

ERGA: An Explorer–Refiner Game with Backdoor-Adjusted Verdict Aggregation for LLM-based Claim Verification

Wanlong Yu, Feng Liu, Zhiliang Tian*, Zhen Huang*

Yonghui Wang, Jingyuan Huang, Dongsheng Li

PDL Lab, College of Computer Science and Technology,
National University of Defense Technology

{ ywl, richardlf, tianzhiliang, huangzhen, wyh_, jingyuanhuang, dsli }@nudt.edu.cn

Abstract

LLM-based fact-checking verifies complex claims through decomposition, retrieval, and reasoning aggregation. Existing methods expand verification inputs to cover more factual conditions, but added fragments can be redundant or misleading, compete with decisive evidence, and bias predictions toward surface relevance. Debiasing methods mitigate this bias by estimating shortcut effects from claim-side cues or biased signals in constructed inputs. However, because they calibrate verifier predictions after input construction, they are not designed to repair missing or noisy verification states. Thus, complex fact-checking should balance factual coverage and noise control during verification-state construction. We propose ERGA, an Explorer–Refiner Game with Backdoor-Adjusted Verdict Aggregation. ERGA formulates verification as editable state construction, initializing the state from decomposed verification units, retrieved evidence, and local analyses. It then runs an adversarial Explorer–Refiner game over this state: the Explorer strengthens under-covered factual signals, while the Refiner merges redundancy and separates weak or misleading fragments. Through this game interaction, ERGA obtains a refined state with a better coverage–noise balance and a removed fragment record. For verdict aggregation, ERGA uses the removed-fragment record to condition verifier predictions and aggregates them in a backdoor-adjusted form to reduce residual shortcut effects. Experiments on HoVer and FEVEROUS-S show that ERGA outperforms strong baselines.

1 Introduction

Automated fact-checking aims to determine the veracity of a given claim by retrieving, organizing, and analyzing relevant external evidence (Zeng

et al., 2021; Guo et al., 2022; Chen et al., 2024). This task provides a key technical foundation for identifying misinformation and mitigating the spread of false information online (Lazer et al., 2018). Recent progress in large language models (LLMs) has strengthened their capabilities in semantic understanding (Pan et al., 2026), natural-language reasoning, and generative analysis (Pan et al., 2024), leading to their growing integration into automated fact-checking systems (Zhao et al., 2026b; Wei et al., 2022). LLM-based fact-checking systems often combine claim decomposition, external evidence retrieval, and reasoning aggregation to handle complex claims, making them well suited to multi-hop fact-checking scenarios (Vykopal et al., 2024).

Existing LLM-based fact-checking methods commonly make complex-claim verification explicit by constructing structured intermediate representations before producing the final verdict (Guo et al., 2022; Eldifrawi et al., 2024; Zeng et al., 2021). These methods typically decompose or structure the original claim into smaller verification units, collect relevant information for these targets, perform local reasoning, and aggregate the local results into a final verdict. Representative approaches use such representations to specify what should be verified and to guide evidence collection, local reasoning, and verdict aggregation (Pan et al., 2023; Wang and Shu, 2023; Hu et al., 2025a; Zhao et al., 2024; Liu et al., 2025; Jeon and Lee, 2025; Pham et al., 2025). This design improves information coverage by allowing different parts of a complex claim to be examined with targeted information collection and local analysis. However, broader information coverage also brings more retrieved evidence and LLM-generated content into the verification process. Since retrieval and generation often favor semantic relatedness, some added content may appear relevant to the claim while offering limited or misleading evidential value (Schuster

*Corresponding Authors

et al., 2021; Zheng et al., 2024; Hu et al., 2025c). Once such content participates in later reasoning, it may compete with decisive information and bias the final prediction toward surface relevance.

To mitigate the prediction bias caused by the above issues, another line of work studies debiasing methods for fact verification. These methods operate on constructed verification inputs. They aim to calibrate the final verdict by reducing shortcut cues and spurious claim–evidence correlations (Jeon et al., 2023). A common strategy is to make the shortcut dependence measurable before predicting the verdict. These methods estimate shortcut effects from claim-side cues, annotation artifacts, language bias, or spurious reasoning paths. These estimates are then used to calibrate the final verifier prediction for less shortcut-sensitive evidence and reasoning signals (Schuster et al., 2019; Lee et al., 2021; Tian et al., 2022; Xu et al., 2023; Zhang et al., 2024a; Zhao et al., 2026a). However, this correction is constrained by the constructed input itself. If key information is missing or weak content has already been included, debiasing can only adjust the final prediction over this imperfect input. Thus, the remaining issue is not only how to debias the verifier but also how to construct a verification input that is sufficiently informative and less noisy.

These observations suggest that complex fact-checking should jointly address information coverage and noise control during verification-state construction. Expanding the verification input can recover missing information, but it may also introduce weak or misleading fragments. Once these fragments enter later reasoning, they can distort local evidence integration and become difficult to correct through final-stage debiasing alone. A more suitable framework should therefore recover missing information while separating weakly relevant fragments before they accumulate in later reasoning and affect the final verdict.

In this paper, we propose ERGA, an Explorer–Refiner Game with Backdoor-Adjusted Verdict Aggregation for LLM-based complex fact-checking. To jointly improve information coverage and control noise during verification, ERGA formulates complex fact-checking as a verification-state construction problem. The core idea is to use an adversarial state-editing game to balance two competing objectives: expanding the state to recover missing factual information and refining the state to separate weakly relevant or misleading fragments. Specifically, ERGA decomposes a

complex claim into editable verification units that serve as local targets for later expansion and refinement. The Explorer–Refiner Game then assigns two role-specialized objectives over the shared verification state: the Explorer strengthens undercovered factual signals, while the Refiner separates weak or misleading fragments that could distort the verdict. Through this adversarial interaction, the state is refined toward a better coverage–noise balance, producing a refined verification state S^* and a removed-fragment set R . Finally, Backdoor-Adjusted Verdict Aggregation uses the removed-fragment set R as the adjustment set, samples adjustment conditions from R , and aggregates conditional verifier predictions to approximate backdoor marginalization and mitigate residual spurious correlations. Experiments on complex fact-checking benchmarks, including FEVEROUS-S and HoVer, show that ERGA outperforms all baselines (Aly et al., 2021; Jiang et al., 2020).

Our contributions are threefold. (1) We propose ERGA, a coverage-and-noise-control framework for LLM-based complex fact-checking that uses an Explorer–Refiner Game to dynamically balance verdict-relevant evidence coverage and noisy-fragment separation during verification-state construction. (2) We propose Backdoor-Adjusted Verdict Aggregation, which treats the removed fragments separated by the Refiner as adjustment conditions to mitigate residual shortcut bias in final verdict prediction. (3) Experiments on HoVer and FEVEROUS-S datasets show that ERGA achieves SOTA performance against strong baselines.

2 Related Work

2.1 Fact-Checking Tasks and Benchmarks

Automated fact-checking is commonly formulated as evidence-grounded claim verification, where a system retrieves evidence and predicts a veracity label (Zeng et al., 2021; Guo et al., 2022). FEVER established the retrieval-plus-verdict benchmark, while HoVer and FEVEROUS extended fact verification to multi-hop reasoning and structured Wikipedia evidence (Thorne et al., 2018; Jiang et al., 2020; Aly et al., 2021). Early systems study evidence retrieval, claim–evidence classification, listwise reasoning, and graph-based evidence aggregation, forming the basis for later LLM-based verification frameworks (Soleimani et al., 2020; Jiang et al., 2021; Zhou et al., 2019; Liu et al., 2020).

2.2 LLM-Based Fact-Checking Frameworks

Recent LLM-based fact-checking frameworks commonly construct structured intermediate representations before producing the verdict (Zhang et al., 2026). Program- and logic-guided methods express verification as reasoning programs, logical clauses, or strategy-guided programs (Pan et al., 2023; Wang and Shu, 2023; Hu et al., 2025a). Claim decomposition is also widely used to turn complex claims into local verification targets, but recent analyses show that its benefit depends on whether the added subclaims and retrieved content improve useful evidence coverage or introduce extra noise (Hu et al., 2025c). Planning, agent, and retrieval-reasoning frameworks organize verification through multi-step planning, iterative search, self-reflection, or coordinated agents (Zhao et al., 2024; Xie et al., 2025; Hu et al., 2025b). Other systems improve reliability with logical or causal consistency, claim-vagueness handling, or graph-based claim-evidence structures (Ma et al., 2025; Liu et al., 2025; Huang et al., 2025; Jeon and Lee, 2025; Pham et al., 2025). Adjacent game-theoretic or equilibrium-based LLM reasoning methods coordinate reasoning paths or agents before answer selection (Zhang et al., 2024b; Yi et al., 2025; Pan et al., 2025; Tian et al., 2025). Overall, existing frameworks organize verification targets, evidence, or reasoning traces, while ERGA edits a shared verification state to recover missing information and separate noisy fragments before verdict aggregation.

2.3 Debiasing and Causal Reasoning for Verification

Debiasing work shows that fact verifiers can exploit annotation artifacts or claim-only cues instead of grounded claim-evidence support (Schuster et al., 2019). Subsequent studies reduce shortcut cues or spurious correlations with contrastive augmentation, causal intervention, or counterfactual inference (Lee et al., 2021; Tian et al., 2022; Xu et al., 2023). Causal methods further extend this view to multi-hop reasoning paths and broader LLM reasoning chains, as in Causal Walk, dynamic counterfactual debiasing, and DeCoT (Zhang et al., 2024a; Zhao et al., 2026a; Wu et al., 2024). Overall, these methods primarily reduce shortcut or bias signals after a claim-evidence input or reasoning representation has been formed, whereas ERGA records fragments separated during verification-state con-

struction and constructs adjustment conditions for verdict aggregation.

3 Methods

3.1 Overview

Our proposed method **ERGA** (Fig. 1) consists of three parts, with the complete inference workflow summarized in App. C. (1) **Decomposition-Guided State Initialization** (Sec. 3.2) builds the initial verification state S_0 by decomposing the claim into atomic verification units, retrieving candidate evidence, and generating local analyses. (2) **Explorer-Refiner Game for Coverage-Noise Balancing** (Sec. 3.3) updates S_0 through an Explorer-Refiner editing game, where Explorer improves coverage of missing factual conditions and Refiner merges redundancy or separates weak and potentially misleading fragments. This module outputs the refined state S^* and the removed-fragment set R . (3) **Backdoor-Adjusted Verdict Aggregation** (Sec. 3.4) predicts the final veracity label from S^* and uses adjustment conditions derived from R to reduce bias during verdict aggregation.

3.2 Decomposition-Guided State Initialization

To provide the Explorer-Refiner Game with a structured starting point, we initialize the verification state by decomposing the input claim into subclaims and grounding each sub-claim with retrieved evidence and an evidence-aware local analysis. Since a complex claim often involves multiple factual conditions, each sub-claim serves as a local verification unit that can be checked, expanded, or refined in later game rounds (Hu et al., 2025c). Given an input claim C , an LLM-based decomposer D generates a set of sub-claims:

$$\mathcal{C} = D(C) = \{c_1, \dots, c_n\}. \quad (1)$$

Here, c_i denotes the i -th sub-claim in $\mathcal{C}$ and specifies one local factual condition derived from C . For each sub-claim c_i , we use a modular retriever Ret to retrieve top- K candidate evidence pieces from the external corpus:

$$E_i = \text{Ret}(c_i) = \{e_{i,1}, \dots, e_{i,K}\}. \quad (2)$$

We instantiate Ret with BM25-based sparse retrieval (Robertson and Zaragoza, 2009). Next, an LLM-based local analyzer Ana generates an evidence-aware analysis based on the original claim, the sub-claim, and its candidate evidence:

$$a_i = \text{Ana}(C, c_i, E_i). \quad (3)$$

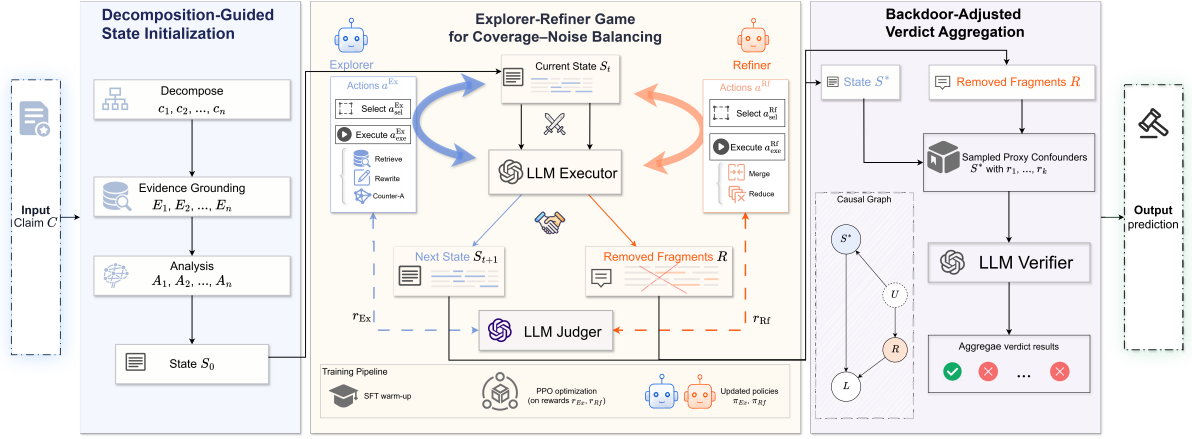


Figure 1: Overview of ERGA. It performs a three-stage verification process from left to right: (1) **Decomposition-Guided State Initialization** constructs the initial verification state S_0 from decomposed sub-claims, retrieved evidence, and local analyses; (2) **Explorer-Refiner Game for Coverage-Noise Balancing** alternately updates S_0 with role-specific Explorer and Refiner policies to obtain the refined state S^* and removed-fragment set R ; and (3) **Backdoor-Adjusted Verdict Aggregation** uses R as adjustment conditions for verifier prediction and aggregates the conditional verdicts into the final label.

The analysis a_i summarizes whether the current evidence supports, refutes, or is insufficient to verify the corresponding sub-claim. Finally, we combine the claim, sub-claims, candidate evidence, and local analyses into the initial verification state:

$$S_0 = (C, \{(c_i, E_i, a_i)\}_{i=1}^n). \quad (4)$$

This state is then passed to the Explorer-Refiner Game for further editing.

3.3 Explorer-Refiner Game for Coverage-Noise Balancing

To balance factual coverage and noise control, ERGA models verification-state refinement as a finite-horizon adversarial state-editing game over the initial verification state S_0 . Explorer and Refiner learn role-specific edit policies over the same evolving state: Explorer strengthens under-covered factual signals, while Refiner constrains expansion by merging redundancy and separating weakly relevant or misleading fragments. We next detail the game through its state transitions, structured edit actions, role-specific rewards, and policy learning procedure.

3.3.1 Game Formulation

Formally, the interaction is defined as a finite sequential game:

$$\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}^{\text{Ex}}, \mathcal{A}^{\text{Rf}}, \Phi^{\text{Ex}}, \Phi^{\text{Rf}}, r^{\text{Ex}}, r^{\text{Rf}} \rangle. \quad (5)$$

Here, $\mathcal{N} = \{\text{Ex}, \text{Rf}\}$ is the player set, $\mathcal{S}$ is the verification-state space, $\mathcal{A}^{\text{Ex}}$ and $\mathcal{A}^{\text{Rf}}$ are the action

spaces, Φ^{Ex} and Φ^{Rf} are state-editing functions, and $r^{\text{Ex}}, r^{\text{Rf}}$ are player-specific rewards. At round t , Explorer first edits the current state $S_t \in \mathcal{S}$:

$$S'_t = \Phi^{\text{Ex}}(S_t, a_t^{\text{Ex}}), \quad a_t^{\text{Ex}} \in \mathcal{A}^{\text{Ex}}. \quad (6)$$

Refiner then edits the intermediate state:

$$S_{t+1} = \Phi^{\text{Rf}}(S'_t, a_t^{\text{Rf}}), \quad a_t^{\text{Rf}} \in \mathcal{A}^{\text{Rf}}. \quad (7)$$

Thus, each round consists of an exploration transition followed by a refinement transition, coupling coverage expansion and noise control within the same verification state.

3.3.2 Explorer and Refiner Edit Actions

At each decision step, player $p \in \{\text{Ex}, \text{Rf}\}$ observes a role-specific decision state S_t^p , where Explorer observes $S_t^{\text{Ex}} = S_t$ and Refiner observes $S_t^{\text{Rf}} = S'_t$. We parameterize each edit action as a selection-execution pair $a_t^p = (a_{t,\text{sel}}^p, a_{t,\text{exe}}^p)$. The selection variable $a_{t,\text{sel}}^p$ identifies whether the evidence or analysis component should be revised, and the execution variable $a_{t,\text{exe}}^p$ specifies the edit operation applied to that component. Accordingly, the role-specific policy π_θ^p first samples $a_{t,\text{sel}}^p$ with $\pi_{\theta,\text{sel}}^p(\cdot | S_t^p)$ and then samples $a_{t,\text{exe}}^p$ with $\pi_{\theta,\text{exe}}^p(\cdot | S_t^p, a_{t,\text{sel}}^p)$. This selection-execution factorization makes each edit component-aware: the policy localizes the part of the verification state to revise before choosing a role-specific operation. The selected operation is sent to the LLM executor, and Φ^p writes the operation output back to the verification state.

Explorer expands factual coverage with three operations: (1) Retrieve supplements missing

factual evidence for the selected component; (2) Rewrite reorganizes existing evidence or analysis into a clearer verification signal; and (3) Counterfactual-Analysis constructs contrastive checks over entities, time, scope, quantities, or assumptions that may affect the support/refutation relation. The Explorer edit produces the intermediate state S'_t . Refiner controls noise with two operations: (1) Merge combines overlapping entries that express the same factual signal; and (2) Reduce separates weakly relevant, low-value, or potentially misleading fragments while preserving verdict-relevant signals in the active state. The Refiner edit produces the next state S_{t+1} and a separated-fragment set ΔR_t , which is accumulated as $R_{t+1} = R_t \cup \Delta R_t$ with $R_0 = \emptyset$.

3.3.3 Reward Design and Game Stabilization

We use an LLM judge J to assign role-specific payoffs to completed state-editing transitions during training. Explorer receives $r_t^{\text{Ex}} = J^{\text{Ex}}(S_t, a_t^{\text{Ex}}, S'_t)$ for editing S_t into S'_t , where J^{Ex} rewards missing-evidence supplementation, factual-signal reorganization, and contrastive checks that clarify the support/refutation relation. Refiner receives $r_t^{\text{Rf}} = J^{\text{Rf}}(S'_t, a_t^{\text{Rf}}, S_{t+1}, \Delta R_t)$ for editing S'_t into S_{t+1} and separating fragment ΔR_t , where J^{Rf} rewards redundancy reduction and weak-fragment separation while penalizing the loss of verdict-relevant information.

These payoffs couple coverage expansion and noise control over the same evolving verification state. Let Π^p denote the policy space of player p , τ denote the stopping round, and γ denote the discount factor. For $p \in \{\text{Ex}, \text{Rf}\}$, the discounted return is $V^p(\pi^{\text{Ex}}, \pi^{\text{Rf}}) = \mathbb{E}[\sum_{t=0}^{\tau-1} \gamma^t r_t^p]$, which defines player p 's payoff over an editing trajectory and is used in the following unilateral-deviation condition. We use the ϵ -Nash condition as a stability target for the finite-horizon state-editing game:

$$V^p(\pi^{p,*}, \pi^{-p,*}) \geq V^p(\pi^p, \pi^{-p,*}) - \epsilon, \quad \forall p \in \{\text{Ex}, \text{Rf}\}, \forall \pi^p \in \Pi^p. \quad (8)$$

Here, $-p$ denotes the other player. This condition characterizes a stable coverage-noise balance, where neither Explorer nor Refiner can obtain substantial payoff improvement by changing its policy alone. Details of the ϵ -Nash stability criterion and reward-stability proxy are provided in App. B.1.

3.3.4 Policy Training and Inference

We learn Explorer and Refiner policies with SFT warm-up followed by alternating PPO optimization. SFT initializes each role-specific factored policy from teacher-labeled editing trajectories, where each action is a selection-execution pair $a_t^p = (a_{t,\text{sel}}^p, a_{t,\text{exe}}^p)$. This warm-up teaches the policy to select an editable state component and then choose a valid operation conditioned on that component, following Sec. 3.3.2. The complete SFT objective and sampling details are provided in App. B.2.

PPO (Schulman et al., 2017) then optimizes the two policies using the role-specific rewards in Sec. 3.3.3. During each Explorer-Refiner rollout, the two players act in the transition order defined in Sec. 3.3.1, producing role-specific trajectories for policy updates. For player p , the structured-action log-probability sums the selection score and the conditioned execution score: $\ell_t^p(\theta) = \log \pi_{\theta,\text{sel}}^p(a_{t,\text{sel}}^p | S_t^p) + \log \pi_{\theta,\text{exe}}^p(a_{t,\text{exe}}^p | S_t^p, a_{t,\text{sel}}^p)$. The PPO ratio is $\rho_t^p(\theta) = \exp(\ell_t^p(\theta) - \ell_t^p(\theta_{\text{old}}))$, and $\bar{\rho}_t^p(\theta)$ denotes its clipped value. With advantage A_t^p , the clipped policy loss is:

$$\mathcal{L}_{\text{clip}}^p(\theta) = -\mathbb{E}_t[\min(\rho_t^p(\theta) A_t^p, \bar{\rho}_t^p(\theta) A_t^p)]. \quad (9)$$

This joint-ratio update keeps component selection and operation execution coupled during optimization. Alternating PPO updates improve each role under trajectories generated by the other role's current policy, while clipping and KL monitoring limit abrupt policy drift. The full PPO objective further includes value fitting, entropy regularization, and SFT-prior matching, as detailed in App. B.3.

At inference time, starting from S_0 , the learned policies alternate Explorer and Refiner edits for a finite editing horizon, without using the reward judge or gold label. If the stopping round is τ , the game outputs $S^* = S_\tau$ and $R = R_\tau$. The pair (S^*, R) forms the interface to final verdict aggregation.

3.4 Backdoor-Adjusted Verdict Aggregation

To mitigate residual shortcut effects during final aggregation, we use the removed-fragment set R to instantiate adjustment conditions for verification. Although the Explorer-Refiner Game separates weak or misleading fragments, the refined state may still retain residual construction-time shortcut and noise signals. We therefore keep S^* fixed and aggregate verifier predictions over adjustment conditions instantiated from R .

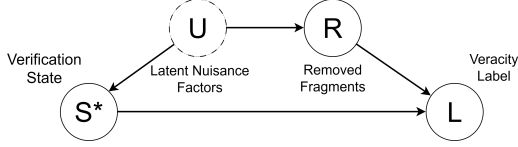


Figure 2: Causal abstraction of Backdoor-Adjusted Verdict Aggregation. S^* is the fixed refined state, L is the veracity label, U denotes unobserved construction-time nuisance factors, and R supplies removed-fragment adjustment conditions.

Backdoor adjustment (Pearl, 2009) is a causal debiasing method for the scenarios where the outcome prediction may depend on nuisance factors in addition to the treatment of interest. It fixes the treatment and marginalizes over an adjustment variable, so that the resulting estimate accounts for different nuisance conditions. The adjustment variable captures nuisance or confounding conditions that may affect both the treatment and the outcome; marginalizing over it reduces the dependence of the estimate on any single nuisance condition. Given treatment T , outcome Y , and adjustment variable Z , the standard form is:

$$p(Y \mid do(T)) = \sum_{z \in Z} p(Z = z)p(Y \mid T, Z = z). \quad (10)$$

Here, $do(T)$ means that the treatment is fixed, and each value z defines one adjustment condition.

We apply the backdoor adjustment to our method and introduce a removed-fragment proxy adjustment method that uses the fragments in removed fragments set R as proxy conditions for the adjustment variable Z . In our model, the treatment T corresponds to the refined verification state S^* in claim verification and the outcome Y corresponds to the veracity label L . The latent variable U corresponds to adjustment variable Z , it represents construction-time nuisance factors that affect the construction of S^* and bias the verifier’s label prediction. Since U is unobserved, exact adjustment over U is unavailable. We therefore use R as its observable proxy adjustment variable. This proxy design is tied to our verification task: during the construction of S^* , the Refiner separates weak, noisy, or misleading fragments into R , so these removed fragments provide observable traces of construction-time shortcut factors. We therefore use each removed fragment $r \in R$ as one observable proxy adjustment condition while keeping S^* fixed. Fig. 2 illustrates this proxy-variable backdoor adjustment: $S^* \rightarrow L$ is the main prediction

path, $S^* \leftarrow U \rightarrow R$ indicates that construction-time nuisance factors affect both the retained and removed information, and $R \rightarrow L$ denotes conditioning verifier predictions on removed-fragment adjustment conditions.

Specifically, following the above idea of using backdoor adjustment, we estimate the adjusted score of each label l by marginalizing verifier predictions over proxy adjustment conditions from R :

$$\begin{aligned} \hat{p}_{BA}(l \mid do(S^*)) &= \sum_{r \in R} q(r \mid R) p(L = l \mid S^*, r) \\ &\approx \frac{1}{K} \sum_{k=1}^K \mathbf{1}[V(S^*, r_k) = l]. \end{aligned} \quad (11)$$

Here, $q(r \mid R)$ is the empirical sampling distribution over removed-fragment adjustment conditions. The first line follows the backdoor-style marginalization over proxy conditions, and the second line gives its sampling approximation when the LLM verifier V returns hard labels. The final label is obtained by voting:

$$l_{\text{final}} = \arg \max_l \hat{p}_{BA}(l \mid do(S^*)). \quad (12)$$

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our method on HoVer (Jiang et al., 2020) and FEVEROUS-S (Aly et al., 2021). HoVer evaluates many-hop claim verification and is reported on its 2-hop, 3-hop, and 4-hop validation subsets. FEVEROUS-S is the sentence-evidence subset of FEVEROUS and evaluates textual Wikipedia evidence verification. All experiments use the open-book setting, where annotated gold evidence is unavailable at inference time.

Baselines. We compare ERGA with two controlled baseline groups. The first group consists of causal verifier baselines, including Causal-CRFV (Zhao et al., 2026a) and Causal Walk (Zhang et al., 2024a). The second group consists of LLM-reasoning baselines, including ProgramFC (Pan et al., 2023), BiDeV (Liu et al., 2025), VeGraph (Pham et al., 2025), and BOOST (Hu et al., 2025a).

Metrics and Implementation. We report Macro-F1 as the primary metric and Accuracy as the secondary metric. For HoVer, we report hop-wise results; for FEVEROUS-S, we report results on the full sentence-evidence evaluation set. All baselines are evaluated under matched open-book settings,

Category	Method	HoVer-2		HoVer-3		HoVer-4		FEVEROUS-S	
		F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.
Causal	Causal-CRFV (Zhao et al., 2026a)	58.99	61.63	67.12	67.85	61.17	61.98	70.43	70.56
	Causal Walk (Zhang et al., 2024a)	65.77	66.25	<u>70.19</u>	<u>70.19</u>	57.70	60.92	66.95	67.08
LLM	ProgramFC (Pan et al., 2023)	71.96	72.29	61.09	61.96	54.14	58.33	62.73	66.41
	BiDeV (Liu et al., 2025)	68.76	69.80	56.74	58.96	54.39	59.96	71.65	73.11
	VeGraph (Pham et al., 2025)	73.92	74.16	65.81	65.91	59.61	59.63	77.50	78.06
	BOOST (Hu et al., 2025a)	71.96	72.02	66.46	66.49	<u>62.44</u>	<u>63.23</u>	<u>79.82</u>	<u>80.11</u>
	ERGA	74.68	74.69	71.43	71.44	66.62	66.89	82.76	82.78

Table 1: Comparison of Macro-F1 and Accuracy for ERGA and baselines on HoVer and FEVEROUS-S. Bold and underline indicate the best and second-best results in each column. Our improvements over all baselines are significant under Holm-corrected McNemar tests with $p < 0.05$; see App. E.

with full details in App. A.

4.2 Overall Performance

Table 1 presents the main comparison of ERGA and baselines on HoVer and FEVEROUS-S. ERGA achieves the best performance on all eight reported metrics, improving over the strongest baseline under the same setting by 2.28 Macro-F1 points and 2.03 Accuracy points on average. The gain is smaller on HoVer-2, as two-hop claims leave less room for state refinement; it becomes larger on HoVer-3 and HoVer-4 because the verifier must preserve more factual constraints while filtering weak or redundant fragments. The improvement on FEVEROUS-S further shows that ERGA remains effective on sentence-evidence verification beyond HoVer-style hop-labeled claims. These results indicate that constructing a better coverage-noise-balanced verification state improves complex fact-checking under the open-book setting. Representative qualitative cases are provided in App. G.

4.3 Ablation Study

Table 2 presents the ablation results of ERGA on HoVer and FEVEROUS-S, showing that the full model consistently outperforms all ablated variants. Removing Decomposition-Guided State Initialization (w/o State Initialization, Sec. 3.2) causes the largest performance drop, indicating that decomposition is crucial for exposing local factual conditions and organizing retrieved evidence and local analyses into a usable initial state S_0 . Removing the Explorer-Refiner Game (w/o Explorer-Refiner Game, Sec. 3.3) also degrades performance, showing the necessity of explicit state editing for balancing evidence coverage and noisy-fragment separation. Removing Backdoor-Adjusted Verdict Aggregation (w/o Proxy Adjustment, Sec. 3.4) further

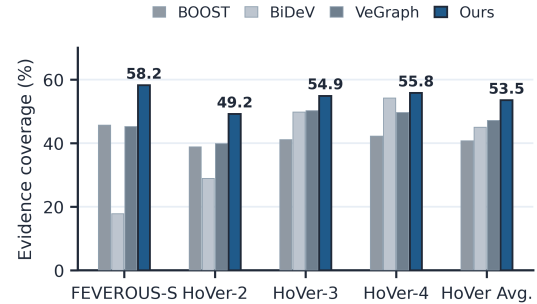


Figure 3: Evidence coverage in verification contexts on HoVer and FEVEROUS-S. Coverage is the average percentage of annotated evidence units matched by the context used before verdict prediction.

hurts performance, suggesting that the R -derived condition pool provides useful auxiliary signals for final verdict aggregation.

4.4 Analysis Study of Evidence Coverage

To show that ERGA preserves claim-deciding evidence before prediction, we measure evidence coverage in verification contexts on HoVer and FEVEROUS-S. For each example, we split the annotated evidence into evidence units and match them against the context used by each method before verdict prediction. The metric is the percentage of matched evidence units, averaged within each dataset or hop group. As shown in Fig. 3, ERGA achieves the highest coverage on both datasets, reaching 58.2% on FEVEROUS-S and 53.5% on HoVer on average. It also remains best across 2-, 3-, and 4-hop HoVer claims. These results show that the refined state S^* preserves more claim-deciding evidence before verdict prediction.

4.5 Analysis Study of Coverage-Noise Balance

To examine whether ERGA improves retained con-

Variants	HoVer-2		HoVer-3		HoVer-4		FEVEROUS-S	
	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.
Full ERGA	74.68	74.69	71.43	71.44	66.62	66.89	82.76	82.78
w/o State Initialization (Sec. 3.2)	69.47	70.34	62.67	63.81	58.75	62.08	70.97	73.13
w/o Explorer–Refiner Game (Sec. 3.3)	71.05	71.14	68.91	69.32	66.05	66.22	81.40	81.47
w/o Proxy Adjustment (Sec. 3.4)	70.69	70.69	68.33	68.39	66.20	66.22	79.82	79.95

Table 2: Ablation study of ERGA on HoVer and FEVEROUS-S. Each w/o variant removes one module from the full model. Scores are reported in percentage points, and bold numbers indicate the best result in each column.

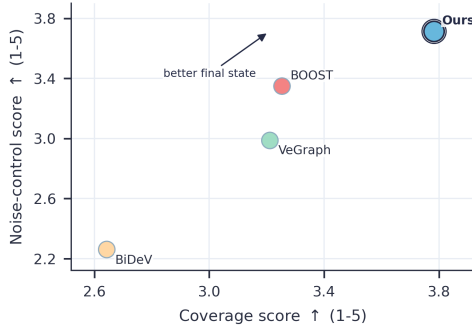


Figure 4: Coverage–noise balance of retained verification contexts. Each point is the unweighted dataset-level average over HoVer and FEVEROUS-S. Coverage and noise control are reported on 1–5 scales, where higher values indicate better context quality.

text quality rather than simply adding more text, we evaluate the coverage–noise balance on HoVer and FEVEROUS-S. For each claim, an LLM judge assesses the retained verification context along two dimensions: verification-relevant coverage and noise control. Coverage measures whether the context contains information needed for verification, while noise control measures whether it avoids weak, redundant, misleading, or underspecified fragments that may distract the verifier. As shown in Fig. 4, ERGA achieves the best average scores on both dimensions, reaching 3.79 in coverage and 3.72 in noise control. The strongest baseline scores are 3.25 and 3.35, respectively. This pattern suggests that ERGA constructs a better coverage–noise-balanced context for final prediction, rather than gaining by merely retaining more content.

4.6 Analysis Study of Adjustment Conditions

To examine whether removed fragments provide useful adjustment conditions, we conduct a condition-removal intervention on HoVer and FEVEROUS-S. This intervention keeps the upstream outputs, refined state S^* , verifier, and aggregation procedure unchanged, but disables the R -derived condition pool used for conditional verifier

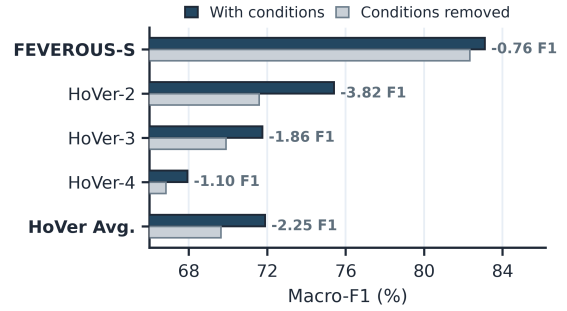


Figure 5: Effect of removing R -derived adjustment conditions. The variant removes adjustment conditions derived from the removed-fragment set R during verdict aggregation. Labels report the resulting Macro-F1 drop.

calls. Since examples without generated conditions are unaffected, we report Macro-F1 on examples where ERGA produces at least one adjustment condition. As shown in Fig. 5, removing these conditions decreases Macro-F1 by 0.76 points on FEVEROUS-S and 2.25 points on HoVer. The drop is also consistent in direction across HoVer hop groups. These results indicate that the removed fragments are not merely discarded noise; they provide sample-specific conditions that help decide when auxiliary verdicts should affect final aggregation.

5 Conclusions

In this paper, we propose ERGA for LLM-based complex fact-checking, focusing on how to construct an informative and less noisy verification state before final prediction. ERGA first initializes the state with decomposed verification units, retrieved evidence, and local analyses. It then uses an Explorer–Refiner Game to expand missing factual signals while separating redundant, weakly relevant, or potentially misleading fragments. Finally, Backdoor-Adjusted Verdict Aggregation uses the separated fragments as proxy adjustment conditions for final verdict prediction. Experiments on HoVer and FEVEROUS-S show that ERGA

achieves the best performance across all reported Macro-F1 and Accuracy metrics.

Limitations

This work follows prior complex fact-checking studies and evaluates on English, Wikipedia-based benchmarks with textual evidence. These benchmarks support controlled comparison, while claims that require broader evidence sources remain outside the present scope.

Like other LLM-based fact-checking frameworks, ERGA uses multiple LLM calls to construct and refine intermediate states. This improves the controllability of verification, but also introduces additional computational cost. Reducing this cost while keeping the verification process easy to inspect is an important direction for future work.

Ethical Considerations

This work aims to support reliable automated fact-checking by improving evidence-grounded verification for complex claims. By organizing retrieved evidence and separating weak or misleading fragments, ERGA is intended to reduce the risk of unsupported or noisy information affecting final verdicts.

Same as existing fact-checking systems, ERGA cannot guarantee fully correct judgments in all scenarios, especially when the retrieved evidence is incomplete or the LLM-generated analysis contains unsupported content. Therefore, we recommend using ERGA as an assistant to human fact-checkers rather than as an autonomous decision-making tool. For high-stakes claims, such as those involving public health, elections, legal decisions, or individuals, users should further verify the evidence and combine system outputs with human review.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No.62376284 and No.62306330.

References

- Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. [FEVEROUS: Fact extraction and VERification over unstructured and structured information](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.
- Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2024. [Complex Claim Verification with Evidence Retrieved in the Wild](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3569–3587, Mexico City, Mexico. Association for Computational Linguistics.
- Islam Eldifrawi, Shengrui Wang, and Amine Trabelsi. 2024. [Automated Justification Production for Claim Veracity in Fact Checking: A Survey on Architectures and Approaches](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6679–6692, Bangkok, Thailand. Association for Computational Linguistics.
- Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. [A survey on automated fact-checking](#). *Transactions of the Association for Computational Linguistics*, 10:178–206.
- Sture Holm. 1979. [A simple sequentially rejective multiple test procedure](#). *Scandinavian Journal of Statistics*, 6(2):65–70.
- Qisheng Hu, Quanyu Long, and Wenya Wang. 2025a. [BOOST: Bootstrapping Strategy-Driven Reasoning Programs for Program-Guided Fact-Checking](#). *Preprint*, arXiv:2504.02467.
- Qisheng Hu, Quanyu Long, and Wenya Wang. 2025b. [Coordinating Search-Informed Reasoning and Reasoning-Guided Search in Claim Verification](#). *Preprint*, arXiv:2506.07528.
- Qisheng Hu, Quanyu Long, and Wenya Wang. 2025c. [Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance?](#) In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6313–6336, Albuquerque, New Mexico. Association for Computational Linguistics.
- Yani Huang, Richong Zhang, Zhijie Nie, Junfan Chen, and Xuefeng Zhang. 2025. [A Graph-based Verification Framework for Fact-Checking](#). *Preprint*, arXiv:2503.07282.
- Eojin Jeon, Mingyu Lee, Juhyeong Park, Yeachan Kim, Wing-Lam Mok, and SangKeun Lee. 2023. [Improving Bias Mitigation through Bias Experts in Natural Language Understanding](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11053–11066, Singapore. Association for Computational Linguistics.
- Hyewon Jeon and Jay-Yoon Lee. 2025. [GraphCheck: Multipath fact-checking with entity-relationship graphs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24728–24745, Suzhou, China. Association for Computational Linguistics.

- Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. [Exploring Listwise Evidence Reasoning with T5 for Fact Verification](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410, Online. Association for Computational Linguistics.
- Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. [HoVer: A Dataset for Many-Hop Fact Extraction And Claim Verification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.
- David M. J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts, and Jonathan L. Zittrain. 2018. [The science of fake news](#). *Science*, 359(6380):1094–1096.
- Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheon-Eum Park, and Kyomin Jung. 2021. [CrossAug: A contrastive data augmentation method for debiasing fact verification models](#). In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pages 3181–3185, New York, NY, USA. Association for Computing Machinery.
- Yuxuan Liu, Hongda Sun, Wenya Guo, Xinyan Xiao, Cunli Mao, Zhengtao Yu, and Rui Yan. 2025. [BiDeV: Bilateral defusing verification for complex claim fact-checking](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1):541–549.
- Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. [Fine-grained fact verification with kernel graph attention network](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.
- Huanhuan Ma, Weizhi Xu, Yifan Wei, Liuji Chen, Liang Wang, Qiang Liu, Shu Wu, and Liang Wang. 2024. [EX-FEVER: A dataset for multi-hop explainable fact verification](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9340–9353, Bangkok, Thailand. Association for Computational Linguistics.
- Jiatong Ma, Linmei Hu, Rang Li, and Wenbo Fu. 2025. [LoCal: Logical and Causal Fact-Checking with LLM-Based Multi-Agents](#). In *Proceedings of the ACM on Web Conference 2025*, pages 1614–1625, Sydney NSW Australia. ACM.
- Quinn McNemar. 1947. [Note on the sampling error of the difference between correlated proportions or percentages](#). *Psychometrika*, 12(2):153–157.
- Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. [Fact-checking complex claims with program-guided reasoning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6981–7004, Toronto, Canada. Association for Computational Linguistics.
- Shilong Pan, Zhiliang Tian, Liang Ding, Haoqi Zheng, Zhen Huang, Zhihua Wen, and Dongsheng Li. 2024. [POMP: Probability-driven meta-graph prompter for LLMs in low-resource unsupervised neural machine translation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9976–9992, Bangkok, Thailand. Association for Computational Linguistics.
- Shilong Pan, Zhiliang Tian, Zhen Huang, Wanlong Yu, Zhihua Wen, Xinwang Liu, Kai Lu, Minlie Huang, and Dongsheng Li. 2025. [AGD: Adversarial game defense against jailbreak attacks in large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17391–17406, Vienna, Austria. Association for Computational Linguistics.
- Shilong Pan, Zhiliang Tian, Wanlong Yu, Zhen Huang, Qingyu Qiu, Zihan Chen, Zhonghao Sun, Minlie Huang, and Dongsheng Li. 2026. [WALKSAFE: Risk-aware graph random walk with Bi-GRPO for LLM safety](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(38):32655–32663.
- Judea Pearl. 2009. [Causality](#). Cambridge university press.
- Hoang Pham, Thanh-Do Nguyen, and Khac-Hoai Nam Bui. 2025. [Verify-in-the-graph: Entity disambiguation enhancement for complex claim verification with interactive graph representation](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5181–5197, Albuquerque, New Mexico. Association for Computational Linguistics.
- Stephen E. Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: Bm25 and beyond](#). *Foundations and Trends in Information Retrieval*, 3(4):333–389.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *arXiv.org*, arXiv:1707.06347.
- Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. [Get your vitamin C! robust fact verification with contrastive evidence](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

- Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. [Towards debiasing fact verification models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, Hong Kong, China. Association for Computational Linguistics.
- Amir Soleimani, Christof Monz, and Marcel Worring. 2020. [BERT for evidence retrieval and claim verification](#). In *Advances in Information Retrieval*, pages 359–366, Cham. Springer International Publishing.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. [FEVER: A large-scale dataset for fact extraction and VERification](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.
- Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. 2022. [Debiasing NLU Models via Causal Intervention and Counterfactual Reasoning](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11376–11384.
- Zhiliang Tian, Jingyuan Huang, Zejiang He, Zhen Huang, Menglong Lu, Linbo Qiao, Songzhu Mei, Yijie Wang, and Dongsheng Li. 2025. [LLM-based rumor detection via influence guided sample selection and game-based perspective analysis](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 28402–28414. Association for Computational Linguistics.
- Ivan Vykopal, Matúš Pikuliak, Simon Ostermann, and Marián Šimko. 2024. [Generative Large Language Models in Automated Fact-Checking: A Survey](#). *Preprint*, arXiv:2407.02351.
- Haoran Wang and Kai Shu. 2023. [Explainable claim verification via knowledge-grounded reasoning with large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6288–6304, Singapore. 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. Curran Associates, Inc.
- Junda Wu, Tong Yu, Xiang Chen, Haoliang Wang, Ryan Rossi, Sungchul Kim, Anup Rao, and Julian McAuley. 2024. [DeCoT: Debiasing Chain-of-Thought for Knowledge-Intensive Tasks in Large Language Models via Causal Intervention](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14073–14087, Bangkok, Thailand. Association for Computational Linguistics.
- Zhuohan Xie, Rui Xing, Yuxia Wang, Jiahui Geng, Hasan Iqbal, Dhruv Sahnan, Iryna Gurevych, and Preslav Nakov. 2025. [FIRE: Fact-checking with iterative retrieval and verification](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2901–2914, Albuquerque, New Mexico. Association for Computational Linguistics.
- Weizhi Xu, Qiang Liu, Shu Wu, and Liang Wang. 2023. [Counterfactual Debiasing for Fact Verification](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6777–6789, Toronto, Canada. Association for Computational Linguistics.
- Xie Yi, Zhanke Zhou, Chentao Cao, Qiyu Niu, Tongliang Liu, and Bo Han. 2025. [From debate to equilibrium: Belief-driven multi-agent LLM reasoning via bayesian nash equilibrium](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 72277–72316. PMLR.
- Xia Zeng, Amani S. Abumansour, and Arkaitz Zubiaga. 2021. [Automated fact-checking: A survey](#). *Language and Linguistics Compass*, 15(10):e12438.
- Congzhi Zhang, Linhai Zhang, and Deyu Zhou. 2024a. [Causal walk: Debiasing multi-hop fact verification with front-door adjustment](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19533–19541.
- Yue Zhang, Shicheng Zhou, Xiaopeng Li, Zhiliang Tian, Yifu Gao, Shiyi Zhang, Wenqing Hou, Yuying Liu, and Bin Zhou. 2026. [KnowFC: Navigating knowledge conflicts in large language model-based fact-checking](#). In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining, WSDM 2026, Boise, ID, USA, February 22-26, 2026*, pages 996–1006. ACM.
- Ziqi Zhang, Cunxiang Wang, Xiao Xiong, Yue Zhang, and Donglin Wang. 2024b. [Nash CoT: Multi-Path Inference with Preference Equilibrium](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14572–14587, Miami, Florida, USA. Association for Computational Linguistics.
- Jianbo Zhao, Yuzhong Peng, Zongbao Yang, Zhichen Chen, Changan Yuan, Xiao Qin, Ruxin Wang, and Hao Zhang. 2026a. [Dynamic debiasing of multi-hop fact verification via counterfactual reasoning](#). *Knowledge-Based Systems*, 332:114875.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao

Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, and 4 others. 2026b. [A survey of large language models](#). *Frontiers of Computer Science*, 20:2012627.

Xiaoyan Zhao, Lingzhi Wang, Zhanghao Wang, Hong Cheng, Rui Zhang, and Kam-Fai Wong. 2024. [PACAR: Automated Fact-Checking with Planning and Customized Action Reasoning Using Large Language Models](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12564–12573, Torino, Italia. ELRA and ICCL.

Liwen Zheng, Chaozhuo Li, Xi Zhang, Yu-Ming Shang, Feiran Huang, and Haoran Jia. 2024. [Evidence Retrieval is almost All You Need for Fact Verification](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9274–9281, Bangkok, Thailand. Association for Computational Linguistics.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. [GEAR: Graph-based evidence aggregating and reasoning for fact verification](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy. Association for Computational Linguistics.

A Implementation Details

This appendix reports the implementation settings used in the experiments, including the LLM executor, policy optimization, retrieval setting, and inference hyperparameters.

All controlled LLM-reasoning baselines and the LLM-executed components of ERGA use the same API LLM, `gemin-3.1-flash-lite-preview`, within each reported comparison. The API LLM is used for claim decomposition, state editing, reward judging, and verdict generation, and is not fine-tuned. Retrieval uses FlashRAG BM25 over the benchmark-specific Wikipedia corpus.

The learned policies are lightweight role-specific action policies. They use a ModernBERT-base encoder with LoRA adapters, a component-selection head, a conditional operation-execution head, and a scalar value head. During policy learning, gold labels are used only for reward assignment and are never included in inference prompts. At inference time, the learned Explorer and Refiner policies select discrete edit actions, while the API LLM executes the selected textual operation. The maximum number of Explorer–Refiner game rounds and the number of conditional verdict samples are selected on the development set and then fixed for test evaluation. The prompt templates used by the LLM executor and verifier are listed in App. F. Table 3

summarizes the main implementation, training, and inference hyperparameters.

B Training Details

B.1 Nash Equilibrium Condition

Sec. 3.3.3 gives the compact ϵ -Nash condition for the Explorer–Refiner game. Here, we expand the condition into its role-wise form and clarify how the stability target is reflected in finite-horizon policy learning. Let τ be the stopping round and γ be the discount factor. For player $p \in \{\text{Ex}, \text{Rf}\}$, the discounted return is:

$$V^p(\pi^{\text{Ex}}, \pi^{\text{Rf}}) = \mathbb{E}\left[\sum_{t=0}^{\tau-1} \gamma^t r_t^p\right]. \quad (13)$$

Let $(\pi_*^{\text{Ex}}, \pi_*^{\text{Rf}})$ denote a stable policy pair and $V_*^p = V^p(\pi_*^{\text{Ex}}, \pi_*^{\text{Rf}})$. For Explorer, the condition requires:

$$V_*^{\text{Ex}} \geq V^{\text{Ex}}(\pi^{\text{Ex}}, \pi_*^{\text{Rf}}) - \epsilon, \quad \forall \pi^{\text{Ex}} \in \Pi^{\text{Ex}}. \quad (14)$$

For Refiner, it requires:

$$V_*^{\text{Rf}} \geq V^{\text{Rf}}(\pi_*^{\text{Ex}}, \pi^{\text{Rf}}) - \epsilon, \quad \forall \pi^{\text{Rf}} \in \Pi^{\text{Rf}}. \quad (15)$$

The two inequalities instantiate the same unilateral-deviation principle for the two editing roles. Explorer is stable when changing its coverage-oriented policy alone cannot yield substantial payoff improvement under the current Refiner policy. Refiner is stable when changing its reduction-oriented policy alone cannot yield substantial payoff improvement under the current Explorer policy. Thus, the equilibrium condition characterizes a stable role interaction in which coverage expansion and noise control are jointly balanced.

In finite-horizon alternating PPO, this stability target is monitored through role-wise payoff changes across rollout rounds. After a minimum number of interaction rounds, an Explorer–Refiner rollout is considered stable when the batch-level payoffs of both roles vary only marginally over consecutive rounds. This reward-stability signal serves as the practical proxy for the Nash-style stability target during policy learning.

B.2 SFT

For each training claim, we first construct the initial verification state S_0 following Sec. 3.2. We then collect supervised editing trajectories for Explorer and Refiner. Given the current decision

Parameter	Value	Parameter	Value
Experimental Setup			
GPU	NVIDIA A800 80GB	API LLM executor / judge / verifier	gemini-3.1-flash-lite-preview
Retriever	BM25-based sparse retriever via FlashRAG	Open-book corpus	benchmark-specific Wikipedia corpus
Retrieved candidates per query K_{ret}	3	Maximum evidence length	no truncation
Default adjustment samples K	3	Maximum adjustment samples	16
Policy Network			
Policy encoder	ModernBERT-base	Maximum input length	4096
Policy heads	selection; conditional execution	Value head	scalar critic
LoRA rank / alpha / dropout	32 / 64 / 0.2	LoRA target modules	Wqkv, Wo, W1, W2
SFT Warm-up			
SFT data-collection episodes	10	SFT epochs / batch size	3 / 16
SFT learning rate	5×10^{-5}	SFT balancing mode / prior weight	joint / 0.2
PPO Optimization and Stabilization			
PPO episodes / epochs per update	50 / 1	Maximum game rounds T_{max}	5
Batch size / mini-batch size	32 / 8	Update interval	8 episodes
Optimizer / learning rate	AdamW / 5×10^{-6}	Discount factor / GAE lambda	0.99 / 0.95
Policy clip / value clip	0.1 / 0.5	Value loss coefficient	0.15
Entropy coefficient	0.20 to 0.08 linear decay	Rollout exploration mixture	0.20
Target KL	0.03	Nash-style early stopping	enabled
Reward-delta threshold / patience	0.03 / 2	Minimum rounds before early stop	2

Table 3: Main implementation and optimization hyperparameters used by ERGA.

state, a teacher LLM selects a state component and an operation according to the action space in Sec. 3.3.2. Each labeled action is executed in the editing environment to produce the next decision state, yielding role-specific trajectories for both players. For player $p \in \{\text{Ex}, \text{Rf}\}$, each trajectory contains a decision state S_t^p and a teacher-labeled action $a_t^p = (a_{t,\text{sel}}^p, a_{t,\text{exe}}^p)$. The selection loss is:

$$\mathcal{L}_{\text{sel}}^p = -\mathbb{E}_{\mathcal{T}^p} \log \pi_{\theta,\text{sel}}^p(a_{t,\text{sel}}^p | S_t^p). \quad (16)$$

The execution head is trained with teacher forcing conditioned on the selected component:

$$\mathcal{L}_{\text{exe}}^p = -\mathbb{E}_{\mathcal{T}^p} \log \pi_{\theta,\text{exe}}^p(a_{t,\text{exe}}^p | S_t^p, a_{t,\text{sel}}^p). \quad (17)$$

The SFT objective is:

$$\mathcal{L}_{\text{SFT}}^p = \mathcal{L}_{\text{sel}}^p + \mathcal{L}_{\text{exe}}^p + \lambda_{\text{sft}} \mathcal{L}_{\text{prior}}^p. \quad (18)$$

The prior term regularizes the action distribution estimated from supervised trajectories and reduces collapse toward a single component or operation.

B.3 PPO Optimization

After SFT warm-up, PPO optimizes the action policies with the rewards in Sec. 3.3.3. For player p , advantages are estimated with a value baseline. The temporal-difference residual is:

$$\delta_t^p = r_t^p + \gamma b^p(S_{t+1}^p) - b^p(S_t^p). \quad (19)$$

The advantage is computed as:

$$A_t^p = \sum_{k=0}^{\tau-t-1} (\gamma\lambda)^k \delta_{t+k}^p. \quad (20)$$

For compact notation, let $m_t^p = a_{t,\text{sel}}^p$ and $o_t^p = a_{t,\text{exe}}^p$. The joint log-probability of the selected component and operation is:

$$\ell_t^p(\theta) = \log \pi_{\theta,\text{sel}}^p(m_t^p | S_t^p) + \log \pi_{\theta,\text{exe}}^p(o_t^p | S_t^p, m_t^p). \quad (21)$$

The PPO ratio is:

$$\rho_t^p(\theta) = \exp(\ell_t^p(\theta) - \ell_t^p(\theta_{\text{old}})). \quad (22)$$

The clipped ratio is:

$$\bar{\rho}_t^p(\theta) = \text{clip}(\rho_t^p(\theta), 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}}). \quad (23)$$

The clipped policy loss is:

$$\mathcal{L}_{\text{clip}}^p(\theta) = -\mathbb{E}_t[\min(\rho_t^p(\theta) A_t^p, \bar{\rho}_t^p(\theta) A_t^p)]. \quad (24)$$

The full training loss adds value fitting and stability regularization:

$$\mathcal{L}_{\text{train}}^p = \mathcal{L}_{\text{clip}}^p + c_v \mathcal{L}_V^p - \beta \mathcal{H}^p + \lambda_{\text{prior}} \mathcal{L}_{\text{prior}}^p. \quad (25)$$

PPO uses advantage normalization, value clipping, entropy regularization, action-prior matching from SFT trajectories, and target-KL early stopping.

C Model Workflow Pseudocode

Algorithm 1 summarizes the complete inference workflow of ERGA, corresponding to the three modules in Sec. 3.1. The inference-time budgets, including the maximum game rounds T_{max} and the adjustment sample budget K , are reported in Table 3.

Algorithm 1 ERGA Inference

Require: Claim C , decomposer D , retriever Ret , analyzer Ana

Require: Policies $\pi^{\text{Ex}}, \pi^{\text{Rf}}$, maximum rounds $T_{\max}$, adjustment budget K

Ensure: Final label l_{final}

Stage I: Decomposition-Guided State Initialization

```
1:  $\mathcal{C} \leftarrow D(C) = \{c_i\}_{i=1}^n$ 
2: for  $i = 1$  to  $n$  do
3:    $E_i \leftarrow \text{Ret}(c_i)$ 
4:    $a_i \leftarrow \text{Ana}(C, c_i, E_i)$ 
5: end for
6:  $S_0 \leftarrow (C, \{(c_i, E_i, a_i)\}_{i=1}^n)$ 
7:  $S \leftarrow S_0, R \leftarrow \emptyset$ 
```

Stage II: Explorer–Refiner Game

```
8: for  $t = 0$  to  $T_{\max} - 1$  do
9:    $S_t \leftarrow S$ 
10:   $S_t^{\text{Ex}} \leftarrow S_t$ 
11:   $a_t^{\text{Ex}} \sim \pi^{\text{Ex}}(\cdot \mid S_t^{\text{Ex}})$ 
   where  $a_t^{\text{Ex}} = (a_{t,\text{sel}}^{\text{Ex}}, a_{t,\text{exe}}^{\text{Ex}})$ 
12:   $S_t' \leftarrow \Phi^{\text{Ex}}(S_t, a_t^{\text{Ex}})$ 
13:   $S_t^{\text{Rf}} \leftarrow S_t'$ 
14:   $a_t^{\text{Rf}} \sim \pi^{\text{Rf}}(\cdot \mid S_t^{\text{Rf}})$ 
   where  $a_t^{\text{Rf}} = (a_{t,\text{sel}}^{\text{Rf}}, a_{t,\text{exe}}^{\text{Rf}})$ 
15:   $(S_{t+1}, \Delta R_t) \leftarrow \Phi^{\text{Rf}}(S_t', a_t^{\text{Rf}})$ 
16:   $R \leftarrow R \cup \Delta R_t$ 
17:   $S \leftarrow S_{t+1}$ 
18:  if no valid edit is produced by either role then
19:    break
20:  end if
21: end for
22:  $S^* \leftarrow S$ 
```

Stage III: Backdoor-Adjusted Verdict Aggregation

```
23: if  $R = \emptyset$  then
24:    $l_{\text{final}} \leftarrow V(S^*)$ 
25:   return  $l_{\text{final}}$ 
26: end if
27: for  $k = 1$  to  $K$  do
28:   Select or sample  $r_k \sim q(\cdot \mid R)$ 
29:    $l_k \leftarrow V(S^*, r_k)$ 
30: end for
31:  $l_{\text{final}} \leftarrow \arg \max_{l \in \mathcal{L}} \sum_{k=1}^K \mathbf{1}[l_k = l]$ 
32: return  $l_{\text{final}}$ 
```

D Additional Experimental Results

This section reports additional controlled comparisons and diagnostic analyses of ERGA. All cells report Macro-F1 / Accuracy in percentage points unless noted otherwise.

D.1 Comparison with Simpler Refinement Strategies

We compare the full method with prompt-only Explorer–Refiner, SFT-only, and three simpler iterative refinement strategies. All variants use the same evidence access, LLM backend, temperature, downstream verdict procedure, and editing-stage budget. Prompt-only Explorer–Refiner retains the same roles and edit operations but removes the learned policies, while SFT-only retains the supervised policies without PPO. Retrieve–revise, Self–

Refine, and two-agent reflection replace the learned Explorer–Refiner procedure with simpler iterative editing.

As shown in Table 4, ERGA consistently outperforms the prompt-only and simpler iterative variants on both datasets. The comparison with SFT-only further shows the additional contribution of PPO beyond supervised policy initialization.

D.2 Generalization to EX-FEVER

We further evaluate generalization on EX-FEVER (Ma et al., 2024), a multi-hop explainable fact-verification benchmark built from hyperlinked Wikipedia documents. We use the same evaluation subsets for all methods, comprising 500 examples for each of the 2-hop and 3-hop settings.

Table 5 shows that ERGA achieves the best Macro-F1 and Accuracy in both hop settings. The gains on this additional benchmark support the generalization of the method across datasets and reasoning depths.

D.3 Partial-Evidence Diagnostics

We isolate HoVer and FEVEROUS-S examples for which retrieval matches at least one but not all annotated evidence units. These cases capture a failure mode in which locally relevant evidence remains incomplete for the full claim. We additionally construct five post-hoc slices using fixed claim markers: negation or restrictive scope (not/no/only), conjunction (and/both), comparison (than/more), temporal scope (before/after/year), and causal or purpose relations (because/due to/designed to). All variants use identical slices; annotated evidence is used only to define the diagnostic subset and is unavailable at inference. The five slices pool both datasets and may overlap.

All deltas in Table 6 favor the full method across both dataset aggregates and all five diagnostic slices. These group-level results complement the qualitative examples in App. G and show that both game-based state editing and proxy-adjusted aggregation contribute under incomplete-evidence conditions.

E Significance Testing

We conduct exact McNemar tests (McNemar, 1947), implemented as two-sided exact binomial tests on discordant pairs, to test whether ERGA and each baseline have equal paired error rates. This test evaluates paired claim-level correctness

Method	FEVEROUS-S	HoVer
ERGA (SFT + PPO)	82.76 / 82.78	70.91 / 71.01
Prompt-only Explorer–Refiner	81.74 / 81.78	68.82 / 69.01
SFT-only / no-PPO	79.67 / 79.78	69.89 / 70.01
Retrieve–revise	81.72 / 81.78	68.79 / 69.01
Self-Refine	76.72 / 76.78	67.83 / 67.97
Two-agent reflection	75.74 / 75.78	67.21 / 67.36

Table 4: Controlled comparison with simpler refinement strategies. HoVer results are averaged over the 2-hop, 3-hop, and 4-hop settings.

Method	2-hop	3-hop
Causal-CRFV	61.20 / 62.00	58.11 / 59.00
Causal Walk	60.55 / 61.20	60.04 / 60.60
ProgramFC	62.59 / 63.60	55.81 / 58.80
BiDeV	67.92 / 69.20	61.56 / 64.00
VeGraph	71.33 / 72.20	63.32 / 65.40
BOOST	77.27 / 77.60	66.61 / 68.20
ERGA	79.78 / 80.00	71.52 / 72.40

Table 5: Results on the EX-FEVER 2-hop and 3-hop evaluation subsets. Each cell reports Macro-F1 / Accuracy.

rather than Macro-F1; Macro-F1 remains the primary metric reported in the main results. We treat each dataset as a separate hypothesis family because HoVer and FEVEROUS-S are reported and interpreted separately in Sec. 4.2. For HoVer, we pool the 2-hop, 3-hop, and 4-hop examples only for this claim-level paired significance test, while hop-wise performance remains reported in Table 1. For each dataset, we apply Holm–Bonferroni correction (Holm, 1979) over the six baseline comparisons. Let n_{01} denote the number of examples where ERGA is incorrect and the baseline is correct, and n_{10} denote the number of examples where ERGA is correct and the baseline is incorrect. In all comparisons, $n_{10} > n_{01}$, so the corrected p values test whether the observed paired advantage of ERGA is unlikely under equal paired error rates. Significance markers in Table 7 are based on corrected p values: $^*p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

F Prompt Templates

We summarize the prompt templates used by ERGA across state initialization, state editing, reward judging, and verdict aggregation. The templates follow the notation in Sec. 3: the input claim C , sub-claims c_i , evidence E_i , local analyses a_i , refined state S^* , and proxy adjustment conditions sampled from the removed-fragment record R . In implementation, the Evidence and Analysis components are serialized as `evidence_list` and `analysis_list`. Non-

empty U fields produced by Refiner are accumulated in `u_list`, which corresponds to the removed-fragment record R used for proxy adjustment. Explorer-produced auxiliary signals, such as retrieved anchors and stress-test records, are serialized separately in `information_explored_list` as explored context for state construction and trace inspection. Gold labels are used only in the training reward prompts and are never included in inference prompts. Below, we report condensed templates that preserve the original input fields, decision rules, and output constraints used in implementation.

F.1 State Initialization

Decomposition. The decomposition prompt is instantiated as:

You are a fact-checking expert. Decompose the complex claim into distinct, verifiable sub-claims.

Each sub-claim should be: (1) atomic, focusing on one specific fact; (2) verifiable against evidence; (3) non-overlapping with other sub-claims; and (4) complete, so that the full list covers the original claim.

Output one tagged line per unit: [subclaim1]

...

The prompt may include dataset-specific optimization guidelines when available. Its output defines the local verification units $\{c_i\}_{i=1}^n$ used to construct S_0 .

Local analysis. The local analysis prompt is instantiated as:

Diagnostic slice	ERGA	Δ vs. w/o Game	Δ vs. w/o Proxy Adjustment
HoVer aggregate	67.23 / 67.69	+1.29 / +1.65	+1.30 / +1.71
FEVEROUS-S aggregate	73.91 / 79.70	+3.44 / +3.85	+6.05 / +6.96
Negation / restrictive scope	70.32 / 70.44	+4.83 / +4.93	+1.19 / +1.23
Conjunction / composition	72.91 / 73.12	+4.33 / +3.73	+4.57 / +4.51
Comparison / superlative	69.98 / 69.98	+3.37 / +2.26	+3.82 / +3.62
Temporal scope	69.87 / 70.06	+2.40 / +1.39	+2.54 / +2.31
Causal / purpose relation	75.11 / 77.14	+1.33 / +1.43	+6.18 / +7.14

Table 6: Performance on partial-evidence examples. The ERGA column reports Macro-F1 / Accuracy; Δ columns report the corresponding percentage-point gains over each ablation.

Dataset	Baseline	n_{01}	n_{10}	Holm-corrected p
FEVEROUS-S	ProgramFC	266	751	$2.49 \times 10^{-53***}$
FEVEROUS-S	BiDeV	275	336	$1.51 \times 10^{-2*}$
FEVEROUS-S	BOOST	271	350	$3.45 \times 10^{-3**}$
FEVEROUS-S	VeGraph	269	553	$6.09 \times 10^{-23***}$
FEVEROUS-S	Causal-CRFV	296	658	$1.04 \times 10^{-31***}$
FEVEROUS-S	Causal Walk	293	758	$2.45 \times 10^{-47***}$
HoVer	ProgramFC	449	739	$1.78 \times 10^{-16***}$
HoVer	BiDeV	493	849	$1.13 \times 10^{-21***}$
HoVer	BOOST	449	608	$3.37 \times 10^{-6***}$
HoVer	VeGraph	562	656	$7.68 \times 10^{-3**}$
HoVer	Causal-CRFV	727	991	$8.21 \times 10^{-10***}$
HoVer	Causal Walk	735	915	$2.06 \times 10^{-5***}$

Table 7: Holm-corrected two-sided exact binomial McNemar test results. Here n_{10} denotes examples where ERGA is correct and the baseline is incorrect, and n_{01} denotes the reverse.

You are a logical reasoning expert. Analyze each sub-claim in the context of the evidence and the main claim.

For each sub-claim, produce a self-contained, evidence-grounded analysis. The analysis should identify whether the evidence supports the unit, contradicts it, or leaves a missing bridge. Keep the analysis concise and grounded in the provided evidence.

Output one tagged line per unit: [analysis1]
...

The resulting analyses $\{a_i\}_{i=1}^n$ form the Analysis component of the initial verification state.

F.2 Explorer

retrieve. For retrieve, the implementation builds a focused query from the original claim and the state component selected by Explorer. This keeps the claim anchor while adding the selected focus, such as a missing entity, date, role, location, numerical value, comparison base, or purpose condition. When LLM filtering is enabled, retrieved passages are compressed with the following template:

Given the claim, current focus, and retrieved evidence, keep only facts useful for deciding whether the claim is supported or refuted. Prioritize claim-deciding anchors: names, dates, numbers, titles, locations, roles, event relations, and explicit contradictions. If the retrieved passages are only partially useful, keep the useful anchor rather than discarding everything. Deduplicate repeated facts and preserve exact conflicts.

Output: Facts: ...

Verdict Signal: SUPPORT / REFUTE / MIXED / UNCLEAR

Missing Check: ...

Output IRRELEVANT only if the retrieved text has no plausible verification value.

rewrite. For evidence, the rewrite prompt is instantiated as:

Turn the evidence into a short verdict-ready anchor for the claim. Keep only the facts that matter for support versus refute. Preserve exact names, dates, numbers, titles, locations, negations, and contradictions. Remove background, repetition, and generic summary wording. Do not add new facts.

Output: 1. <fact anchor>

2. <fact anchor or contradiction>

Decisive Anchor: <single best anchor>

For analysis, the rewrite prompt is instantiated as:

Rewrite the analysis into a concise verification rationale. Preserve the claim-deciding support/refute logic. Make hidden assumptions, missing links, or contradictions explicit. Do not introduce facts not present in the evidence or explored context.

Output: Rationale: ...

Weak Link: ...

Verdict Direction: SUPPORT / REFUTE / UNCLEAR

counterfactual-analysis. The counterfactual-analysis prompt is instantiated as:

Inspect the current verification state for one fragile condition that could change the verdict. Identify exactly one condition, ambiguity, weakest assumption, or missing link that could change support into refute, or refute into support. If the decisive condition is already explicit, write Failure Mode: None and name the decisive evidence.

Output: Failure Mode: ...

Why It Matters: ...

Needed Verification: ...

This operation identifies a fragile verification condition for later checking; it does not directly change the retained evidence or analysis lists.

F.3 Refiner

merge. The merge prompt is instantiated as:

Merge overlapping or redundant items without losing any claim-deciding fact. Preserve exact names, dates, quantities, negations, comparisons, and contradictions. If two items conflict, keep the conflict explicit rather than smoothing it away. Move removed background or redundant fragments into U.

Output: Refined: <merged item or numbered list>

U: <removed fragment or None>

Change Note: <short explanation>

reduce. The reduce prompt is instantiated as:

Remove noise, repetition, and non-decisive background while preserving all information needed to verify the claim. Do not delete evidence that decides support versus refute. If a removed phrase may still encode a shortcut, ambiguity, or weak condition, place it in U.

Output: Refined: <reduced item or numbered list>

U: <removed fragment or None>

Change Note: <short explanation>

Non-empty U fields are accumulated into the removed-fragment record R .

F.4 Reward Judging and Verdict Prompts

Reward judge. During policy learning, reward prompts score the strategic value of completed state edits. For Explorer, the judge receives the claim, training-only target label, pre-action state, selected action, and generated output:

Score the strategic value of one exploration move. Reward actions that add a missing factual anchor, clarify a weak support/refute link, or identify one concrete hidden condition. Penalize redundant retrieval, generic doubt, hallucination, and unsupported content. Output only a float in $[-1, 1]$.

For Refiner, the judge receives the claim, training-only target label, selected action, original content, refined content, removed fragment, and change note:

Score the strategic value of one refinement move. Reward edits that reduce redundancy or noise while preserving all claim-deciding facts. Penalize unrelated merges, destructive reduction, hallucinated content, and deletion of decisive evidence. Output only a float in $[-1, 1]$.

Final adjudication. The plain adjudication template receives the claim, retained evidence, local analyses, and explored context from the refined verification state:

Based strictly on the provided evidence, analysis, and additional context, determine the truthfulness of the claim. Treat analysis as a helper summary rather than proof. Do not use outside knowledge. Answer strictly with SUPPORT or REFUTE.

Conditional verdict. For Backdoor-Adjusted Verdict Aggregation, the conditional verifier receives the claim C , retained evidence and analyses from S^* , and one removed-fragment proxy condition $r_k \sim q(r \mid R)$ serialized as auxiliary context:

You are a skeptical fact-checker. Assume claims may contain subtle errors until proven otherwise. Cross-check each detail in the claim against retained evidence. Treat Analysis as a helper summary, not proof. Treat Auxiliary Context according to its label. A Stress-Test Hint is adversarial and may be speculative; only use it if the issue is confirmed by retained evidence. If Auxiliary Context conflicts with retained evidence, the evidence wins. If any claim-critical detail is contradicted, unsupported, or fabricated, the verdict must be REFUTE. Output:

REASONING: <2-4 sentence verification>

VERDICT: SUPPORT or REFUTE

This design prevents a removed-fragment condition from directly overriding S^* while still testing whether the final label is stable under proxy adjustment conditions.

F.5 Role-Specific Action Selection

The action-selection prompts are role-specific teacher prompts used to construct SFT trajectories. They are not used as final inference prompts. After SFT, PPO rollouts use the learned Explorer and Refiner policies to output the same discrete action indices.

Explorer selection prompt.

You are an expert strategist for a Fact-Checking Explorer Agent. Your goal is to decide the best next action to verify the claim.

Selection space: $\emptyset$ =analysis_list,
1=evidence_list.

Execution space: $\emptyset$ =retrieve, 1=rewrite,
2=counterfactual-analysis.

Do not default to one habitual pair. Choose evidence_list when the main problem is factual grounding. Choose analysis_list when the main problem is reasoning quality. Choose retrieve only when the state lacks a concrete factual anchor. Choose rewrite when useful material is present but noisy or indirect. Choose counterfactual-analysis when the state may hide a scope, entity, date, comparison, or assumption mismatch.

Output valid JSON with keys: selection, execution, rationale.

Refiner selection prompt.

You are an expert strategist for a Fact-Checking Refiner Agent. Your goal is to decide the best next refinement action.

Selection space: $\emptyset$ =evidence_list,
1=analysis_list.

Execution space: $\emptyset$ =merge, 1=reduce.

Choose evidence_list when factual snippets contain duplicate retrievals, long source background, or conflicting anchors that should be preserved explicitly. Choose analysis_list when reasoning contains redundancy, weak assumptions, or unsupported elaboration. Choose merge when overlapping items should be combined without losing decisive facts. Choose reduce when an item contains removable noise or background.

Output valid JSON with keys: selection, execution, rationale.

G Case Study

Tables 8–10 report representative cases selected from paired prediction outputs. Each row is written from the saved result fields rather than from a post-hoc explanation. We include methods whose outputs contain inspectable process records: ERGA uses evidence_list, analysis_list, information_explored_list, and inference_results; BOOST uses program_code and trajectory; BiDeV uses certain_claim, sub_claims, and verify_results; VeGraph uses graph, verified_graph_triplets, and remaining_subclaims. For baseline systems, we mainly quote the saved intermediate content. For ERGA, we additionally highlight the **retained decisive signal** in S^* , the **separated adjustment context**, and the resulting **stable**

verdict. Boolean saved labels are mapped as true/SUPPORT and false/REFUTE. Rows colored green are correct predictions, and rows colored red are incorrect predictions.

G.1 Case 1: Negated Film-Cast Conjunction

Dataset/id: HoVer / f91ee05f-fcf9-42b1-b3b8-a799041c9861. **Original claim (verbatim):** The Straight Story and Never a Dull Moment do not both Star Dick Van Dyke. **Gold label:** SUPPORT.

G.2 Case 2: Comparative Polarity

Dataset/id: FEVEROUS-S / 35168. **Original claim (verbatim):** Bullenkuhle belongs to this group, albeit considerably bigger, this sinkhole has a diameter of about 130 meters (430 ft) and attains a depth of 15 meters (49 ft). **Gold label:** REFUTE.

G.3 Case 3: Supported Timeline versus Unsupported Purpose

Dataset/id: FEVEROUS-S / 814. **Original claim (verbatim):** The First Nations Governance Act introduced by Robert Nault in Canada in 2002 because he wanted representatives of indigenous groups included in the House of Commons but it was rejected in 2003. **Gold label:** REFUTE.

H AI Assistants Disclosure

We use AI tools for language polishing, grammar correction, and improving English expressions.

Method	Saved output	Saved record and highlighted signal
ERGA	SUPPORT correct	Retained in S^*: evidence_list keeps the decisive cast contrast: <i>Never a Dull Moment</i> stars Dick Van Dyke, whereas <i>The Straight Story</i> (1999) is a separate film not starring him. Separated context: information_explored_list flags entity/title mismatch and asks to verify the cast list of the 1999 David Lynch film. Adjusted verdict: saved conditional predictions remain SUPPORT.
BOOST	SUPPORT correct	program_code: retrieve the casts of <i>The Straight Story</i> and <i>Never a Dull Moment</i> ; verify whether Dick Van Dyke is in each cast; return true unless both checks are true. trajectory: retrieves <i>The Straight Story</i> with Richard Farnsworth, Sissy Spacek, and Harry Dean Stanton.
BiDeV	REFUTE wrong	sub_claims: <i>The Straight Story</i> stars Dick Van Dyke; <i>Never a Dull Moment</i> stars Dick Van Dyke; the two films do not both star Dick Van Dyke. verify_results: [false, true, false].
VeGraph	REFUTE wrong	graph: <i>The Straight Story</i> stars Dick Van Dyke; <i>Never a Dull Moment</i> stars Dick Van Dyke. remaining_subclaims: <i>Never a Dull Moment</i> stars Dick Van Dyke is true, while the other two subclaims remain unresolved.

Table 8: Case 1 result records extracted from saved outputs. The error-sensitive part is the negated conjunction over two film-cast predicates.

Method	Saved output	Saved record and highlighted signal
ERGA	REFUTE correct	Retained in S^*: analysis_list keeps the decisive polarity conflict: the claim says “ considerably bigger ”, while the evidence says “ considerably smaller .” Separated context: generic sinkhole-lake background is kept outside the decisive comparison signal. Adjusted verdict: information_explored_list records Verdict Signal: REFUTE, and conditional verdicts remain REFUTE.
BOOST	SUPPORT wrong	program_code: retrieve Bullenkuhle dimensions; ask diameter, depth, and feature type; verify “Bullenkuhle is a sinkhole with a diameter of about 130 meters and a depth of 15 meters.” trajectory: retrieved Bullenkuhle evidence about sinkhole type, diameter, depth, and sinkhole-lake background.
BiDeV	REFUTE correct	certain_claim: Bullenkuhle belongs to sinkhole lakes and marshes, albeit considerably bigger; Kanlıdivane is located in Mersin Province, Turkey, and has the stated dimensions. sub_claims: Bullenkuhle belongs to sinkhole lakes and marshes; Kanlıdivane is located in Mersin Province; Kanlıdivane has the stated diameter/depth. verify_results: [true, true, true, false].
VeGraph	SUPPORT wrong	graph: Bullenkuhle is a sinkhole; belongs to a group; compared to hidden_entity_1; has diameter 130 meters; attains depth 15 meters. verified_graph_triplets: Bullenkuhle is compared to sinkhole lakes and marshes / Arendsee. remaining_subclaims: sinkhole, group membership, diameter, and depth are all true.

Table 9: Case 2 result records extracted from saved outputs. The decisive contradiction is “considerably bigger” versus “considerably smaller”.

Method	Saved output	Saved record and highlighted signal
ERGA	REFUTE correct	Retained in S^*: evidence_list preserves the supported timeline: Robert Nault proposed the Act in 2002, and it was abandoned in 2003. Decisive mismatch: analysis_list marks the original purpose clause as unsupported: the evidence concerns elections, financial management, administration, and the Canadian Human Rights Act, not House-of-Commons membership. Adjusted verdict: information_explored_list separates the committee-review detail from the claimed House-of-Commons membership; conditional verdicts remain REFUTE.
BOOST	REFUTE correct	program_code: retrieve “First Nations Governance Act Robert Nault 2002 purpose”; ask the primary purpose and 2003 outcome; verify whether the stated purpose and outcome align with the evidence. trajectory: the Act would reform election codes, financial management, and administration; Nault’s stated purpose was to make native reserves more democratic and accountable.
BiDeV	SUPPORT wrong	certain_claim: the Act was introduced in 2002 “to make native reserves more democratic and accountable,” but was rejected in 2003. question_answers: “to make native reserves more democratic and accountable.” sub_claims: introduction in 2002; purpose was democratic/accountable reserves; rejected in 2003. verify_results: [true, true, true].
VeGraph	SUPPORT wrong	graph: Robert Nault wanted hidden_entity_1; hidden_entity_1 to be included in the House of Commons; the Act was introduced because of hidden_entity_2. verified_graph_triplets: Robert Nault wanted representatives of indigenous groups; the Act was introduced because of making native reserves more democratic and accountable; that purpose is linked to including representatives in the House of Commons. remaining_subclaims: proposer, Canada, 2002, and 2003 outcome are true.

Table 10: Case 3 result records extracted from saved outputs. The timeline facts are supported, while the original purpose clause differs from the evidence-supported reform purpose.