

Assembling the CREW: A Collaborative Multi-agent Reinforcement Learning Framework for Automated Related Work Generation

Hai-Dang Dang^{1*}, Bao-Yen Pham^{1*}, Bao Nguyen²

Tran Thi Huong³, Huynh Thi Thanh Binh¹

¹Hanoi University of Science and Technology ²The Chinese University of Hong Kong

³Hanoi University of Industry

Abstract

Automatic Related Work Generation (RWG) significantly reduces the human time and effort required to author the Related Work Section (RWS) of a research paper. However, prior methods leveraging multi-agent Large Language Models (LLMs) typically rely on a predefined workflow, where each agent is responsible for a specific step in the entire process. This rigid, static inter-agent coordination limits the adaptive collaboration required to synthesize complex scientific literature. To address this limitation, we propose CREW (Collaborative Reinforcement Learning for Related Work Generation), a novel framework where LLM agents bypass heuristic pipelines to dynamically coordinate by autonomously selecting actions, such as Retrieve, Disseminate, Compose, and Critique, driven by a policy optimized via Independent Proximal Policy Optimization (IPPO). Extensive experiments on a standard RWG benchmark demonstrate that our approach yields substantial quality improvements over strong existing baselines, while significantly reducing token costs. Code is available at <https://github.com/YenPBao/CREW-Collaborative-MARL.git>

1 Introduction

To translate an initial idea into scientific research, scholars often explore a range of literature, carefully summarizing, synthesizing, and contrasting earlier findings. This process is both time-consuming and labor-intensive. Indeed, literature synthesis represents a major bottleneck in the scientific workflow, particularly given the rapid growth of scientific publications. Addressing this challenge, Hoang and Kan (2010) introduced Related Work Generation (RWG) as a research direction within Natural Language Processing (NLP). The task aims to automatically transform an initial concept, typically consisting of a title and an abstract,

together with a list of potential related references, into a coherent and logical related work draft that comprehensively covers key findings from prior studies with accurate citations.

Overall, RWG research has evolved along three main methodological directions. Firstly, extractive approaches select and concatenate sentences directly from source documents, making them simple to implement and generally faithful to the original content. However, their limited ability to paraphrase and abstract often results in related-work sections that are redundant and poorly structured (Hu and Wan, 2014; Chen and Zhuge, 2016; Deng et al., 2021). Secondly, abstractive supervised models move beyond sentence extraction by incorporating task-specific components and structured representations, including relation-aware RWG models and structured scientific summarization methods (Chen et al., 2021, 2022; Xiao et al., 2022). Although these models achieve strong performance within their training domains, their reliance on task-specific training and annotated datasets limits scalability and adaptation to new domains or topics. Thirdly, LLM-based methods leverage strong generative capabilities to produce coherent and logically structured related work sections. However, single-LLM approaches remain prone to hallucination and constrained by context capacity, as high-quality synthesis often requires jointly reasoning over many references. Even with long-context models (e.g., GPT-4o with a 128K-token capacity), simply concatenating all references into a single input remains ineffective, as long contexts can dilute attention and cause important inter-document relationships to be overlooked. Retrieval-based alternatives reduce the context burden but may omit relevant evidence and weaken coverage (Izacard and Grave, 2021; Wu et al., 2022). To overcome these obstacles, recent research has leveraged multi-agent LLM frameworks to decompose the overall RWG process into smaller subtasks, assigning each

*Both authors contributed equally to this research.

agent to a specific part of the workflow. This division of labor reduces the workload each agent must process, thereby lowering the risk of exceeding the context window (Wu et al., 2023; Li et al., 2023; Liu et al., 2025). However, fixed subtask assignments force the overall process to proceed sequentially, increasing inference time. More importantly, these approaches lack a mechanism that allows multiple agents to collaborate on the same subtask when it becomes critical. If too few agents are assigned to an important subtask at a particular stage, it can become a bottleneck, producing low-quality intermediate outputs that limit subsequent agents. For example, after agents have moved from reading to drafting, they may find that the collected evidence remains insufficient, making further drafting ineffective despite multiple agents being assigned to writing. Therefore, agents need mechanisms to proactively select context-dependent actions and coordinate more effectively.

Motivated by these observations, we introduce a multi-agent framework that enables flexible, context-dependent action selection through Multi-Agent Reinforcement Learning (MARL). MARL enables independently operating agents to coordinate their decisions through a shared learned policy rather than a centralized orchestrator, thereby reducing computational overhead and improving scalability. To achieve this, we formulate RWG as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), where each agent receives a distinct local observation consisting of its internal knowledge state, shared resources, and peer actions. At the same time, our environment provides a team-level global reward based on the quality of the collaboratively constructed draft and its alignment with human-written references. Importantly, beyond the conventional actions of literature retrieval and drafting employed in prior work, we introduce two additional actions, *knowledge dissemination* and *draft critiquing*, to promote proactive information sharing and adaptive inter-agent coordination throughout the RWG process. The framework is then optimized using IPPO with parameter sharing to ensure computational efficiency and scalability. Extensive experiments show that our framework improves over competitive baselines in LLM-as-a-judge evaluation and valid citation grounding, while reducing inference time and token costs.

Our contributions can be summarized as follows:

1. We introduce two collaborative actions, Disseminate and Critique, which enable agents to share useful knowledge and provide mutual feedback, improving both coordination and the quality of the generated draft.
2. We propose a MARL framework based on a Dec-POMDP formulation, allowing agents to proactively select context-dependent actions from their local observations. We further optimize the decision-making policy using IPPO with parameter sharing for efficient multi-agent coordination.
3. We conduct extensive experiments showing that our framework improves over strong existing methods across multiple evaluation metrics. Additional transferability experiments demonstrate that a decision-making policy trained with one LLM can be reused with different inference LLMs, indicating that the learned coordination strategy is not tied to a single backbone model.

2 Related Work

RWG has attracted growing attention for its role in supporting scientific writing and is commonly categorized into Extractive RWG (ERWG) and Abstractive RWG (ARWG) (Li and Ouyang, 2024). ERWG methods construct related work sections by selecting and concatenating representative sentences from cited papers. Early approaches such as ReWoS (Hoang and Kan, 2010) rely on hierarchical rule-based extraction at both coarse- and fine-grained topic levels, while subsequent methods incorporate topic modeling and learning-based importance estimation, e.g., clustering sentences with PLSA and ranking them using regression models (Hu and Wan, 2014). To better capture target-reference relationships, later studies exploit citation contexts (Chen and Zhuge, 2016) or sentence-level embeddings with coherence-aware reordering, as in SERGE (Deng et al., 2021). Graph-based methods explicitly model inter-paper relations, such as ToC-RWG (Wang et al., 2020) and bibliographic graph-based neural summarization (Wang et al., 2018). Overall, despite being citation-faithful, ERWG methods remain limited in producing comparative and analytical discussions, as they depend heavily on pre-existing comparative sentences that are difficult to obtain in practice.

To address these limitations, recent work has shifted toward ARWG, where models generate

related-work text in their own words. ARWG approaches can be broadly categorized into citation-level and section-level generation. Citation-level ARWG focuses on generating short descriptions for individual citations based on local citation context and cited paper content, typically using citation context and abstracts due to input length constraints (Luu et al., 2021; Gu and Hahnloser, 2024; Jung et al., 2022; Li et al., 2024; Xiao et al., 2022). These methods generally assume one cited paper per citation and model citations in isolation, limiting their ability to capture cross-paper relationships. Section-level ARWG extends this formulation by generating longer related-work paragraphs that contain multiple citations, explicitly modeling interactions among referenced papers to improve global coherence. Following early attention-based models (Wang et al., 2018), a common strategy is to model papers and their interactions as graphs, enabling structured reasoning over citations, topics, and semantic dependencies. Building upon this paradigm, Chen et al. (2021) introduced RRG, which captures relationships among referenced papers using a relation graph and iteratively refines document representations prior to generation. Similarly, Chen et al. (2022) proposed TAG, integrating graph-guided interactions at both the encoding and decoding stages, alongside contrastive learning, to encourage well-structured outputs. In this work, we focus on section-level ARWG.

Despite their demonstrated effectiveness, existing section-level ARWG methods are typically trained from scratch, resulting in substantial computational and memory overhead when scaled to large corpora with many reference papers. To overcome this, a promising research direction is to leverage the robust generative capabilities of pretrained LLMs to bypass costly retraining. However, fully unlocking LLMs for this task requires overcoming their inherent context-window limitations. This challenge is especially pronounced in section-level ARWG, where identifying complex latent links necessitates simultaneously processing a massive volume of text across numerous reference papers. Prior attempts to handle such input-length constraints commonly rely on long-context models, iterative reading-and-summarization strategies (Miller et al., 2016; Chevalier et al., 2023), and retrieval-based methods (Izacard and Grave, 2021; Wu et al., 2022). However, each direction has notable drawbacks: long-context processing can dilute the effectiveness of attention, iterative

reading may overwrite early evidence, and retrieval risks omitting crucial inter-document nuances.

To circumvent these limitations, our framework leverages multi-LLM collaboration to effectively distribute reasoning and knowledge processing across multiple agents. Unlike many existing multi-LLM systems that rely on rigid predefined interaction patterns or static self-reflection loops (Wu et al., 2023; Li et al., 2023; Shinn et al., 2023), our approach proposes a highly adaptable paradigm grounded in multi-agent reinforcement learning. For instance, while Liu et al. (2025) recently proposed a multi-agent system that uses citation graphs and specialized roles (Selector, Reader, and Writer), imposing these fixed roles forces agents into sequential pipelines, restricting their ability to address subtasks as they become critical. Conversely, our approach offers a flexible collaboration paradigm by introducing explicit dynamic cooperation via discussion actions among complementary LLM agents and enabling step-wise situational action selection, which allows agents to dynamically determine subsequent actions, reducing unnecessary interactions and computational overhead while improving reasoning reliability.

3 Methodology

Following the setting of Select, Read, and Write (Liu et al., 2025), the input consists of the author’s initial idea, represented by the title $\mathcal{T}_C$ and abstract $\mathcal{A}_C$ of the citing paper C , together with a set of potential reference papers $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$. Given these inputs, our framework transforms the reference set into a coherent related work section $\mathcal{S}$ that synthesizes the key ideas most relevant to C with accurate citation grounding. The overall workflow of the proposed framework is summarized in Figure 1.

3.1 Dec-POMDP Formulation

In this work, we cast RWG as a Dec-POMDP, which naturally accommodates (i) *partial observability*, since each of M agent sees only the shared working documents and its own private knowledge, (ii) all agents act *independently* based on their own distinct observations, without relying on a centralized controller, thereby distributing the computation and enhancing scalability. Thus, agents operate independently and in parallel over a series of timesteps: at each timestep t , the environment is in state s_t , each agent i receives an observation o_t^i

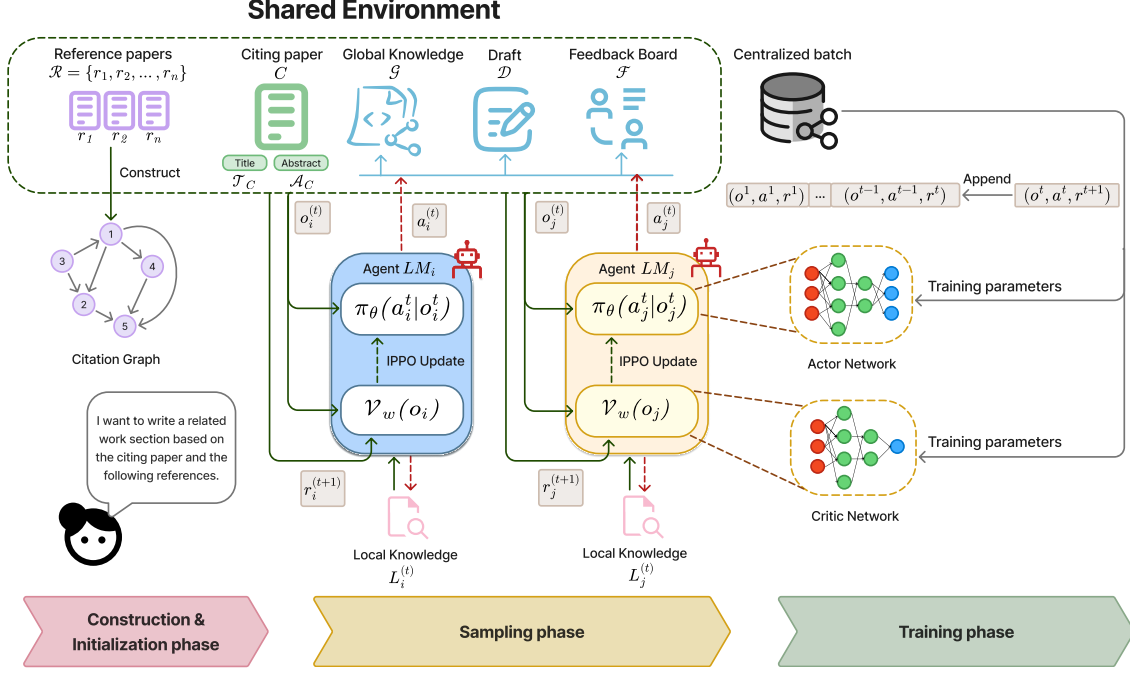


Figure 1: Overview of our multi-agent framework. Agents operate independently and in parallel over a series of timesteps. At each step, each agent observes the input information and current working documents, selects one of four actions, and receives a reward guided by the human-written related-work section as the gold standard. The collected interactions are aggregated into a buffer to train the policy, which is then used as the agents’ decision-making policy during inference to produce the final output.

as its partial view of the state, selects an action a_t^i , and then receives a reward r_t^i from the environment. We define the *states*, *observations*, *actions*, and *rewards* as follows.

States. At each time step t , the global state $s_t \in \mathcal{S}$ is defined as:

$$s_t = \langle \mathcal{T}_C, \mathcal{A}_C, \mathcal{R}, L_t, G_t, D_t, F_t \rangle, \quad (1)$$

where the first three components are the static task inputs introduced above. The remaining four components correspond to the four types of documents that we propose for the collaborative writing process. **Local knowledge** $L_t = \{L_t^1, \dots, L_t^M\}$ is the collection of agent-private knowledge states; each L_t^i accumulates reading summaries obtained through Agent i ’s individual exploration of the citation graph via the Retrieve action and remains *invisible* to other agents. **Global knowledge** G_t is a shared knowledge base accessible to all agents, aggregated from individual local contributions via the Disseminate action; it serves as the collective understanding of the literature that any agent can draw upon when drafting or critiquing. **Draft** D_t is the evolving text of the related-work section, iteratively refined by agents through the Compose action. **Feedback** F_t stores the accumulated critiques and

revision suggestions produced by agents through the Critique action, thereby providing structured signals to systematically guide subsequent draft revisions. The detailed explanation of the *Actions* is provided below.

Observations. Since Agent i cannot access the local knowledge of other agents, each agent receives only a partial observation $o_t^i \in \mathcal{O}^i$:

$$o_t^i = \text{Embed}(\langle L_t^i, G_t, D_t, F_t \rangle), \quad (2)$$

Directly feeding the raw textual content of each document into the policy network would result in an excessively large observation space, making convergence during training prohibitively difficult. To maintain a compact state representation, the embedding operator $\text{Embed}(\cdot)$ encodes each component $k \in \{L, G, D, F\}$ into a five-dimensional numerical feature vector $\mathbf{x}_k = [v_{1,k}, \dots, v_{5,k}]$, with all feature values clipped to $[0, 1]$. The resulting observation has 20 dimensions and is independent of the number of agents M , so the policy architecture does not need to change when the team size varies. The five features are defined as follows:

(i) **Semantic relevance** ($v_{1,k}$): The cosine similarity, denoted by $c(\cdot, \cdot)$, between the embedding $\mathbf{e}_k$ of component k and the initial-idea embedding

$\mathbf{e}_{\text{idea}} = \text{embed}(\mathcal{T}_C \oplus \mathcal{A}_C)$, where the initial idea consists of the title $\mathcal{T}_C$ and abstract $\mathcal{A}_C$:

$$v_{1,k} = c(\mathbf{e}_k, \mathbf{e}_{\text{idea}}). \quad (3)$$

(ii) **Coverage and Novelty** ($v_{2,k}$): A component-specific measure defined as:

$$v_{2,k} = \begin{cases} 1 - \alpha \delta_{\text{id}} - (1 - \alpha) \delta_{\text{sem}}, & k = L, \\ 1 - c(\mathbf{e}_{L_i}, \mathbf{e}_G), & k = G, \\ c(\mathbf{e}_D, \mathbf{e}_G), & k = D, \\ c(\mathbf{e}_F, \mathbf{e}_{\text{idea}} - \mathbf{e}_D), & k = F, \end{cases} \quad (4)$$

where for Local Knowledge (L), the coverage feature penalizes redundancy against other agents. Specifically, $\delta_{\text{id}} = |\mathcal{P}_i \cap \mathcal{P}_{\neg i}|/|\mathcal{P}_i|$ calculates the identifier-level overlap ratio between the set of papers read by Agent i ($\mathcal{P}_i$) and those read by all other agents ($\mathcal{P}_{\neg i}$). Concurrently, $\delta_{\text{sem}} = \max_{j \neq i} c(\mathbf{e}_{L_i}, \mathbf{e}_{L_j})$ captures the maximum semantic similarity between Agent i 's local knowledge and that of any peer. For Global Knowledge (G), it measures semantic novelty relative to the shared knowledge. For the Draft (D), it quantifies semantic coverage with reward to the G . For Feedback Board F , it measures alignment with the *gap vector* $\mathbf{e}_{\text{idea}} - \mathbf{e}_D$.

(iii) **Personal action distribution** ($v_{3,k}$): The historical frequency with which Agent i has selected action k , defined explicitly as:

$$v_{3,k} = \frac{n_k^i}{\sum_{a \in \mathcal{A}} n_a^i}, \quad (5)$$

where n_k^i is the accumulated count of action k executed by Agent i .

(iv) **Token capacity** ($v_{4,k}$): The remaining normalized token capacity for action k :

$$v_{4,k} = \max\left(0, 1 - \frac{\tau_k}{\tau_{\text{max}}}\right), \quad (6)$$

where τ_k is the current token usage associated with action k , and τ_{max} represents the predefined maximum allowable token budget.

(v) **Social activity distribution** ($v_{5,k}$): The proportion of action k performed by *other* agents,

$$v_{5,k} = \frac{n_k^{\neg i}}{n_k^{\text{all}}}, \quad (7)$$

providing Agent i with awareness of team-level coordination, where $n_k^{\neg i}$ is the number of times

action k has been performed by all agents except Agent i , and n_k^{all} is the total number of times action k has been performed by all agents.

Actions. The joint action space is $\mathcal{A} = \mathcal{A}^1 \times \dots \times \mathcal{A}^M$; each agent i selects one of four actions: Retrieve, Disseminate, Compose, or Critique, and the joint action is $\mathbf{a} = (a_t^1, \dots, a_t^M)$. Each action is realized by a dedicated prompt template (Appendix G). The Retrieve action selects a candidate paper p_c^{t-1} to read and updates the agent's local knowledge. Disseminate merges local into shared knowledge; Critique reviews the current draft; Compose revises the draft from all available sources:

$$L_t^i = \text{Retrieve}(L_{t-1}^i, p_c^{t-1}) \quad (8)$$

$$G_t = \text{Disseminate}(L_t^i, G_{t-1}) \quad (9)$$

$$F_t = \text{Critique}(L_t^i, D_{t-1}) \quad (10)$$

$$D_t = \text{Compose}(D_{t-1}, G_{t-1}, L_t^i, F_{t-1}) \quad (11)$$

Rewards. We adopt a *delta-based* cooperative reward structure. The reward decomposes into two terms: a *draft-quality gain* reflecting overall synthesis progress, and a *component-quality gain* reflecting improvements in the specific artifact targeted by the agent's action:

$$r_t^i = (1 - \lambda^t) \Delta Q_D^{(t)} + \lambda^t \Delta Q_{k(a^i)}^{(t)}, \quad (12)$$

where $Q_k^{(t)}$ denotes the quality of component $k \in \{L, G, D, F\}$ at time step t , and $\Delta Q_k^{(t)} = Q_k^{(t)} - Q_k^{(t-1)}$ denotes its quality gain. The decaying coefficient λ^t encourages agents to first improve the background documents before gradually shifting attention toward composing the final draft. The quality Q_k is estimated by document-specific heuristic functions, depending on the type of document. These heuristics primarily reference the human-written related-work section, whose embedding is denoted by $\mathbf{e}_{\text{gold}}$. Detailed quality formulas and the credit-assignment procedure are provided in Appendix A.2 and Appendix A.3.

3.2 Proposed Algorithm

As illustrated in Figure 1, we optimize the cooperative policy using parameter-shared IPPO, with full pseudocode provided in Appendix A.1. The framework alternates between sampling and training: agents interact asynchronously under a shared behavioral policy, collect rewards from Section 3.1, and aggregate rollouts to update a single shared

Actor and Critic. This design keeps the policy scalable while reducing coordination overhead across agents.

The sampling phase. During data collection, M agents asynchronously select actions from the discrete space $\mathcal{A}$ based on the shared policy π_θ . This interaction occurs over a subset of papers, spanning several time steps across multiple epochs. The reward is calculated by measuring the specific semantic and structural improvements made to the draft, compared with the ground truth one. These transition tuples $(o_t^i, a_t^i, r_t^i, o_{t+1}^i)$ are temporarily stored in local experience buffers.

The training phase. Once a predefined stopping condition is met, trajectories from all local buffers are aggregated into a centralized batch $\mathcal{W}$ to update the shared networks. For each sample $j \in \mathcal{W}$, PPO defines the probability ratio

$$\rho_j(\theta) = \frac{\pi_\theta(a_j | o_j)}{\pi_{\theta_{\text{old}}}(a_j | o_j)}. \quad (13)$$

The clipped surrogate objective for the Actor is

$$\mathcal{L}^{\text{CLIP}}(\theta) = \frac{1}{|\mathcal{W}|} \sum_{j \in \mathcal{W}} \min \left(\rho_j(\theta) \hat{A}_j, \text{clip}(\rho_j(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_j \right), \quad (14)$$

where $\hat{A}_j$ is estimated with Generalized Advantage Estimation (GAE) (Schulman et al., 2016). The Critic is trained against the empirical return $\hat{R}_j$ using the value loss

$$\mathcal{L}^{\text{VF}}(\phi) = \frac{1}{|\mathcal{W}|} \sum_{j \in \mathcal{W}} \left(V_\phi(o_j) - \hat{R}_j \right)^2. \quad (15)$$

During optimization, the Actor maximizes the clipped surrogate objective while the Critic minimizes the value loss; this centralized optimization cycle is repeated until the policy converges.

The actor and critic networks. To ensure parameter efficiency and facilitate knowledge transfer across agents, our framework maintains a single Actor network (π_θ) and Critic network (V_ϕ) applied uniformly to all agents. The Actor maps the 20-dimensional observation vector to action logits $\mathbf{z} \in \mathbb{R}^4$ for categorical sampling, while the Critic outputs a scalar value estimate $\hat{V}_\phi(o_t^i)$ as the variance-reduction baseline. Both networks share the same compact feedforward design, differing only in their final output heads; detailed configurations are provided in Table 6 in Appendix D.

4 Experimental Results and Discussions

To evaluate the CREW framework, we design a series of experiments addressing overall performance, architectural choices, and cost efficiency.

4.1 Experimental Setup

Dataset. We evaluate our framework on the OARELATEDWORK dataset (Docekal et al., 2024), currently the primary dataset supporting full-text-based related work generation. Specifically, we use 2% of the papers (1829 papers) for training and 0.4% of the papers (366 papers) for evaluation, as detailed in Appendix B.

Implementation Details. For the main results, we use Qwen 2.5 72B Instruct¹ for both policy training and inference. In the transferability study, we keep the trained controller fixed and swap the inference backbone to LLaMA 8B², Gemini 2.0 Flash³, Qwen 2.5 72B Instruct, and GPT-4o⁴. Complete IPPO hyperparameters and network architecture are provided in Appendix D, while prompt templates are provided in Appendix G.

Evaluation Metrics. Following the evaluation protocol of Select, Read, and Write (SRW) (Liu et al., 2025), we adopt an *LLM-as-a-Judge* setup. For robustness and fairness, final scores are computed as the arithmetic mean of ratings from three judges on a 1–5 Likert scale: GPT-4o, LLaMA 3.3 70B Instruct⁵, and DeepSeek-V3⁶. In addition to standard metrics (*Coverage*, *Logic*, and *Relevance*), we report *Citation Verification*, defined as the proportion of generated citations that exist in the database. To assess operational efficiency, we further report total input/output tokens and inference time for the baselines. Detailed definitions and evaluation procedures for all metrics are provided in Appendix C.

4.2 Baselines

We situate our proposed CREW framework against widely-adopted baseline paradigms (with detailed behavioral analysis in Appendix D.3):

Multi-Agent Methods. We compare against SRW (Liu et al., 2025) and our own Orchestrator baseline, inspired by HuggingGPT (Shen et al., 2023), with both baselines using Qwen 2.5 72B

¹Version qwen-2.5-72b-september2024

²Version llama-3.1-8b-instruct-july2024

³Version gemini-2.0-flash-february2025

⁴Version gpt-4o-may2024

⁵Version llama-3.3-70b-december2024

⁶Version deepseek-v3-december2024

Table 1: Performance of different models on the OARELATEDWORK dataset. The best and the runner-up results are shown in **bold** and underlined, respectively. Our method outperforms the strong existing baseline across all quality and citation metrics while using fewer tokens.

Model	LLM-based Evaluation (Avg.)				Citation	Cost Efficiency	
	Cov. $\uparrow$	Logic $\uparrow$	Rel. $\uparrow$	Overall $\uparrow$	Ver.(%) $\uparrow$	Tokens $\downarrow$	Inf. Time (s) $\downarrow$
PRIMERA	1.18	1.41	2.04	1.54	0.00	0.34K	2
GPT-4o (Long Context)	3.20	3.28	3.76	3.41	48.61	6.61K	38
GPT-4o (RAG)	3.19	3.30	3.76	3.42	<u>96.40</u>	<u>3.68K</u>	<u>37</u>
SRW (Qwen 2.5 72B)	3.38	3.73	<u>4.17</u>	3.76	80.40	210.37K	585
Orchestrator (Qwen 2.5 72B)	<u>3.41</u>	<u>3.71</u>	4.27	<u>3.80</u>	99.67	181.29K	822
CREW (Qwen 2.5 72B, 4 Agents)	3.44	3.73	4.27	3.82	<u>99.33</u>	198.74K	755

Table 2: Comparison with a naive orchestration strategy on the OARELATEDWORK dataset. The best and the runner-up results are shown in **bold** and underlined, respectively.

Model	LLM-based Evaluation (Avg.)				Citation
	Cov. $\uparrow$	Logic $\uparrow$	Rel. $\uparrow$	Overall $\uparrow$	Ver.(%) $\uparrow$
Naive Orchestrator (Qwen 2.5 72B)	3.22	3.24	3.73	3.40	98.67
Orchestrator (Qwen 2.5 72B)	<u>3.41</u>	<u>3.71</u>	4.27	<u>3.80</u>	99.67
CREW (Qwen 2.5 72B, 4 Agents)	3.44	3.73	4.27	3.82	<u>99.33</u>

Instruct to match the main setting. The Orchestrator incorporates CREW’s proposed architecture, including the shared workspace, local knowledge, and action space; its key distinction is that it replaces the autonomous RL-based policy controller with a centralized LLM coordinator that assigns actions to the agents. Further implementation and behavioral details are provided in Appendix D.3.

Single LLM Methods. We evaluate GPT-4o with either the full document context or standard Retrieval-Augmented Generation (RAG) (Lewis et al., 2020).

Abstractive Methods. We benchmark against PRIMERA (Xiao et al., 2022), a strong sequence-to-sequence model for multi-document synthesis.

4.3 Comparison Results

We present four primary experiments to validate our framework’s capabilities, covering main performance, model transferability, benchmark transferability on a different dataset, and ablation analysis. The scalability analysis over different numbers of agents is deferred to Appendix E.

Main Results. As shown in Table 1, CREW improves over the SRW baseline in overall LLM-judged quality by 0.06 points on the 1–5 scale and citation verification by 23.54%, while reducing token usage by 12.90%. These gains indicate that learned coordination improves generation quality,

citation validity, and efficiency. Compared with our Orchestrator baseline, CREW achieves higher overall quality (+0.02 points on the 1–5 scale) and lower inference time. This modest quality margin is expected, as the Orchestrator inherits CREW’s architecture, including the shared workspace, local knowledge, and action space, and differs primarily by replacing the learned decentralized policy with a centralized LLM coordinator, as described in Section 4.2. To examine this structural contribution, we compare against a Naive Orchestrator adapted from HuggingGPT (Shen et al., 2023), where the coordinator handles problem decomposition, task generation, and iterative task assignment based on prior agent feedback. Table 2 shows that this variant obtains Overall and Citation Verification scores of 3.40 and 98.67%, respectively, substantially underperforming both the architecture-aware Orchestrator (3.80 Overall) and CREW (3.82 Overall). These results indicate that the strong performance of the Orchestrator derives substantially from CREW’s structural design. CREW further avoids the coordinator as a single point of failure. Under a simulated 5% API failure rate, CREW and the Orchestrator encounter similar numbers of failed calls, but CREW finishes in 812 rather than 1012 seconds because unaffected agents can continue operating independently without depending on an orchestrator.

Table 3: Model transferability across inference LLMs using the controller trained with Qwen 2.5 72B Instruct.

Inference Model	LLM-based Evaluation (Avg.)				Citation	Cost Efficiency
	Cov. $\uparrow$	Logic $\uparrow$	Rel. $\uparrow$	Overall $\uparrow$	Ver.(%) $\uparrow$	Tokens $\downarrow$
LLaMA 8B (Long-context)	2.97	2.99	3.17	3.04	53.23	20.52K
LLaMA 8B + CREW	3.31	3.42	3.96	3.57	82.01	161.49K
Gemini 2.0 Flash (Long-context)	3.11	3.17	3.54	3.28	98.61	20.93K
Gemini 2.0 Flash + CREW	3.31	3.60	4.23	3.71	96.70	156.10K
Qwen 2.5 72B Instruct (Long-context)	3.19	3.35	3.73	3.42	65.83	21.57K
Qwen 2.5 72B Instruct + CREW	3.31	3.33	4.17	3.79	96.20	121.81K
GPT-4o (Long Context)	3.20	3.28	3.76	3.42	48.61	6.61K
GPT-4o + CREW	3.43	3.77	4.26	3.82	96.70	139.10K

Table 4: Cross-domain transferability on Multi-News without policy retraining or reward adaptation. Best and runner-up results are shown in **bold** and underlined.

Framework	LLM-based Evaluation (Avg.)				ROUGE		
	Overall $\uparrow$	Cov. $\uparrow$	Logic $\uparrow$	Rel. $\uparrow$	R-1 $\uparrow$	R-2 $\uparrow$	R-L $\uparrow$
GPT-4o (Long Context)	3.53	3.36	3.70	3.54	40.57	11.56	19.24
SRW (Qwen 2.5 72B)	3.37	3.20	3.59	3.31	39.13	12.18	19.24
Orchestrator (Qwen 2.5 72B)	<u>3.75</u>	<u>3.53</u>	<u>3.76</u>	<u>3.96</u>	<u>41.81</u>	<u>13.03</u>	20.61
CREW (Ours)	3.84	3.65	3.91	3.97	43.74	14.08	<u>20.29</u>

Table 5: Ablation study of key CREW components and their effects on generation quality and citation reliability.

Component Configuration	LLM-based Evaluation (Avg.)				Citation	Cost Efficiency
	Cov. $\uparrow$	Logic $\uparrow$	Rel. $\uparrow$	Overall $\uparrow$	Ver.(%) $\uparrow$	Tokens $\downarrow$
w/o Draft Quality Reward	3.50	3.70	4.30	3.83	44.07	139.02K
w/o Feedback Action	2.33	2.54	2.85	2.58	10.00	37.54K
w/o Social Activity Observation	2.97	3.30	3.65	3.30	0.00	85.57K
CREW (Full System)	3.45	3.73	4.27	3.82	96.20	152.70K

Model Transferability. We evaluate whether the learned 3-agent coordination policy transfers across backbones without retraining the controller. As shown in Table 3, the policy trained with Qwen 2.5 72B Instruct remains effective when agents use other inference models to execute selected actions. CREW raises the overall score of LLaMA 8B from 3.04 to 3.57 and Gemini 2.0 Flash from 3.28 to 3.71, while GPT-4o reaches 3.82. Although citation verification varies with the generation model, every transferred configuration substantially exceeds the corresponding long-context setting in overall quality. These results indicate that the controller learns a reusable coordination protocol rather than model-specific generation patterns. The consistent gains across model sizes and providers suggest that coordination knowledge is largely decoupled from the linguistic capabilities of the executor, allowing CREW to adopt upgraded backbones without repeating policy training.

Benchmark Transferability. We study cross-domain generalization on Multi-News (Fabbri et al., 2019), where the system must synthesize multiple news articles describing the same event. We adapt only the task prompts and represent articles for each event as a fully connected graph; the learned policy and reward-trained controller remain frozen. Alongside LLM-based quality dimensions, we report ROUGE-1, ROUGE-2, and ROUGE-L to measure lexical agreement with reference summaries. Table 4 shows that CREW achieves an overall score of 3.84, outperforming the Orchestrator and SRW by 0.09 and 0.47 points, respectively. It also obtains the best Coverage, Logic, Relevance, ROUGE-1, and ROUGE-2 scores, while remaining competitive on ROUGE-L. The gains across semantic judgments and reference-based metrics suggest that the policy preserves useful collaboration behaviors when the document genre, output purpose, and evidence structure differ from scientific related work

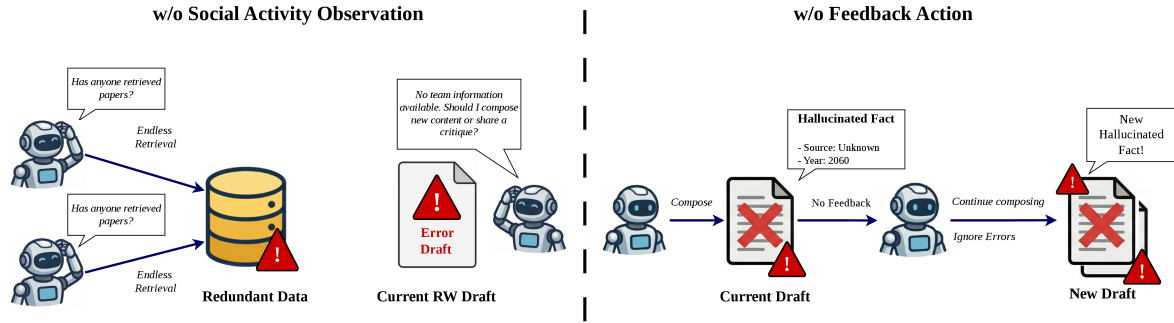


Figure 2: Ablation failure cases of CREW. Removing social activity observation leads to poor coordination and redundant agent behavior, while removing the feedback action prevents agents from correcting citation errors introduced by others, reducing both citation accuracy and logical coherence.

generation. Notably, these improvements require neither controller retraining nor reward adaptation, indicating that the learned action-selection strategy remains useful when the evidence units shift from scientific papers to news articles. This result broadens CREW’s potential applicability to other multi-document synthesis tasks requiring agents to coordinate evidence gathering, integration, and revision across heterogeneous sources.

Ablation Studies. We conduct ablation studies to examine the contribution of key components under the IPPO objective. As shown in Table 5, removing observation, reward, or action-space components generally degrades performance. Removing the draft-quality reward weakens the direct signal for improving the generated draft, causing agents to rely more on retrieval and feedback while performing fewer effective writing updates. Consequently, citation verification drops to nearly half of the full CREW system, suggesting that retrieved evidence is less effectively grounded in the final draft. Although the overall score slightly increases in this setting, the gain is marginal and comes at the cost of substantially lower citation reliability.

Figure 2 illustrates two representative failure cases. Without social activity observation, agents cannot track the action patterns of others, leading to redundant retrieval, overlapping behaviors, and weaker coordination; accordingly, Coverage, Logic, and Relevance decrease to around 3.30 on average. Removing the feedback action causes the largest degradation, with the Overall score dropping to 2.58, because agents lose an explicit mechanism for flagging hallucinated citations, missing evidence, and inconsistent claims. These results highlight the importance of both coordination-aware observation and feedback-driven revision in CREW.

5 Conclusion

We introduced CREW, a collaborative multi-agent reinforcement learning framework for automated related work generation. Instead of relying on a fixed pipeline, CREW allows multiple LLM agents to dynamically select among retrieval, knowledge dissemination, drafting, and critique actions according to a learned IPPO policy. Agents make decentralized decisions from their local observations while coordinating through shared knowledge, drafts, and feedback. This combination preserves agent autonomy without sacrificing the information exchange required for coherent scientific synthesis. Experiments on the OARELATEDWORK benchmark show that CREW improves generation quality and citation grounding over strong baselines while maintaining competitive efficiency. Comparisons with centralized and naive orchestrators indicate that effective collaboration depends on both the system structure and learned action selection. Ablations show that feedback, social-activity observations, and quality-oriented rewards are important for coordination, revision, and citation reliability. Transfer experiments demonstrate that the learned policy remains effective across different inference backbones and can adapt for broader multi-document synthesis without retraining. Scalability experiments show that additional agents can contribute complementary retrieval, writing, and review capacity, although gains must be balanced against inference cost and latency. Overall, CREW demonstrates that reinforcement-learned, decentralized coordination offers a flexible alternative to rigid LLM pipelines and provides a promising foundation for adaptive, citation-aware scientific writing assistants.

Limitations

Although CREW achieves strong empirical performance, several limitations remain. First, the current system operates solely over a fixed candidate pool rather than dynamically querying live scholarly databases, potentially missing newly published papers or relevant background literature. Second, our automated reward signals and LLM-dependent critique may not fully capture nuanced scholarly conventions, theoretical framing, or field-specific writing styles. Third, while the learned policy can be generalized to broader multi-document synthesis tasks beyond related work generation simply by modifying action prompts, our evaluation breadth across other domains is currently confined to a preliminary study on Multi-News. Fourth, due to resource constraints, our human evaluation remains relatively limited in scale, despite exhibiting strong alignment with LLM-as-a-judge assessments. Finally, while parallel agents reduce wall-clock latency in principle, practical deployments are still constrained by backbone LLM reasoning capabilities, API rate limits, and hardware availability. Future work will integrate dynamic scholarly search engines, expand human evaluation, and explore broader scientific domains.

Ethical Considerations

CREW is intended to assist researchers in drafting and organizing related work sections, not to replace human scholarly judgment. Because LLM-based systems can still generate unsupported claims, misinterpret cited work, or overstate relationships among papers, users should carefully verify all generated statements, citations, and comparisons before publication. The system should be used as a writing aid that accelerates evidence organization and revision, while final responsibility for accuracy, attribution, and intellectual contribution remains with the authors. There is also a risk that automated related work generation could encourage superficial literature review practices if users accept outputs without critical reading. To mitigate this risk, CREW emphasizes citation grounding and critique actions, but these safeguards do not eliminate the need for human review. In addition, evaluations based on LLM judges may reflect biases present in the judging models, including preferences for fluent writing or dominant research paradigms. We therefore recommend transparent reporting when such tools are used and encourage future work on

stronger provenance tracking, uncertainty estimation, and human-centered evaluation protocols.

References

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Nee-lakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Jingqiang Chen and Hai Zhuge. 2016. [Summarization of related work through citations](#). In *12th International Conference on Semantics, Knowledge and Grids, SKG 2016, Beijing, China, August 15-17, 2016*, pages 54–61. IEEE Computer Society.
- Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Rui Yan, Xin Gao, and Xiangliang Zhang. 2022. [Target-aware abstractive related work generation with contrastive learning](#). In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22*, page 373–383, New York, NY, USA. Association for Computing Machinery.
- Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao, and Rui Yan. 2021. [Capturing relations between scientific papers: An abstractive model for related work section generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6068–6077, Online. Association for Computational Linguistics.
- Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. [Adapting language models to compress contexts](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore. Association for Computational Linguistics.
- Zekun Deng, Zixin Zeng, Weiye Gu, Jiawen Ji, and Bolin Hua. 2021. [Automatic related work section generation by sentence extraction and reordering](#). In *Proceedings of the 1st Workshop on AI + Informetrics (AI+I2021) co-located with the iConference 2021, Virtual Event, March 17th, 2021*, CEUR Workshop Proceedings, pages 101–110. CEUR-WS.org.
- Martin Docekal, Martin Fajcik, and Pavel Smrz. 2024. [Oarelatedwork: A large-scale dataset of related work sections with full-texts from open access sources](#). *Preprint*, arXiv:2405.01930.
- Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. [Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.
- William Feller. 1968. *An Introduction to Probability Theory and Its Applications*, 3 edition, volume 1. John Wiley and Sons.

- Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. [Counterfactual multi-agent policy gradients](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Nianlong Gu and Richard Hahnloser. 2024. [Controllable citation sentence generation with language models](#). In *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, pages 22–37, Bangkok, Thailand. Association for Computational Linguistics.
- Cong Duy Vu Hoang and Min-Yen Kan. 2010. [Towards automated related work summarization](#). In *Coling 2010: Posters*, pages 427–435, Beijing, China. Coling 2010 Organizing Committee.
- Yue Hu and Xiaojun Wan. 2014. [Automatic generation of related work sections in scientific papers: An optimization approach](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633, Doha, Qatar. Association for Computational Linguistics.
- Gautier Izacard and Edouard Grave. 2021. [Leveraging passage retrieval with generative models for open domain question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.
- Shing-Yun Jung, Ting-Han Lin, Chia-Hung Liao, Shyan-Ming Yuan, and Chuen-Tsai Sun. 2022. [Intent-controllable citation text generation](#). *Mathematics*, 10(10).
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#). In *Advances in Neural Information Processing Systems*.
- Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. [Camel: Communicative agents for "mind" exploration of large language model society](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 51991–52008. Curran Associates, Inc.
- Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul M. B. Vitányi. 2004. [The similarity metric](#). *IEEE Transactions on Information Theory*, 50(12):3250–3264.
- Xiangci Li, Yi-Hui Lee, and Jessica Ouyang. 2024. [Cited text spans for scientific citation text generation](#). In *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, pages 90–104, Bangkok, Thailand. Association for Computational Linguistics.
- Xiangci Li and Jessica Ouyang. 2024. [Related work and citation text generation: A survey](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13846–13864, Miami, Florida, USA. Association for Computational Linguistics.
- J. Lin. 1991. [Divergence measures based on the shannon entropy](#). *IEEE Transactions on Information Theory*, 37(1):145–151.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *ACM Comput. Surv.*, 55(9).
- Xiaochuan Liu, Ruihua Song, Xiting Wang, and Xu Chen. 2025. [Select, read, and write: A multi-agent framework of full-text-based related work generation](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7009–7028, Vienna, Austria. Association for Computational Linguistics.
- Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith. 2021. [Explaining relationships between scientific documents](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2130–2144, Online. Association for Computational Linguistics.
- Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. [Key-value memory networks for directly reading documents](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.
- OpenRouter. 2026. OpenRouter API Documentation. <https://openrouter.ai/docs>. Accessed: May 24, 2026.
- John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. [High-dimensional continuous control using generalized advantage estimation](#). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. [Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 38154–38180. Curran Associates, Inc.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.
- J Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, Niall Williams, Yashas Lokesh, and Praveen Ravi. 2021. [Pettingzoo: Gym for multi-agent reinforcement learning](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 15032–15043. Curran Associates, Inc.
- Pancheng Wang, Shasha Li, Haifang Zhou, Jintao Tang, and Ting Wang. 2020. [Toc-rwg: Explore the combination of topic model and citation information for automatic related work generation](#). *IEEE Access*, 8:13043–13055.
- Yongzhen Wang, Xiaozhong Liu, and Zheng Gao. 2018. [Neural related work summarization with a joint context-driven attention mechanism](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1776–1786, Brussels, Belgium. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. [A prompt pattern catalog to enhance prompt engineering with chatgpt](#). *Preprint*, arXiv:2302.11382.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryan W White, Doug Burger, and Chi Wang. 2023. [Autogen: Enabling next-gen llm applications via multi-agent conversation](#). *Preprint*, arXiv:2308.08155.

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. [Memorizing transformers](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohen. 2022. [PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. [React: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

A Algorithmic and Methodological Details

This section provides algorithmic and methodological details for the proposed framework.

A.1 Pseudocode of the Proposed Framework

Appendix A.1 summarizes the full training, sampling, and inference procedure of CREW. Algorithm 1 shows how agents interact with shared artifacts over multiple rounds, collect rewards from artifact quality improvements, update the shared IPPO, and finally generate the related work draft through learned action sampling at inference time.

A.2 Reward Formulation

This appendix details our reward function, which defines component-wise quality improvement rewards for four artifacts: draft, local knowledge, global knowledge, and feedback. These rewards provide dense and interpretable learning signals that encourage agents to produce coherent, informative, and target-paper-relevant related work drafts. Draft-level improvement is measured against the human-written related work section, denoted as e_{gold} .

(i) **Draft quality** $\hat{Q}_D$ evaluates semantic relevance alongside a *compression-density* score, using lossless compression as a lightweight proxy for textual regularity (Li et al., 2004):

$$\hat{Q}_D = \gamma_{D,1} c(D, \text{gold}) + \gamma_{D,2} (1 - C(D)/|D|), \quad (16)$$

where $C(D)$ is the lossless compressed byte-length and $|D|$ is the original byte-length of the draft.

(ii) **Local quality** $\hat{Q}_L$ balances relevance and topic density against a redundancy penalty:

$$\hat{Q}_L = \gamma_{L,1} c(L_i, \text{gold}) + \gamma_{L,2} c(L_i, \text{tgt}) - \gamma_{L,3} \rho_i, \quad (17)$$

where $c(L_i, \text{tgt})$ evaluates how topically focused the readings are, and $\rho_i = \max_{h \in \mathcal{H}_i} c(L_i, h)$ penalizes semantic overlap with the agent’s recent reading history $\mathcal{H}_i$.

(iii) **Global quality** $\hat{Q}_G$ rewards alignment with the gold standard, semantic novelty relative to prior shared knowledge G_{t-1} , and lexical distributional similarity to the gold standard:

$$\hat{Q}_G = \gamma_{G,1} c(G, \text{gold}) + \gamma_{G,2} (1 - c(G, G_{t-1})) + \gamma_{G,3} (1 - \text{JS}(p_G \| p_{\text{gold}})), \quad (18)$$

where $\text{JS}(\cdot \| \cdot)$ measures lexical coverage of the global knowledge against the gold standard, since lower distributional divergence indicates that G covers a more similar set of unigram evidence (Lin, 1991).

(iv) **Feedback quality** $\hat{Q}_F$ measures whether the critique targets draft deficiencies, using the gap vector $\mathbf{g} = \mathbf{e}_{\text{gold}} - \mathbf{e}_D$:

$$\hat{Q}_F = c(\mathbf{e}_F, \mathbf{g}). \quad (19)$$

A.3 Joint Action and Credit Assignment

Joint Collaboration. When multiple agents simultaneously perform actions that modify shared resources, namely Disseminate, Critique, and Compose, our framework employs a segment-level coordination mechanism. Each agent selects the text segment it is most confident in editing and reports a confidence score $c_i \in [0, 1]$ in JSON format. Because agents have distinct observations shaped by their individual exploration trajectories, they naturally gravitate toward different segments, mirroring human collaboration in which authors refine different parts of a manuscript in parallel. The environment then merges these edits into a cohesive document. If multiple agents select the same segment, the agent with the highest confidence score c_i is granted editing rights.

Algorithm 1 CREW with Independent PPO

Input: Target paper p , paper database $\mathcal{P}$, citation graph $\mathcal{G}$, agents $\mathcal{A} = \{A_i\}_{i=1}^M$, horizon T

Output: Final related-work draft D_T

```
1: Initialize shared actor model  $\pi_\theta$  and shared critic model  $V_\phi$ .
2: while  $step < step_{max}$  do
3:   // Loop through training steps
4:   Initialize shared batch buffer  $\mathcal{W} \leftarrow \emptyset$ .
5:   while  $\mathcal{W}$  does not reach a predefined size do
6:     Create  $M$  empty episode caches  $E^1, \dots, E^M$ 
7:     Initialize shared states  $D_0, G_0, F_0, C_0 \leftarrow \emptyset$ 
8:     // Reset the environment
9:     Initialize local memory  $L_i^0 \leftarrow \emptyset$  for each  $A_i$ 
10:    for  $t = 1$  to  $T$  do
11:      for each agent  $A_i \in \mathcal{A}$  in parallel do
12:        Observe  $o_t^i = \text{OBSERVE}(D_{t-1}, G_{t-1}, F_{t-1}, C_{t-1}, L_{t-1}^i)$ 
13:        Sample  $a_t^i \sim \pi_\theta(\cdot | o_t^i)$ , where  $a_t^i \in \{\text{Retrieve}, \text{Disseminate}, \text{Critique}, \text{Compose}\}$ 
14:        if  $a_t^i = \text{Retrieve}$  then
15:           $(L_t^i, C_t) \leftarrow \text{RETRIEVE}(L_{t-1}^i, C_{t-1}, \mathcal{P}, \mathcal{G})$ 
16:        else if  $a_t^i = \text{Disseminate}$  then
17:           $G_t \leftarrow \text{DISSEMINATE}(L_t^i, G_{t-1})$ 
18:        else if  $a_t^i = \text{Critique}$  then
19:           $F_t \leftarrow \text{CRITIQUE}(D_{t-1}, L_t^i, G_{t-1})$ 
20:        else if  $a_t^i = \text{Compose}$  then
21:           $D_t \leftarrow \text{COMPOSE}(D_{t-1}, G_{t-1}, L_t^i, F_{t-1}, C_{t-1})$ 
22:        end if
23:         $r_t^i \leftarrow \text{REWARD}(D_t, G_t, F_t, C_t, L_t^i)$ 
24:        Store  $(o_t^i, a_t^i, r_t^i, \log \pi_\theta(a_t^i | o_t^i), V_\phi(o_t^i))$  into cache  $E^i$ 
25:      end for
26:    end for
27:    for each agent  $A_i \in \mathcal{A}$  do
28:      Append cache  $E^i$  into the shared batch buffer  $\mathcal{W}$ .
29:    end for
30:  end while
31:  // Stop sampling, start training
32:  Estimate advantages  $\hat{A}$  with GAE on the shared batch  $\mathcal{W}$ 
33:  Compute actor loss  $J_\pi(\theta)$  and critic loss  $J_V(\phi)$  on the shared batch  $\mathcal{W}$ 
34:  Update shared parameters  $\theta$  and  $\phi$  using Adam and gradient clipping
35: end while
36: Inference: At inference time, each agent samples its action from the learned shared policy,  $a_t^i \sim \pi_\theta(\cdot | o_t^i)$ , over  $T$  rounds, and the final draft  $D_T$  is returned.
```

We formalize the likelihood of such overlap using the Birthday Problem and Stirling’s approximation (Feller, 1968). Under a strictly uniform random policy, which serves as a worst-case baseline where agents select shared actions with probability $p = 3/4$ and choose segments uniformly, the unconditional probability of overlapping edits for M agents across K segments is approximated as:

$$P(\text{overlap}) \approx \sum_{n=2}^M \left[1 - \exp\left(-\frac{n(n-1)}{2K}\right) \right] \times \binom{M}{n} \left(\frac{3}{4}\right)^n \left(\frac{1}{4}\right)^{M-n}. \quad (20)$$

For example, with $M = 3$ agents and $K = 6$ segments, the overlap probability is only $\approx 25.78\%$, suggesting that parallel co-editing remains efficient with limited coordination overhead.

Credit Assignment. Once the joint draft is gener-

ated, it is evaluated to compute a global reward. To fairly distribute this shared reward among participating agents, we draw inspiration from the counterfactual multi-agent policy-gradient framework COMA (Foerster et al., 2018) and apply difference rewards. Formally, the difference reward for agent i is defined as:

$$D_i(z) = G(z) - G(z_{-i}), \quad (21)$$

where $G(z)$ is the global evaluation score of the fully composed text z , and $G(z_{-i})$ is the score of the text synthesized without the contributions of agent i .

B Dataset Construction and Citation Graph

To ensure a rigorous and reproducible evaluation of the generated related work sections, this appendix

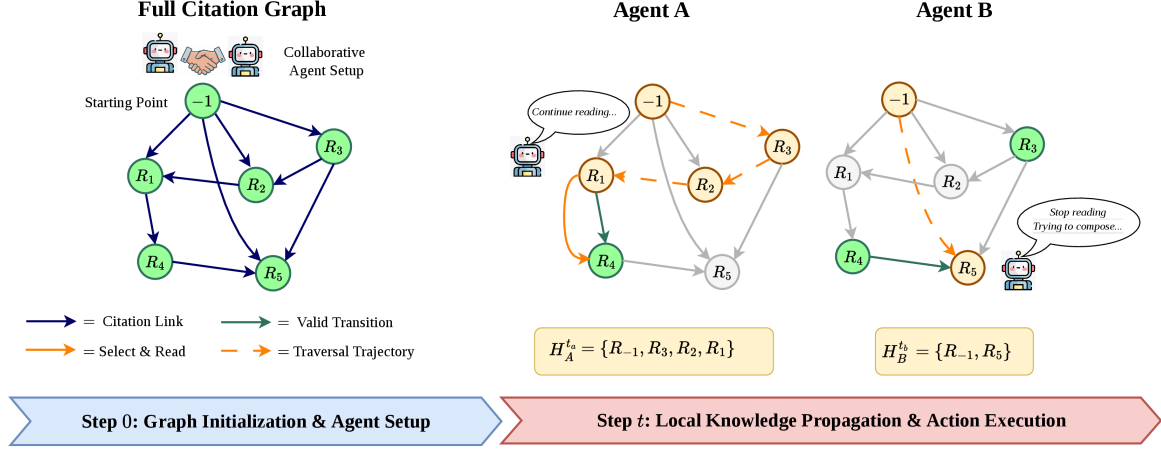


Figure 3: Citation graph construction and agent traversal in CREW. The target paper is assigned the sentinel ID -1 and linked to its cited references to form the working citation graph. Agents then traverse the graph under local observation, selecting reachable citations and incrementally updating their reading histories.

provides detailed information on the processing of the OARELATEDWORK dataset.

B.1 OARELATEDWORK Dataset

We leverage the OARELATEDWORK dataset, which contains 91,445 academic papers stored in JSONL format. Each paper record includes five fields: a unique integer id, title, abstract, related_work (the gold-standard related work section), and referenced (the list of cited paper IDs).

In our experiments, we split the dataset into two non-overlapping subsets for policy learning and experimental evaluation. Specifically, we use 2% of the papers for training and 0.4% of the papers for evaluation. The training subset is used to optimize both the actor and critic networks during PPO training: the actor learns agents’ action-selection policies, while the critic estimates value functions for evaluating state-action trajectories. The evaluation subset is kept disjoint from the training subset, ensuring that no target paper used for training appears in the test set.

B.2 Citation Graph Construction

From the OARELATEDWORK dataset, we construct a directed citation graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, as illustrated in Figure 3. The graph is initialized from the target paper, assigned the sentinel ID -1 , and its connected reference papers; agents then traverse valid citation transitions to select and read papers, producing agent-specific trajectories and local histories. We represent the graph as two adjacency lists:

- **Outgoing adjacency** (citation_adj_out): For each paper p_i , the list of papers it cites, i.e., $\mathcal{N}_{\text{out}}(p_i) = \{p_j \mid e_{i,j} \in \mathcal{E}\}$.
- **Incoming adjacency** (citation_adj_in): For each paper p_i , the list of papers that cite it, i.e., $\mathcal{N}_{\text{in}}(p_i) = \{p_j \mid e_{j,i} \in \mathcal{E}\}$.

During evaluation, a target paper P_{target} is selected from the dataset and assigned the sentinel ID -1 . Its outgoing citation edges are injected into the graph, and its direct neighbors (papers with at least 5 valid connections) serve as the reference pool $\mathcal{P} = \mathcal{N}_{\text{out}}(P_{\text{target}})$ from which the multi-agent system must read and synthesize. The gold-standard related_work field of P_{target} is used as the ground truth for evaluation.

C Evaluation metrics

C.1 LLM-as-a-judge

Evaluating long-form academic text generation is inherently complex, as traditional n -gram metrics fail to capture narrative flow, logical transitions, and citation discipline. We adopt an **LLM-as-a-Judge** protocol (using GPT-4o) scoring on a 1–5 Likert scale across three dimensions.

Coverage. Evaluates whether the generated related work comprehensively covers all key areas.

- *Score 1:* Limited coverage, touching small portions and lacking key areas.
- *Score 2:* Covers some parts but has noticeable omissions of significant areas.

- *Score 3*: Generally comprehensive but misses a few key points.
- *Score 4*: Covers most key areas comprehensively, with only minor topics left out.
- *Score 5*: Comprehensively covers all key and peripheral topics with detailed discussions.

Logic. Assesses the logical flow and structural coherence.

- *Score 1*: Lacks logic, no clear connections between sentences.
- *Score 2*: Weak logical flow, content arranged in a disordered manner.
- *Score 3*: Reasonable structure, though transitions could be improved (e.g., repeating).
- *Score 4*: Good logical consistency, natural transitions, only slightly rigid.
- *Score 5*: Tightly structured, logically clear, smooth transitions without redundancy.

Relevance. Evaluates focus on the target paper’s core subject.

- *Score 1*: Outdated or unrelated to the field, no alignment with the topic.
- *Score 2*: Somewhat on topic but with several digressions.
- *Score 3*: Generally on topic despite minor unrelated details.
- *Score 4*: Mostly on topic and focused, without frequent digressions.
- *Score 5*: Exceptionally focused, seamlessly contributing to core topic understanding.

C.2 Citation Verification

To complement LLM-based quality evaluation, we further assess whether the generated related work draft is grounded in the provided citation database. A high-quality draft should not only be coherent and relevant, but also cite the correct papers from the available reference pool without introducing unsupported or hallucinated references. We therefore use citation verification to measure the validity of citations produced in the generated draft.

Let $\hat{\mathcal{C}}(D)$ denote the set of paper IDs cited in a generated draft D , and let $\mathcal{P}_{\text{ref}}$ denote the valid reference pool associated with the target paper. We

define citation verification as the proportion of generated citations that can be correctly matched to valid papers in the reference pool:

$$CV(D) = \frac{|\hat{\mathcal{C}}(D) \cap \mathcal{P}_{\text{ref}}|}{\max(1, |\hat{\mathcal{C}}(D)|)}. \quad (22)$$

A higher $CV(D)$ indicates that the generated draft cites papers grounded in the provided reference pool, while a lower score suggests the presence of invalid, unsupported, or hallucinated citations.

D Implementation Details

In this section, we provide the comprehensive setup details required to reproduce our framework, including system configurations, specific hyperparameter choices for IPPO, and the network architectures.

D.1 System and Training Setup

Our behavioral policy is optimized using IPPO, driven by the open-weight **Qwen 2.5 72B Instruct** model to ensure robust local reasoning during centralized training. The training setup involves a shared Actor and Critic network (Parameter Sharing) for multi agents interacting globally. The training is conducted for 3 training epochs, with $K = 4$ PPO optimization epochs per update and an episode horizon (max rounds) of 15 steps per paper. Actions are capped to a context length of 6144 tokens, and GPU memory utilization is scaled dynamically. For inference to demonstrate policy transferability, we swap the generation engine with LLaMA 8B, Gemini 2.0 Flash, Qwen 2.5 72B Instruct, and GPT-4o. Table 6 summarizes the core IPPO hyperparameters, LLM settings, reward coefficients, and network configurations.

D.2 Network Architectures

The shared actor–critic layer configurations, including parameter counts, are detailed in Table 6. The observation vector contains 20 dimensions derived from graph features, draft quality, and social metrics.

D.3 Baselines Detailed Analysis

To accurately situate the performance of our framework within the current literature, we compare CREW against three groups of baseline paradigms: multi-agent methods, single-agent LLM methods, and abstractive supervised models.

Multi-Agent Methods. We include two multi-agent baselines. *Select, Read, and Write* (SRW) (Liu et al., 2025) is a specialized architecture for related-work generation that decomposes the task into selector, reader, and writer roles. For a fair comparison with our experimental budget, we run SRW for 15 rounds of select-read interactions, followed by one final write round to generate the related-work section. This setting preserves SRW’s intended multi-stage evidence-gathering workflow while matching the number of interaction rounds used in our experiments. We also compare against an *Orchestrator* baseline inspired by HuggingGPT (Shen et al., 2023). It keeps the same shared documents, agent pool, and action space as CREW but replaces the learned IPPO decision-making policy with a centralized LLM coordinator. The Orchestrator is run for the same 15 rounds as CREW and assigns actions to individual agents from the full system state. This baseline tests whether CREW’s gains come from reinforcement-learned decentralized coordination rather than simply from using multiple agents and shared working memory.

Single-Agent LLM Methods. We evaluate GPT-4o under two single-agent settings. In *GPT-4o (Long Context)*, the model receives the target paper context and the available reference information in a single prompt, then directly generates the related-work section. This setting represents a monolithic generation strategy that relies on the model’s long-context reasoning ability. In *GPT-4o (RAG)*, we first retrieve the most relevant chunks using embedding-based similarity and then prompt GPT-4o with the retrieved evidence, representing a standard retrieval-augmented search-and-generate pipeline.

Abstractive Models. PRIMERA (Xiao et al., 2022) is a supervised sequence-to-sequence model pre-trained for multi-document summarization. We adapt it to related-work generation on OARELATEDWORK, using it as an abstractive baseline that tests how far a specialized summarization model can go without explicit multi-agent reasoning, citation-graph exploration, or adaptive action selection.

D.4 System Hyperparameters and Configurations

We implemented CREW in Python 3.12.12 and used PettingZoo (Terry et al., 2021) to construct the multi-agent reinforcement learning environment.

Table 6: System hyperparameters and configurations used in CREW.

Category	Notation	Description	Value
RL	α_ω	Actor learning rate	3×10^{-4}
	α_ϕ	Critic learning rate	1×10^{-3}
	γ	Discount factor	0.99
	λ	GAE parameter	0.95
	ϵ	Clipping threshold	0.2
	K	PPO epochs	4
	c_e	Entropy coefficient	0.01
	T	Rollout steps	15
LLM	τ	Temperature	0.3
	p	Top- p sampling	0.9
	$N_{\max}$	Max output tokens	1400
Reward	$\gamma_{D,1}$	Draft quality coefficient	0.8
	$\gamma_{D,2}$	Draft quality coefficient	0.2
	$\gamma_{L,1}$	Local quality coefficient	0.5
	$\gamma_{L,2}$	Local quality coefficient	0.3
	$\gamma_{L,3}$	Local quality coefficient	0.2
	$\gamma_{G,1}$	Global quality coefficient	0.5
	$\gamma_{G,2}$	Global quality coefficient	0.3
	$\gamma_{G,3}$	Global quality coefficient	0.2
Network	π_1	Policy hidden layer 1	672
	π_2	Policy hidden layer 2	1056
	π_3	Policy output layer 3	132
	V_1	Value hidden layer 1	672
	V_2	Value hidden layer 2	1056
	V_3	Value output layer 3	33

All actor-critic components were trained locally on a workstation equipped with an NVIDIA RTX A5000 GPU with 24GB of GDDR6 memory. LLM-based generation and evaluation were conducted through the OpenRouter API (OpenRouter, 2026), which provides a unified interface for accessing different language model backends. Unless otherwise specified, all baselines and ablation studies were run under the same software and hardware environment. Table 6 summarizes the system hyperparameters and configurations used in CREW, including IPPO optimization settings, LLM decoding parameters, and reward coefficients.

E Scalability Analysis

Multi-Agent Collaboration. To further analyze the effect of agent collaboration, we conduct an additional experiment by varying the number of agents from 1 to 5 while keeping the same backend language model for all settings. This ensures that the comparison reflects the contribution of the multi-agent organization rather than differences in model capability. For each configuration, we evaluate the generated related work quality, execution time, and total token consumption. As shown in Table 7, increasing the number of agents generally improves the quality scores, especially in logic and citation quality, since different agents can specialize in retrieval, feedback, knowledge updating, and writing. Meanwhile, execution time only increases moderately because agents can per-

Table 7: Impact of scaling the number of collaborative agents on generation quality, citation grounding, and token consumption.

Number of Agents	LLM-based Evaluation (Avg.)				Citation	Cost Efficiency	
	Cov. $\uparrow$	Logic $\uparrow$	Rel. $\uparrow$	Overall $\uparrow$	Ver.(%) $\uparrow$	Tokens $\downarrow$	Time(s) $\downarrow$
1	3.04	3.36	3.82	3.41	86.10	42.26K	359
2	3.41	3.69	4.27	3.79	100.00	90.64K	778
3	3.31	3.33	4.17	3.79	96.20	121.81K	644
4	3.44	3.73	4.27	3.82	99.33	198.74K	755
5	3.48	3.77	4.36	3.87	99.17	246.49K	833

form several actions in parallel, although total token usage grows with the number of agents. These results suggest that multi-agent collaboration provides a better trade-off between generation quality and computational cost than a single-agent setup.

Discussion. Although parallel agents should ideally keep wall-clock time nearly stable, inference time increases mildly and then stabilizes in Table 7. This is largely due to OpenRouter API latency, which is external to our system. At timestep t , the system waits for the slowest agent, $T_t = \max_{1 \leq i \leq M} \tau_i^{(t)}$. If each agent independently experiences a delayed API response with probability p (Feller, 1968), and response time is T_{wait} under delay and T_{normal} otherwise, then $\mathbb{E}[T_t] = T_{\text{wait}} - (T_{\text{wait}} - T_{\text{normal}})(1-p)^M$. Therefore, $\lim_{M \rightarrow \infty} \mathbb{E}[T_t] = T_{\text{wait}}$ for $0 < p \leq 1$. Thus, API variability may raise latency, but the expected timestep duration remains bounded rather than growing unboundedly with the number of agents.

F Qualitative Experiments

F.1 Human Evaluation

To complement the automatic evaluation, we conducted a human evaluation on 20 papers randomly selected from the Artificial Intelligence domain. Following the evaluation protocol of SRW (Liu et al., 2025), three graduate students performed pairwise comparisons between outputs generated by CREW and SRW. The evaluators assessed each pair in terms of coverage, logic, relevance, and overall quality. CREW achieved win rates of 53% for coverage, 60% for logic, and 57% for relevance, yielding an average overall win rate of 56.7%. These results indicate that human judges consistently preferred CREW to SRW across all evaluation dimensions. This human preference closely aligns with the automatic comparison in Table 1, jointly demonstrating the effectiveness of CREW’s

learned coordination in producing higher-quality related work; the qualitative example in Section F.2 further supports this conclusion by showing that CREW provides more citation-grounded synthesis and clearer connections between prior studies and the target paper than the baselines.

F.2 Case Study

We use a challenging target paper on tweet wikification to illustrate how CREW handles short-text entity linking, collective inference, graph-based semi-supervision, and low-label training while grounding claims in the provided citation pool. Figure 4 shows the target paper, and Figure 5 compares outputs from CREW and the baselines.

G Prompt Templates

We present the detailed prompt templates that drive each agent action. The RETRIEVE action is further organized into two sequential phases: *paper selection*, which selects candidate papers related to the target work, and *paper reading*, which reads the selected papers and extracts useful evidence for later writing. Other prompts are designed following common prompt-engineering practices (Brown et al., 2020; Wei et al., 2022; Yao et al., 2023; White et al., 2023; Liu et al., 2023), including role specification, explicit task instructions, step-by-step task decomposition, and output format constraints. These designs help reduce ambiguity and make the generated outputs more stable and easier to evaluate (Liu et al., 2025).

Target Paper for Qualitative Case Study

Title. *Collective Tweet Wikification based on Semi-supervised Graph Regularization*

Abstract. Wikification for tweets aims to automatically identify each concept mention in a tweet and link it to a concept referent in a knowledge base such as Wikipedia. Due to the shortness of a tweet, a collective inference model incorporating global evidence from multiple mentions and concepts is more appropriate than a non-collective approach that links each mention independently... To identify semantically related mentions for collective inference, it detects meta path-based semantic relations through social networks. Compared with a strong supervised baseline trained from 100% labeled data, the proposed approach achieves comparable performance with 31% labeled data and obtains a 5% absolute F1 gain with 50% labeled data.

Figure 4: Qualitative case-study target paper. The framed title and abstract of the target paper.

GPT-4o (RAG)

The task of wikification for tweets, which involves linking concept mentions in tweets to corresponding entries in a knowledge base, shares similarities with various challenges in entity linking and relation extraction... Distant supervision, as explored in studies [110538048, 26702637, 5884944], utilizes existing knowledge bases to generate training data by aligning known facts with text. However, this approach often introduces noise due to incorrect assumptions about entity relationships, necessitating methods to mitigate this issue, such as the label-free distant supervision method proposed in [26702637]... Overall, the intersection of distant supervision, collective inference, and context expansion techniques provides a robust framework for addressing the challenges of wikification in tweets.

GPT-4o (Long Context)

... Local methods, such as those employed by SemTag (Dill et al., 2003) and Wikify! (Mihalcea and Csomai, 2007), focus on resolving each mention independently by considering the similarity between a mention and an entity... Recent advancements have focused on semi-supervised and unsupervised learning techniques to mitigate the reliance on large labeled datasets. For instance, semi-supervised methods, such as those explored by Druck et al. (2008) and Bellare et al. (2009), incorporate unlabeled ...In the domain of semi-supervised learning, techniques such as graph-based label propagation (Wang and Cohen, 2007) and posterior regularization (Ganchev et al., 2010) have shown promise in effectively utilizing both labeled and unlabeled data. These methods are particularly advantageous in scenarios where ...

PRIMERA

Collective Tweet Wikification based on Semi-supervised Graph Regularization. Wikification for tweets aims to automatically identify each concept mention in a tweet and link it to a concept referent in a knowledge base, e.g., Wikipedia. In order to identify semantically related mentions for collective inference, we detect meta path-based semantic relations through social networks. Compared to the strong supervised baseline trained from 100% labeled data, our proposed approach achieves comparable performance with 31% labeled information and obtains 5% absolute F1 gain with 50% labelled data.

Select, Read and Write (SRW)

...One notable approach is the use of global coherence, where the relationships between multiple mentions...For instance, the work in "Efficient Collective Entity Linking with Stacking" [5310325] proposes a fast collective disambiguation method that combines local and global... Similarly, "Local and Global Algorithms for Disambiguation to Wikipedia" [7610609] analyzes the effectiveness of global...In summary, the integration of global evidence, context expansion, semi-supervised learning, graph-based methods, and collaborative ranking represents a promising direction...

CREW

The field of entity linking, particularly in the context of short texts like tweets, has seen significant advancements through the integration of collective inference models and graph-theoretic methods.... Efficient collective entity linking, as proposed by [5310325], addresses the challenge of disambiguating entity mentions in text by leveraging a stacking approach that combines local and global evidence.... Similarly, [9854940] explores the use of graph-theoretic methods, specifically centrality and regularization, to enhance class-instance label propagation. The work in [7610609] further contributes to this area by comparing local and global algorithms for disambiguation to Wikipedia. This study highlights the importance of utilizing the Wikipedia link structure to ensure coherent disambiguation, although it notes that local disambiguation remains a strong baseline. Our proposed semi-supervised graph regularization model builds upon these foundational works by incorporating both local and global evidence from multiple tweets through fine-grained relations, and by utilizing meta path-based semantic relations from social networks to identify semantically related mentions Our proposed semi-supervised graph regularization model builds upon these foundational works by incorporating both local and global evidence....

Figure 5: Case study comparing related work generation outputs from GPT-4o (Long Context), GPT-4o (RAG), Primera, SRW, and CREW. Red highlights mark problematic content such as unsupported citations, weak synthesis, target-abstract copying, or list-like summaries. Green highlights show CREW's citation-grounded synthesis and its ability to connect prior works to the target paper.

Retrieve Action Prompt: Phase 1 – Paper Selection

System Prompt: You are a research worker with excellent paper reading skills.

User Prompt: You are helping write the Related Work section of a scientific paper. Your task is to choose the next paper section to read that adds the most new knowledge, not more of the same.

Inputs: Citation graph: {graph}; current paper (id=-1 = paper being written): {paper_details}; cited papers: {cited_papers}; citing papers: {citing_papers}; working memory: {working_memory}; reading history: {reading_history}; team global knowledge: {global_knowledge}.

Mandatory Reasoning: (1) Coverage audit: list research themes already covered across your memory and team global knowledge, or write “Coverage: empty.” (2) Gap identification: list 2–3 themes not yet covered, considering different method types, modalities, problem formulations, time periods, and contrastive approaches. (3) Candidate scoring: score unread candidates HIGH if their abstract addresses a gap and LOW if their themes are already covered. (4) Selection: pick the highest-scoring HIGH candidate; if all are LOW, choose the one least similar to global knowledge.

Hard Rules: Never select a paper whose themes are more than 60% covered in global knowledge. Never select a paper whose abstract overlaps more than 60% with reading history. If you have read 2+ papers on the same technique, switch clusters. The section must exist exactly in the paper’s structure list: abstract or related_work. Never repeat a (paper_id, section) pair from reading history. Prefer related_work over abstract when available.

Output Format: Return one JSON object only: {{"id": <paper_id>, "section": "<section_name>", "rationale": "<which gap this fills and why it is not already in global knowledge>"}}.

Figure 6: Prompt template for the RETRIEVE action, Phase 1: Paper Selection.

Retrieve Action Prompt: Phase 2 – Paper Reading

System Prompt: You are a research worker with excellent paper reading skills.

User Prompt: You are updating a short working memory (≤ 4096 tokens) for writing the Related Work section.

Inputs: Current paper: {cur_paper}; cited papers: {cited_paper}; just-read section: {section} of paper {paper_id}; verbatim content: {content}; reading history: {reading_history}; previous memory: {memory}; current draft: {draft}; global knowledge: {global_knowledge}.

Novelty Gate: Before writing, state in one sentence the primary contribution of paper {paper_id}. Then check whether this contribution already appears in memory, draft, or global knowledge. If it is more than 60% overlapping, use minimal mode; if genuinely new, use full extraction.

Extraction Rules: Minimal mode extracts one distinctive sentence with [ID]. Full extraction extracts 2–4 sentences covering the new method, dataset, finding, contrast, or limitation. Preserve only claims grounded in the provided content and cite using the paper ID in square brackets.

Output Requirements: Return the updated working memory as plain text. No JSON and no markdown.

Figure 7: Prompt template for the RETRIEVE action, Phase 2: Paper Reading.

Critique Action Prompt

System Prompt: You are a researcher worker with excellent knowledge synthesis skills, specializing in providing constructive academic feedback.

User Prompt: Synthesize feedback insights from your Local Knowledge and integrate them into the existing Shared Feedback to create a unified, high-quality feedback reference for the Related Work draft.

Inputs: Current draft: {draft}; local knowledge: {local_knowledge}; global knowledge base: {global_knowledge}; current shared feedback: {current_feedback}.

Evaluation Dimensions: If current feedback is empty, generate a comprehensive feedback outline evaluating coverage, logic, relevance, diversity, and citation quality. For each dimension, score 1–5 and provide specific edits. Diversity is critical: check whether multiple methodologies, paradigms, and viewpoints are represented, and flag over-reliance on one cluster or technique. Citation quality checks unsupported claims, citation dumping, invented IDs, and whether citations are anchored inline.

Synthesis Rules: Merge overlapping critiques, lead with the highest-impact problems, make every critique actionable with a concrete fix or suggested rewrite, and resolve contradictory prior feedback by keeping the stricter version.

Output Requirements: Return the updated synthesized feedback as plain text. No JSON and no markdown.

Figure 8: Prompt template for the CRITIQUE action.

Compose Action Prompt:

System Prompt: You are a research worker with excellent paper writing skills.

User Prompt: Finalize the Related Work section of a scientific paper.

Target Context: Paper abstract: {abstract}; cited papers: {cited_paper}.

Research Intelligence: Local knowledge: {local_knowledge}; global knowledge: {global_knowledge}.

Collaboration: Existing draft: {draft}; peer feedbacks: {feedbacks}.

Strict Instructions: Group studies by theme rather than summarizing papers in isolation. Use only paper IDs from {cited_paper}; never invent IDs or use numeric indices like [1]. Ensure every claim is supported by provided knowledge. Use formal academic tone and limit length to approximately 500–700 words. If a draft exists, refine it for consistency and feedback rather than rewriting unnecessarily. Every paragraph must start with a clear thematic topic sentence.

Output Format: Return only a JSON object: {{ "related_work": "..."} }.

Figure 9: Prompt template for the COMPOSE action.

Disseminate Action Prompt

System Prompt: You are a researcher worker with excellent knowledge synthesis skills, capable of organizing complex information into clear and logical insights.

User Prompt: Synthesize new insights from your Local Cache and integrate them into the existing Shared Knowledge Base to create a unified, high-quality reference for the entire agent network.

Inputs: Current shared knowledge base: {shared_knowledge}; new information from local cache: {local_cache}.

Quality Rules: If shared knowledge is empty, first generate a professional thematic outline and then append synthesized content into the outline. Do not simply append; merge overlaps, enrich existing points, and place new information logically. Group related concepts, explain relationships, and highlight contrasts using integrative phrasing such as “While [ID1] focuses on X, [ID2] addresses Y by ...”. Resolve contradictions when possible or document divergent views with citations.

Citation Rules: Preserve citations using only paper IDs in square brackets. Cite an ID only when the claim is explicitly supported by Local Cache or current Shared Knowledge. Do not invent claims during synthesis. Citations must appear in the same sentence immediately after supported claims; do not place large citation lists at paragraph ends. Remove untraceable citations.

Output Requirements: Return the updated Shared Knowledge as plain text. No JSON and no markdown.

Figure 10: Prompt template for the DISSEMINATE action.

Long Context Synthesis Prompt

System Prompt: You are an expert researcher with excellent paper writing skills.

User Prompt: You are writing the Related Work section of a scientific paper.

Target Context: Paper abstract: {abstract}.

Neighbor Abstracts: {neighbor_abstracts}.

Strict Instructions: Synthesize abstracts into a professional Related Work section. Group studies by theme and do not summarize papers in isolation. Use only paper IDs from the provided list in square brackets. Maintain a formal academic tone and limit length to approximately 600–800 words. Every paragraph must start with a clear thematic topic sentence.

Output Format: Return only a JSON object: {{ "related_work": "..."} }.

Figure 11: Prompt template for long-context synthesis.

Orchestrator Prompt

System Prompt: You are a senior research coordinator managing a team of AI research agents that collaboratively write a Related Work section. At each round, assign exactly one action to each agent to coordinate the team efficiently.

Available Actions: SEARCH reads a new paper and updates local memory; UPDATE synthesizes local memory into the shared global knowledge; WRITE drafts or refines the Related Work section; FEEDBACK reviews the current draft and provides constructive revision suggestions.

Inputs: Round index: {round_num}/{max_rounds}; target title: {paper_title}; target abstract: {abstract}; current draft preview: {draft_preview}; global knowledge preview: {global_preview}; shared feedback preview: {feedback_preview}; agent states: {agent_states}; recent action history: {action_history}.

Coordination Rules: Promote action diversity across agents; prioritize SEARCH and UPDATE when knowledge is sparse; use WRITE once sufficient evidence has been gathered; alternate WRITE and FEEDBACK when a draft exists; avoid assigning actions that are ineffective for an agent's current local memory; balance work so that all agents contribute.

Phase Guidance: In early rounds, build coverage through searching and updating. In middle rounds, balance additional evidence gathering with writing. In late rounds, focus on writing and feedback to polish the draft.

Output Format: Return only valid JSON with a brief reasoning field and an assignments object mapping each agent ID to {"action": "<SEARCH|UPDATE|WRITE|FEEDBACK>", "rationale": "<why>"}. Do not output any extra text.

Figure 12: Prompt template for the centralized orchestrator baseline.