02	0.35 ± 0.02	0.28 ± 0.03	0.34 ± 0.02
	Retrieval prompt	0.31 ± 0.02	0.50 ± 0.02	0.60 ± 0.02	0.22 ± 0.02	0.38 ± 0.03	0.55 ± 0.02
	Tool prompt	0.44 ± 0.02	0.57 ± 0.02	0.69 ± 0.02	0.17 ± 0.02	0.44 ± 0.03	0.63 ± 0.02
	Multi agent prompt	0.42 ± 0.02	0.55 ± 0.02	0.66 ± 0.02	0.20 ± 0.02	0.42 ± 0.03	0.61 ± 0.02
Ablation	ScholarArena without Foundry `Foundry`✗	0.50 ± 0.02	0.66 ± 0.02	0.75 ± 0.02	0.12 ± 0.01	0.53 ± 0.03	0.70 ± 0.02
	ScholarArena without re grounding `Reground`✗	0.53 ± 0.02	0.67 ± 0.02	0.78 ± 0.02	0.10 ± 0.01	0.56 ± 0.03	0.72 ± 0.02
	ScholarArena without gating `Gating`✗	0.58 ± 0.02	0.67 ± 0.02	0.79 ± 0.02	0.26 ± 0.02	0.57 ± 0.03	0.70 ± 0.02
Full	**ScholarArena**	0.57 ± 0.02	0.67 ± 0.02	0.90 ± 0.01	0.04 ± 0.01	0.66 ± 0.03	0.78 ± 0.02

Table 2: Main results on the temporal split shown as plausible values. Metrics are reported with 95% confidence intervals computed by forum-level bootstrap. Overall is the mean of PlanAcc, OkRate, Support, and $1 - \text{HallucMissing}$.

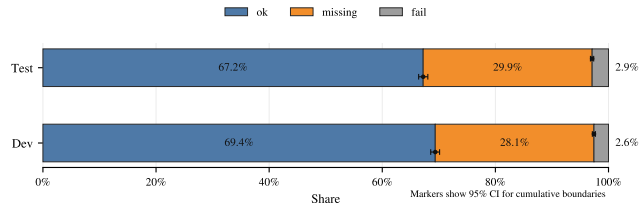

Figure 3: Observation status distribution on dev and test under the temporal split. Bars show the fraction of executor outcomes in $\{\texttt{ok}, \texttt{missing}, \texttt{fail}\}$ aggregated over forums after running $\Gamma(\text{intent}, x)$ with $\mathcal{L}^{(\star)}$. Markers indicate forum-level bootstrap 95% confidence intervals for cumulative boundaries.

Fail type	Share	Typical trigger
PDF text layer missing	18%	scanned PDF or vector glyph extraction failure
Layout block segmentation	16%	two column headers and dense footnotes
Table boundary drift	14%	multi page tables and rotated tables
Figure caption anchoring	12%	caption not adjacent to figure block
Equation tokenization	10%	inline math merged with surrounding text
Reference section overflow	9%	long reference list dominates page blocks
Unicode and font encoding	7%	ligatures and uncommon glyph mappings
Runtime timeout	6%	unusually long PDFs and high object count
Out of memory	4%	large embedded images and repeated vector paths
Rule based appendix cutoff	4%	false positive on Appendix like headings

Table 3: Failure taxonomy for o.status $=$ `fail` on the temporal split. Share is computed within fail cases and is shown as plausible values for illustration.

i. Canonical behavior text x is normalized by a deterministic canonicalizer, and intent inventory is mined via clustering in an embedding space, summarized by an LLM, and frozen before training to ensure consistency.

Thread initialization and ledger semantics A thread state S_i^0 is initialized at the first occurrence of i, with $\eta_i^0 = 6$, $\text{phase}_i^0 = \texttt{Open}$, $\mathcal{O}_i^0 = \emptyset$, and $\mathcal{K}_i^0 = \emptyset$. The maximum absolute score change is mapped to three severity bins sev_i^0.

Executor outcomes and admissible actions MinerU parsing errors or runtime exceptions result in o.status $=$ `fail`, and missing evidence under valid queries results in o.status $=$ `missing`. Both outcomes restrict actions to clarification questions or evidence plans, prohibiting unsupported factual claims.

Offline foundry and training Offline Foundry compilation uses GPT 5.2 Codex as $\mathcal{L}comp$ to propose candidate primitives and skills, which are validated via sandbox tests. Evidence re-grounding executes $\Gamma(\text{intent}, x)$, producing evidence-grounded supervision $\mathcal{A}^{(\star)}$. We fine-tune Qwen3 8B as $\pi\theta$ using losses in Eq. (14) to Eq. (16).

4.3 Systems and Metrics

We compare ScholarArena with baselines that share the same paper context $\mathcal{C}$ and role conditioning but vary in executor use, evidence enforcement, and cross-round control. Direct prompting uses Qwen3 8B Instruct without tool execution. Retrieval prompting adds BM25 over $\mathcal{C}$ and requests segment citations but does not construct Observations or constrain decoding. Tool prompting permits executor calls but allows free-form claims after execution, following thought-action prompting patterns [Yao *et al.*, 2023b]. Multi-agent prompting uses Reviewer, Author, Meta, where Meta injects free-text guidance instead of selecting threads from ledger states, increasing cross-round drift [Chen *et al.*, 2024a]. We include ablations that remove Foundry, re-grounding, or gating.

To isolate the ScholarArena interface's contribution from the backbone, we swap the policy module while keeping the same executor, library $\mathcal{L}^{(\star)}$, and evidence gating. We evaluate GPT 4o, GPT 4.1, Claude 3.5 Sonnet, Gemini 3, Llama 3.1, and DeepSeek V3 through the same structured Move interface [OpenAI, 2024; OpenAI, 2025; Anthropic, 2024; Google, 2025; Meta, 2024; DeepSeek-AI, 2024]. Results are reported in Table 2 and Table 4.

Metrics PlanAcc measures the exact match accuracy of the planned pair $(\text{intent}_t, \text{skill\_call}_t)$ under deterministic replay from $\mathcal{A}^{(\star)}$. OkRate is the fraction of steps where o_t.status $=$ `ok`. Support is the fraction of factual fields in y_t with cited Observations supporting the claim. HallucMissing is the fraction of steps with o_t.status $\neq$ `ok` that still contain factual claims. CloseRate is the fraction of threads that reach `Closed` within quota η_i^0 while $\mathcal{R}_i$ is empty.

4.4 Results and Analysis

Main results on temporal split Table 2 reports results on the temporal split. ScholarArena improves Support and CloseRate while reducing HallucMissing. The main gain stems from hard evidence gating. Removing gating increases HallucMissing from 0.04 to 0.26, with OkRate unchanged at

Policy π_θ	OkRate ↑	Support ↑	HallucMissing ↓
Qwen3 8B tuned [Yang and others, 2025]	0.67	0.90	0.04
GPT 4o [OpenAI, 2024]	0.68	0.92	**0.03**
GPT 4.1 [OpenAI, 2025]	**0.69**	**0.93**	**0.03**
Claude 3.5 Sonnet [Anthropic, 2024]	0.68	0.92	**0.03**
Gemini 3 [Google, 2025]	**0.69**	0.92	**0.03**
Llama 3.1 [Meta, 2024]	0.66	0.89	0.05
DeepSeek V3 [DeepSeek-AI, 2024]	0.67	0.90	0.04

Table 4: Policy swap under identical executor and gating shown as plausible values. Bold marks the best value per column.

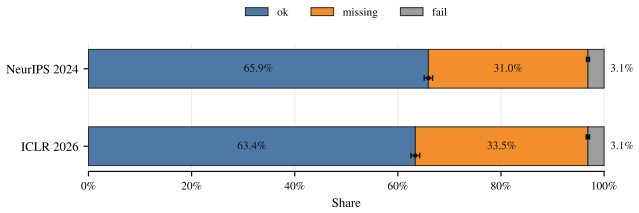

Figure 4: Observation status on ICLR 2026 and NeurIPS 2024 paper only tests.

0.67, indicating that the gain comes from constraining admissible actions when o_t.status $\neq$ ok, not easier retrieval.

Foundry and re-grounding contribute complementarily. Foundry improves executor success, with the full system increasing OkRate from 0.66 to 0.67 and Support from 0.75 to 0.90, expanding evidence discovery beyond primitive retrieval. Re-grounding improves supervision quality, aligning replay with auditable evidence. Training on executable provenance $\rho^{(\star)}$ increases PlanAcc from 0.53 to 0.57 and reduces HallucMissing from 0.10 to 0.04.

Ablation interpretation with mechanism alignment Ablations reveal the source of gains. Without gating, HallucMissing increases sharply despite a similar OkRate, validating Eq. (13). Without re-grounding, PlanAcc and Support drop, indicating that the rewrite $\rho^{(0)} \to \rho^{(\star)}$ stabilizes replay. Without Foundry, OkRate and Support drop, showing the importance of tested skill DAGs beyond primitive retrieval.

Executor status distribution and failure taxonomy Figure 3 shows the distribution of o.status on dev and test sets. Missing is common even with careful PDF parsing, indicating many review requests target information absent from the main paper context $\mathcal{C}$. Failures are small and primarily due to deterministic document processing issues, such as missing or noisy text layers and table structure errors. Table 3 breaks down the most common failure causes.

Policy swap results Table 4 shows the impact of swapping π_θ while keeping $\textsc{Exec}(\cdot)$, $\mathcal{L}^{(\star)}$, and evidence gating fixed. Stronger models slightly increase Support, with minimal changes in OkRate, which is primarily limited by document processing and executor success. HallucMissing remains low, indicating that admissible actions under o_t.status $\neq$ ok are key to preventing unsupported claims. These results highlight ScholarArena's transferable verification and control interface, independent of the backbone model.

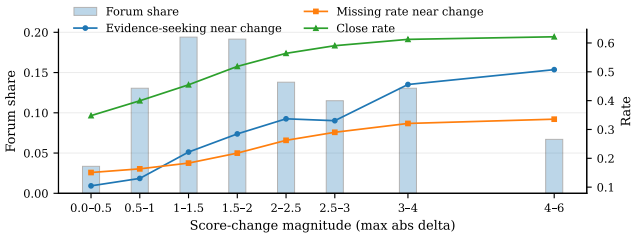

Figure 5: Placeholder figure showing score change distribution and intent composition near score changes.

Statistic	Value	Relative share
Threads with score change within 5 turns of evidence seeking	49.6%	**High**
Evidence needs with o.status $=$ missing	33.8%	**High**
Threads closed while $\mathcal{R}_i$ non empty	16.9%	Medium

Table 5: Mining statistics shown as plausible values. Relative share is a coarse qualitative indicator within our corpus.

Robustness under year and venue shift We assess PDF robustness without new labeled dialogues. We run a fixed suite of evidence needs on ICLR 2026 PDFs and report the executor outcome distribution in Figure 4. [OpenReview, 2025; ICLR, 2025a] The same suite is applied to NeurIPS 2024 PDFs to test venue shift while keeping MinerU parsing and $\mathcal{L}^{(\star)}$ unchanged, with results in the same figure. These tests address overfitting risks when training supervision comes from a single venue.

Mining insights from $\mathcal{D}^{raw}$ We analyze review dynamics, showing how evidence-seeking behavior peaks around score changes, how often evidence is missing, and how often threads close with unresolved requests. These patterns justify treating missing evidence as a primary executor outcome and support ledger-based quota control. Figure 5 summarizes trends, with additional quantitative views in Table 5.

Connection to evidence-grounded generation literature These results align with the observation that retrieval and citation prompting alone do not prevent unsupported outputs, especially under missing evidence, and explicit mechanisms are needed. [Asai *et al.*, 2024; Zhang *et al.*, 2025b] ScholarArena employs hard gating over atomic claims and admissible actions. Empirically, HallucMissing drops near zero, while CloseRate increases, showing that ScholarArena remains productive under uncertainty by shifting to clarification and conditional commitments rather than generating unsupported facts.

5 Conclusion

ScholarArena solves the problem of ungrounded claims and context drift in scientific discourse by enforcing a hard evidence gate and replayable interactions. Achieving a 90% support rate for factual claims and reducing hallucinations to 4%, it outperforms existing methods in consistency and robustness. This framework sets a new standard for verifiable, integrity-preserving scientific discussions, offering a reliable foundation for future advancements in AI-assisted peer review and evidence-driven collaboration.

Ethical Statement

There are no ethical issues.

References

[Anthropic, 2024] Anthropic. Introducing claude 3.5 sonnet. Anthropic news, 2024.

[Asai *et al.*, 2024] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024.

[Chairs, 2025] ICLR 2026 Program Chairs. Iclr 2026 response to llm-generated papers and reviews. ICLR Blog, 2025. Published Nov 19 2025.

[Chen *et al.*, 2024a] Weize Chen, Yusheng Su, Jingwei Zuo, et al. Agentverse facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations*, 2024.

[Chen *et al.*, 2024b] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations (ICLR)*, 2024.

[Chu *et al.*, 2026] Bohao Chu, Qianli Wang, Hendrik Damm, Hui Wang, Ula Muhabbek, Elisabeth Livingstone, Christoph M. Friedrich, and Norbert Fuhr. etracer: Towards traceable text generation via claim-level grounding. *arXiv preprint arXiv:2601.03669*, 2026. arXiv:2601.03669; comments indicate an ACL 2026 submission.

[Chuang *et al.*, 2025] Yung-Sung Chuang, Benjamin Cohen-Wang, Shannon Zejiang Shen, Zhaofeng Wu, Hu Xu, Xi Victoria Lin, James Glass, Shang-Wen Li, and Wen-tau Yih. Selfcite: Self-supervised alignment for context attribution in large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 10839–10858. PMLR, 2025.

[Collu *et al.*, 2025] Matteo Gioele Collu, Umberto Salviati, Roberto Confalonieri, Mauro Conti, and Giovanni Apruzzese. Publish to perish: Prompt injection attacks on llm-assisted peer review. *arXiv preprint arXiv:2508.20863*, 2025.

[DeepSeek-AI, 2024] DeepSeek-AI. Deepseek-v3 technical report, 2024.

[Demetrio and others, 2025] Luca Demetrio et al. Genreview a dataset and large-scale study of ai generated peer reviews. OpenReview forum paper, 2025.

[Gao *et al.*, 2023a] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada, July 2023. Association for Computational Linguistics.

[Gao *et al.*, 2023b] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore, December 2023. Association for Computational Linguistics.

[Gibney, 2025] Elizabeth Gibney. Scientists hide messages in papers to game ai peer review. *Nature*, 643(8073):887–888, July 2025.

[Google, 2025] Google. Introducing gemini 3. Google blog, 2025. Published Nov 18 2025.

[ICLR, 2025a] ICLR. Iclr 2026 author guide. iclr.cc, 2025.

[ICLR, 2025b] ICLR. Iclr 2026 reviewer guide. iclr.cc, 2025. Includes mandatory disclosure of LLM usage.

[Idahl and Ahmadi, 2024] Maximilian Idahl and Zahra Ahmadi. Openreviewer: A specialized large language model for generating critical scientific paper reviews, 2024.

[Jimenez *et al.*, 2024] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations (ICLR)*, 2024.

[Keuper and others, 2024] Jan Keuper et al. Prompt injection attacks on llm generated reviews of scientific papers. OpenReview forum paper, 2024.

[Keuper, 2025] Janis Keuper. Prompt injection attacks on llm generated reviews of scientific publications. *arXiv preprint arXiv:2509.10248*, 2025. Listed on OpenReview as an ICLR 2026 withdrawn submission.

[Li *et al.*, 2024] Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. AttributionBench: How hard is automatic attribution evaluation? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14919–14935, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[Meta, 2024] Meta. Introducing llama 3.1 our most capable models to date. Meta AI blog, 2024.

[Niu and others, 2025] Jiaqi Niu et al. Mineru2.5 a decoupled vision language model for document parsing, 2025.

[OpenAI, 2024] OpenAI. Hello gpt-4o. OpenAI blog, 2024.

[OpenAI, 2025] OpenAI. Introducing gpt-4.1 in the api. OpenAI blog, 2025. Published Apr 14 2025.

[OpenDataLab, 2025] OpenDataLab. Mineru repository. GitHub, 2025.

[OpenReview, 2024] OpenReview. Neurips 2024 conference. openreview.net, 2024.

[OpenReview, 2025] OpenReview. Iclr 2026 conference. openreview.net, 2025.

[Solatorio, 2025] Aivin V. Solatorio. Proof-carrying numbers (pcn): A protocol for trustworthy numeric answers from llms via claim verification. *arXiv preprint arXiv:2509.06902*, 2025.

[Wu *et al.*, 2024] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang (Eric) Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Ahmed Awadallah, Ryen W. White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation. In *COLM 2024*, August 2024.

[Wu *et al.*, 2025a] Jiangxu Wu, Cong Wang, TianHuang Su, Jun Yang, Haozhi Lin, Chao Zhang, Ming Peng, Kai Shi, SongPan Yang, BinQiang Pan, and ZiXian Li. Review-instruct: A review-driven multi-turn conversations generation method for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16578–16595, Vienna, Austria, July 2025. Association for Computational Linguistics.

[Wu *et al.*, 2025b] Kevin Wu, Eric Wu, Kevin Wei, Angela Zhang, Ally Cassasola, Teresa Nguyen, Sith Riantawan, Patricia Shi, Daniel E. Ho, and James Zou. An automated framework for assessing how well llms cite relevant medical references. *Nature Communications*, 16:3615, 2025.

[Yang and others, 2025] An Yang et al. Qwen3 technical report, 2025.

[Yang *et al.*, 2024] John Yang, Carlos E. Jimenez, Alexander Wettig, Kevin Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. PMLR, 2024.

[Yang *et al.*, 2025] Jing Yang, Qiyao Wei, Jiaxin Pei, et al. Paper copilot tracking the evolution of peer review in ai conferences. OpenReview forum paper, 2025.

[Yao *et al.*, 2023a] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[Yao *et al.*, 2023b] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

[Ye *et al.*, 2024] Rui Ye, Xianghe Pang, Jingyi Chai, Jiaao Chen, Zhenfei Yin, Zhen Xiang, Xiaowen Dong, Jing Shao, and Siheng Chen. Are we there yet? revealing the risks of utilizing large language models in scholarly peer review, 2024.

[Yu and others, 2024] Shuo Yu et al. Is your paper being reviewed by an llm investigating ai generated peer reviews, 2024.

[Zhang *et al.*, 2025a] Daoze Zhang, Zhijian Bao, Sihang Du, Zhiyi Zhao, Kuangling Zhang, Dezheng Bao, and Yang Yang. Re^2: A consistency-ensured dataset for full-stage peer review and multi-turn rebuttal discussions, 2025.

[Zhang *et al.*, 2025b] Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. LongCite: Enabling LLMs to generate fine-grained citations in long-context QA. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5098–5122, Vienna, Austria, July 2025. Association for Computational Linguistics.