ATOMICCHECK: Learning Structured Reasoning for Fact Verification via Protocol-Guided Reinforcement Learning

Xingquan Zhang[1,2], Jianghao Chen[1,2], Yang Zhao[1,2,*], Pu Jian[1,2], Haoran Gao[4], Fanyu Meng[4], Junlan Feng[4], Chengqing Zong[1,2,3,*]

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[2]State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, CAS, Beijing, China
[3]School of Information Science and Technology, ShanghaiTech University, Shanghai, China
[4]JIUTIAN Research, Beijing, China
{zhangxingquan2022,zhaoyang2015}@ia.ac.cn,cqzong@nlpr.ia.ac.cn
{gaohaoran, mengfanyu, fengjunlan}@chinamobile.com

Abstract

Fact verification is a critical primitive for ensuring the reliability of Large Language Models (LLMs). While frontier LLMs are effective verifiers, their repeated long-context inference incurs substantial costs; meanwhile, existing specialized verifiers face a fundamental dilemma: label-only models suffer from black-box opacity, while free-form rationale generators, whether trained by distillation or explanation-oriented RL objectives, can produce fluent but not structurally checkable explanations. In this work, we propose **ATOMICCHECK**, a Protocol-Guided Policy Optimization framework that trains specialized verifiers to produce protocol-guided, locally inspectable traces, moving beyond opaque label-only predictions and free-form rationales. Rather than cloning teacher traces or optimizing unstructured explanations, ATOMICCHECK uses RL to make the verification protocol itself part of the policy objective. To preserve semantic integrity during atomic decomposition, we introduce a Contrastive Decomposition Reward that compares decomposed reasoning with a same-backbone holistic counterfactual and penalizes decompositions that reduce verification accuracy. Extensive experiments demonstrate that ATOMICCHECK outperforms existing specialized verifiers and approaches strong frontier LLMs while producing structured traces that expose local checks behind the final decision.

1 Introduction

Large Language Models (LLMs) have achieved remarkable performance on knowledge-intensive tasks such as question answering and summarization (Zhao et al., 2023; Lewis et al., 2020). However, they remain susceptible to hallucinations, generating content that is factually incorrect or unfaithful to the grounding documents (Maynez et al., 2020; Zhang et al., 2023). Consequently, fact verification has emerged as a critical safeguard, where a dedicated verifier must rigorously determine whether a generated claim is attributable to the reference documents (Kryscinski et al., 2020; Fabbri et al., 2022; Wei et al., 2024; Tang et al., 2024a). For this safeguard to be useful in real-world deployments, it must not only be accurate but also expose an inspectable reasoning artifact that helps developers diagnose where a verifier's decision comes from and where it may fail.

A straightforward solution is to use frontier

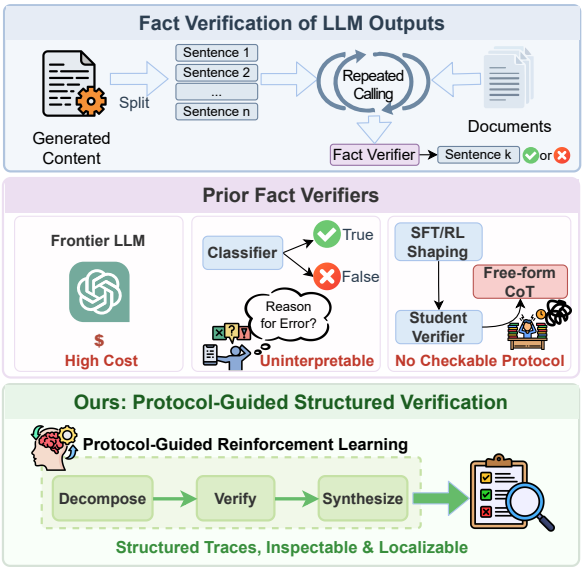


Figure 1: From black-box or free-form verification to protocol-guided verification. Prior specialized verifiers either produce opaque labels or generate free-form CoT without a structured checking protocol. ATOMICCHECK instead trains the verifier to generate structured traces that make local verification decisions inspectable and failures easier to localize.

*Equal corresponding authors.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 4092–4117
November 6-10, 2026 ©2026 Association for Computational Linguistics

LLMs (e.g., GPT-4o (OpenAI, 2024)) as verifiers. While effective (Min et al., 2023; Seo et al., 2025), this approach often requires repeated long-context inference to verify multi-statement responses (Tang et al., 2024a), incurring substantial inference costs that make routine large-scale deployment difficult.

This practical limitation motivates a shift toward specialized small fact verifiers. Existing small verifiers largely fall into two paradigms (Figure 1). (i) **Label-only classification** (Tang et al., 2024a; Lei et al., 2025): models fine-tuned to directly predict a verdict. Although strong in aggregate metrics, they are inherently black-box models, failing to provide inspectable evidence or pinpoint localized failures in partially correct claims. (ii) **Reasoning generation**: models that either imitate teacher CoT (Seo et al., 2025) or use SFT followed by rule-based RL to improve explanations (Si et al., 2025). While these methods improve textual rationales, free-form reasoning lacks explicit local checks, making it hard to verify whether every claim part is covered, where a failure occurs, and whether each step is grounded in evidence. Distillation-based variants further require expensive synthesis of massive reasoning traces and may inherit teacher errors (Gudibande et al., 2024; Agarwal et al., 2024; Liu et al., 2024a; Dai et al., 2025).

To address these limitations, we propose **AtomicCheck**, a framework that moves specialized fact verification beyond opaque label-only predictions (Tang et al., 2024a; Lei et al., 2025) and free-form rationales (Seo et al., 2025; Si et al., 2025) by training verifiers via Protocol-Guided Policy Optimization to follow a structured verification protocol. Rather than treating interpretability as a post-hoc textual rationale, AtomicCheck makes the verification protocol itself part of the policy objective. Specifically, the verifier decomposes claims into checkable units, verifies each unit against evidence, and synthesizes the local judgments into a final verdict. To preserve semantic integrity during atomic decomposition, we introduce a *Contrastive Decomposition Reward*, targeting the "decomposition dilemma" (Hu et al., 2025; Metropolitansky and Larson, 2025): unguided decomposition can fragment claims, omit context, or alter meaning. This reward compares the decomposed trajectory with a same-backbone holistic counterfactual, rewarding decomposition when it corrects holistic errors and penalizing it when decomposition degrades the verdict.

Extensive evaluations demonstrate that AtomicCheck successfully translates this protocol-guided objective into stronger specialized fact verification. On ClearFacts, which spans 14 diverse subsets, AtomicCheck achieves the best overall performance among specialized verifiers and remains competitive with strong frontier LLMs. Moreover, AtomicCheck produces more inspectable verification traces, with structured outputs that expose intermediate claim units, evidence checks, and local judgments behind the final verdict.

2 Related Work

Grounded Fact Verification. Fact verification was originally framed as classifying human-written claims against knowledge bases (Thorne et al., 2018; Wadden et al., 2020). With the rise of Large Language Models (LLMs), the target has expanded to hallucination detection, and methods have shifted toward reasoning-enhanced generative verification (Fabbri et al., 2022; Wang et al., 2024). Despite this shift, recent work suggests a shared core: checking a claim unit against grounding evidence (Seo et al., 2025). Long-form factuality evaluation often decomposes responses into sentences or atomic facts (Min et al., 2023; Wei et al., 2024; Chern et al., 2023; Song et al., 2024), while complex-claim checking similarly verifies subclaims against documents (Pan et al., 2023). We adopt this unified grounded fact-verification formulation (Tang et al., 2024a), treating sentence-level verification as a basic reasoning engine for both human statements and model-generated content.

Fact Verifiers. Frontier LLMs are strong verifiers but costly at scale (Min et al., 2023), motivating specialized verifiers. MiniCheck (Tang et al., 2024a) trains small label-only verifiers on synthetic data and introduces LLM-AggreFact (Tang et al., 2023a; Malaviya et al., 2024); FactCG (Lei et al., 2025) extends this line with graph-based multi-hop training data for complex claims. These models can achieve high benchmark scores, but their label-only outputs remain opaque and still trail strong frontier LLMs on reasoning-intensive benchmarks. ClearCheck (Seo et al., 2025) addresses interpretability by distilling Chain-of-Thought (CoT) capabilities from stronger teachers into smaller students, showing strong gains on ClearFacts, which uses a collection of examples from 14 fact-verification datasets (Tang et al., 2023a; Wadden et al., 2020; Jiang et al., 2020; Jacovi et al., 2024a). FaithLens (Si et al., 2025) combines cold-start

SFT with rule-based RL to improve faithfulness detection and human-readable explanations. However, reasoning-generation verifiers still optimize for free-form rationales rather than a locally inspectable verification procedure, making it difficult to audit claim coverage, failure localization, and evidence grounding, especially under decomposition dilemmas (Hu et al., 2025). Additional related work on verification-oriented decomposition is discussed in Appendix A. **ATOMICCHECK** addresses this gap by shaping *how* the verifier reasons: through protocol-guided reinforcement learning, it makes decomposition, verification, and verdict synthesis part of the learned policy, producing traces that explicitly record which claim units are checked, what evidence supports or refutes each unit, and how local judgments determine the final verdict.

3 Method

3.1 Task Formulation

We formulate fact verification as a conditional generation task. Given a document set $\mathcal{E} = \{d_\ell\}_{\ell=1}^{L}$ and a claim s, the verifier π_θ generates a reasoning trajectory o culminating in a verdict y. The label space is unified as $\mathcal{Y} = \{\mathrm{A}, \mathrm{C}, \mathrm{N}\}$, denoting *Attributable*, *Contradictory*, and *Not Attributable*, respectively. The training set is denoted as $\mathcal{D} = \{(\mathcal{E}_i, s_i, y_i)\}_{i=1}^{N}$. We define a parsing function $g(\cdot)$ that extracts the predicted verdict $\hat{y} = g(o)$ from the output sequence.

3.2 Structured Reasoning Protocol

The core of ATOMICCHECK lies in strict adherence to a verifiable *Atomic Operation Protocol* with constrained structured generation. As illustrated in Figure 2, we model the verification process as a sequence of structured blocks:

(1) **Decomposition** (`<decompose>`): The model decomposes the claim s into m independently verifiable sub-claims $\{c_j\}_{j=1}^{m}$. If s is already atomic, we allow $m = 1$. This step isolates ambiguity and complexity.

(2) **Verification** (`<verify>`): Each sub-claim is verified against the document set $\mathcal{E}$. This block aligns each sub-claim with evidence references or quoted evidence and a local judgment.

(3) **Synthesis** (`<aggregate>` and `<conclusion>`): The model aggregates individual verification results into a coherent explanation and generates the final verdict $\hat{y} \in \mathcal{Y}$.

The protocol is designed to function as a structured inspection trace. The decomposition block defines the units that must be checked; the verification block aligns each unit with evidence and a local judgment; the aggregation block combines these local judgments into a global decision; and the conclusion block exposes a parsable verdict. This parser-visible intermediate state gives the reward computation stable structure for format validation, verdict extraction, and counterfactual comparison, while giving human auditors concrete handles for inspecting claim coverage, evidence use, and aggregation consistency.

3.3 Protocol-Guided Policy Optimization

To train the model to follow the structured reasoning protocol, we introduce Protocol-Guided Policy Optimization. We employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as the underlying optimization objective. For each input $x = (\mathcal{E}, s)$, we sample a group of K outputs $\{o_k\}_{k=1}^{K}$ from the old policy $\pi_{\theta_{\text{old}}}$. The optimization objective is defined as:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{o_k\}_{k=1}^{K}\sim\pi_{\theta_{\text{old}}}}\Bigg[\frac{1}{K}\sum_{k=1}^{K}\Big(\min(\rho_k A_k, \rho_k^{\text{clip}} A_k) - \lambda_{\text{KL}}\mathbb{D}_{\text{KL}}(\pi_\theta||\pi_{\text{ref}})\Big)\Bigg], \tag{1}$$

where $\rho_k = \frac{\pi_\theta(o_k|x)}{\pi_{\theta_{\text{old}}}(o_k|x)}$ is the importance sampling weight, and $\rho_k^{\text{clip}} = \text{clip}(\rho_k, 1-\epsilon, 1+\epsilon)$. Here, ϵ is the clip threshold, λ_{KL} controls the penalty strength of KL divergence, and π_{ref} is the fixed reference policy model. The advantage A_k for the k-th trajectory is computed by normalizing the total reward R_k against the group statistics:

$$A_k = \frac{R_k - \text{mean}(\{R_1, \ldots, R_K\})}{\text{std}(\{R_1, \ldots, R_K\}) + \delta}. \tag{2}$$

Crucially, we guide this optimization via a reward system. To align the model with the structured protocol, we design a reward function consisting of three components:

(a) Outcome Reward (r_{acc}). This sparse reward reflects the correctness of the final verdict, with $r_{\text{acc}} \in \{0, 1\}$:

$$r_{\text{acc}}(o, y) = \mathbb{I}[\hat{y} = y], \tag{3}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

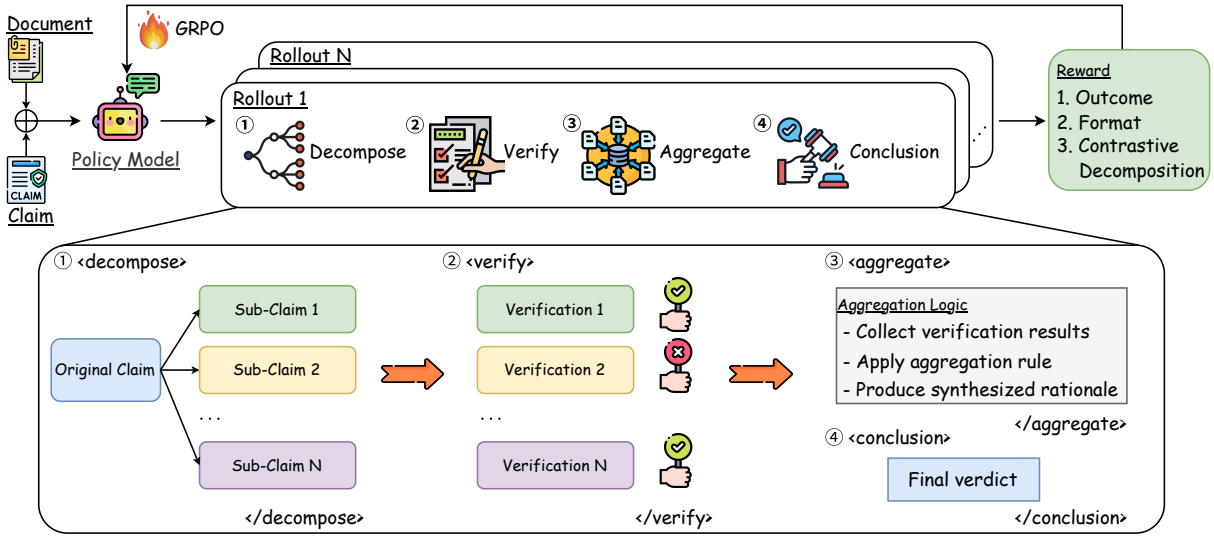


Figure 2: Overview of ATOMICCHECK. By employing Protocol-Guided Policy Optimization with a composite reward system consisting of Outcome Reward, Format Reward, and a novel Contrastive Decomposition Reward, we guide the model to autonomously internalize a structured verification protocol. The process unfolds in three atomic stages: (1) **Decompose** the claim into independently verifiable sub-claims; (2) **Verify** each sub-claim against the document set; and (3) **Synthesize** the results to generate a final **conclusion**.

(b) Format Reward (r_{fmt}). To encourage adherence to the protocol described in Section 3.2, we define a binary format reward $r_{\text{fmt}}(o) \in \{0, 1\}$. It equals 1 if and only if o satisfies constraints: (1) valid usage of allowed tags; (2) correct topological ordering of blocks; and (3) a parsable verdict in the conclusion. This provides a structural validity signal during optimization and discourages invalid shortcuts that emit a verdict without the required intermediate states. Appendix G provides the full validation procedure.

(c) Contrastive Decomposition Reward (r_{cd}). While decomposition enhances interpretability by isolating sub-claims, it also introduces risk. Therefore, decomposition should be favored only when it improves reliability over a holistic check (Hu et al., 2025). To operationalize this principle, we propose a Contrastive Decomposition Reward.

We compare the observed decomposed trajectory against a **same-backbone holistic baseline** o^{hb}—the outcome if the current verifier checked the original claim as a single unit rather than decomposing it into multiple sub-claims. Specifically, we construct o^{hb} by replacing `<decompose>...</decompose>` with `<decompose>`s`</decompose>` (where s is the original claim), then generating the remaining output from the rollout policy $\pi_{\theta_{\text{old}}}$. Using the same backbone makes the comparison a controlled counterfactual: the reward measures the marginal value of decomposition relative to the model's own holistic checking ability, rather than importing an external teacher's bias or rewarding decomposition unconditionally. We then compare the accuracy of the decomposed path against this holistic baseline:

$$r_{\text{cd}} = \begin{cases} +1 & \text{if } r_{\text{acc}}(o) = 1 \wedge r_{\text{acc}}(o^{\text{hb}}) = 0 \\ -1 & \text{if } r_{\text{acc}}(o) = 0 \wedge r_{\text{acc}}(o^{\text{hb}}) = 1 \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

This rewards the model $(+1)$ if decomposition corrects an error that the same-backbone holistic path would have made, and penalizes (-1) if decomposition introduces errors (e.g., via hallucination) where the holistic baseline yields the correct verdict.

Curriculum Stabilization. Since r_{cd} relies on stable format parsing, we introduce a gating mechanism. Let $\bar{r}_{\text{fmt}}^{(t)}$ be the average format reward across all rollouts at step t. We define a gate indicator γ:

$$\gamma = \mathbb{I}[\bar{r}_{\text{fmt}}^{(t)} \geq \tau]. \tag{5}$$

This threshold τ implicitly defines a stabilization period T_{warm}. The final gated reward is:

$$r_{\text{gated}} = \gamma \cdot r_{\text{cd}}. \tag{6}$$

This ensures the decomposition logic is optimized only after the model has mastered the basic structural protocol.

Total Reward. The final reward is a weighted sum:

$$R = (1 - \lambda) r_{\text{acc}} + \lambda r_{\text{fmt}} + \beta r_{\text{gated}}, \quad (7)$$

where λ balances accuracy and format adherence, and β controls the strength of the contrastive shaping signal.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate ATOMICCHECK on the CLEARFACTS benchmark introduced by Seo et al. (2025). CLEARFACTS comprises 14 source subsets spanning diverse domains (e.g., news, dialogue, science) and task types (e.g., RAG, summarization, QA); see Appendix B for details. Following the evaluation protocol of Seo et al. (2025), we report Macro-F1 scores. Consistent with this protocol, we map the outputs of three-way classification models (attributable, not attributable, and contradictory) to a binary label space by merging the latter two classes into the "not attributable" category.

Baselines. We compare against two categories of models: (1) **General LLMs**: comprising **Open-source LLMs** (Llama-3.3-70B-Inst, Llama-3.1-405B-Inst, Qwen2.5-32B-Inst, Qwen3-32B, and reasoning-distilled models (R1-Llama3.3-70B-Inst, R1-Qwen2.5-32B-Inst) (Guo et al., 2025)) and **Closed-source LLMs** (Claude 3.5-Haiku, Claude-3.7-Sonnet, GPT-4o, GPT-5.4, o1, o3-mini, and Gemini-2.0-Flash); (2) **Specialized Verifiers**: MiniCheck (Tang et al., 2024a), FactCG (Lei et al., 2025), FaithLens (Si et al., 2025), and ClearCheck (Seo et al., 2025). Appendix C provides implementation details for the baselines and training efficiency.

Implementation Details. We initialize our policy π_θ from Qwen3-8B and train it on a composite dataset constructed from the training resources used by MiniCheck (MINICHECK-SYN and ANLI) (Tang et al., 2024a; Nie et al., 2020) and the HoVer training set (Jiang et al., 2020). Appendix D.1 provides the dataset composition and ablation. The model is optimized via GRPO for 192 steps, using a global batch size of 256 and a learning rate of 1×10^{-6}. During training, we set the temperature to 1.0 and generate $K = 5$ rollouts per prompt. We apply a KL divergence coefficient of $\lambda_{\text{KL}} = 0.001$ and set the maximum response length to 8,192 tokens. For the reward coefficients, we set the format reward weight $\lambda = 0.5$ and the adaptive shaping coefficient $\beta = 0.1$. We set the gating threshold $\tau = 0.99$, which corresponds to an empirical stabilization period of $T_{\text{warm}} \approx 5$ steps. The format reward is averaged over all rollout samples at each step. All experiments are conducted on a single node equipped with 8 NVIDIA H100 GPUs. Following Seo et al. (2025), we set the evaluation temperature to 0.6.

4.2 Main Results

Table 1 reports Macro-F1 on the 14 CLEARFACTS subsets, which span standard and dialogue summarization, RAG, open-domain QA, and multi-hop claim verification. ATOMICCHECK achieves the best overall performance among specialized verifiers, reaching 86.6 Macro-F1 compared to ClearCheck (83.8), FaithLens (82.1), MiniCheck (81.2), and FactCG (75.3). This improvement is obtained with a compact backbone—approximately 2% of the parameters of Llama-3.1-405B. ATOMICCHECK also surpasses Llama-3.1-405B (82.0), GPT-4o (83.3), and the reasoning-specialized o1 (85.4), while remaining close to o3-mini (87.0) and GPT-5.4 (87.3). Together with the same-backbone ablations below, these results show that the gains come from combining a structured protocol scaffold, RL internalization, and regulated decomposition.

The advantage becomes more pronounced on reasoning-intensive benchmarks. CoverBench requires bridging non-adjacent evidence and maintaining long-range consistency across multiple constraints. ATOMICCHECK achieves 78.9 Macro-F1 on CoverBench, outperforming GPT-4o (77.7) and substantially exceeding specialized baselines. These results demonstrate the effectiveness of Protocol-Guided Policy Optimization, which transforms complex verification into a sequence of controllable atomic checks: the model decomposes a compound statement into independently verifiable sub-claims, verifies each with explicit evidence alignment, and then synthesizes a final decision based on the collection of local verdicts.

Furthermore, ATOMICCHECK generalizes robustly across diverse verification tasks. It achieves strong performance on subsets covering RAG (RAGTruth: 90.2), generative search and open-domain QA (ClaimVerify: 93.2, FC-GPT: 92.4, ExpertQA: 81.5), and dialogue summarization (Tofu-MediaS: 85.7, Tofu-MeetB: 85.4). Notably, on sci-

Models	AggreFact														
	Agg-CNN	Agg-XSum	Claim Verify	Expert QA	FC-GPT	LFQA	RAG Truth	Reveal	Tofu-MediaS	Tofu-MeetB	WiCE	Cover Bench	HoVer	SciFact	Overall
General LLMs															
Llama3.3 70B Inst	66.7	66.7	73.3	72.5	87.9	70.0	77.3	89.8	72.6	84.7	77.1	72.2	78.1	93.3	78.1
Llama3.1 405B Inst	77.7	75.6	85.6	77.9	88.0	82.6	85.7	92.2	78.7	87.8	78.3	77.0	77.6	93.2	82.0
R1-Llama3.3 70B Inst	72.8	65.8	80.8	78.8	89.1	79.8	82.3	95.5	73.4	86.3	81.8	78.5	80.4	92.2	81.7
Qwen2.5 32B Inst	66.7	69.4	78.8	75.7	90.2	84.3	84.8	93.3	83.2	86.3	83.1	76.3	78.8	92.1	81.2
R1-Qwen2.5 32B Inst	64.4	76.9	89.3	79.5	89.1	82.4	86.8	**97.8**	75.2	85.5	82.1	77.0	79.0	94.4	82.9
Qwen3 32B	75.5	68.1	81.3	79.1	91.3	77.4	84.5	95.5	71.5	87.4	82.1	79.4	81.3	93.2	82.5
Gemini 2.0-Flash	71.4	73.6	86.8	77.4	90.2	77.4	83.1	94.4	76.4	83.2	81.9	77.7	79.0	**95.5**	82.0
Claude 3.5-Haiku	56.9	64.1	73.2	76.2	92.4	80.7	84.0	91.0	77.2	81.3	82.3	79.4	82.9	89.6	79.5
Claude 3.7-Sonnet	79.4	78.1	86.4	83.0	90.2	91.1	**91.4**	92.2	91.2	86.3	**91.1**	79.4	82.9	89.6	86.0
GPT-4o	71.4	80.2	84.3	78.8	91.3	82.4	86.9	96.6	80.4	85.6	83.1	77.7	78.2	94.4	83.3
o1	79.1	81.2	83.0	83.8	91.3	83.6	85.7	95.5	77.0	83.5	85.2	81.9	84.6	94.4	85.4
o3-mini	76.7	**83.3**	86.5	**84.4**	91.3	90.8	86.8	94.4	86.8	87.4	88.9	**84.1**	83.1	**95.5**	87.0
GPT-5.4	**81.5**	73.4	**92.0**	78.3	**95.7**	**94.4**	89.2	93.3	**92.4**	**90.4**	84.4	79.8	**86.1**	87.2	**87.3**
Specialized Verifiers															
FactCG	84.9	73.9	82.3	69.7	73.6	86.6	87.1	84.4	81.2	79.2	64.9	55.4	74.4	77.3	75.3
MiniCheck	78.6	76.7	88.6	78.8	90.2	**93.3**	87.9	**96.7**	80.4	84.0	**88.9**	60.0	76.8	89.8	81.2
FaithLens	**85.4**	79.7	88.5	80.7	91.3	88.7	84.5	93.3	82.4	83.2	84.4	65.1	79.7	88.7	82.1
ClearCheck	84.3	**84.3**	89.6	80.9	91.3	89.9	88.0	94.4	80.8	**87.2**	83.2	66.1	84.3	88.6	83.8
ATOMICCHECK	83.5	77.8	**93.2**	**81.5**	**92.4**	89.9	**90.2**	93.3	**85.7**	85.4	85.5	**78.9**	**86.8**	**93.2**	**86.6**

Table 1: Fact verification results across CLEARFACTS subsets (Macro-F1). We categorize methods into two main groups: General LLMs and Specialized Verifiers. **Overall** denotes pooled Macro-F1 over all CLEARFACTS examples. **Bold** and underline indicate the best and second-best performance within each category.

Setting	AggreFact	HoVer	CoverBench	SciFact	Overall
Prompt only	84.5	82.1	76.9	88.5	83.3
Self-train SFT	85.4	78.5	76.2	84.9	82.9
Teacher SFT	86.0	83.6	74.7	87.2	84.0
Outcome-only GRPO	87.0	84.6	76.9	88.6	85.3
ATOMICCHECK	**87.7**	**86.8**	**78.9**	**93.2**	**86.6**

Table 2: Optimization-isolation summary on major benchmark groups (Macro-F1), comparing prompt-only inference, supervised imitation, outcome-only policy optimization, and the complete protocol-guided RL objective. Teacher SFT abbreviates teacher-distilled SFT. The full per-subset table is provided in Appendix C.1.

entific claim verification (SciFact) and multi-hop reasoning (HoVer), ATOMICCHECK reaches 93.2 and 86.8, respectively, surpassing all specialized baselines. These results suggest that internalizing structured reasoning via our RL framework enables a compact verifier to approach strong frontier LLMs on grounded fact verification.

4.3 Ablation Study

To validate the contribution of each design component, Table 3 reports ablations that remove key elements of ATOMICCHECK. We further summarize the separation between protocol prompting, supervised imitation, generic RL, and the complete protocol-guided RL objective in Table 2; Appendix C.1 gives the full table and construction details.

Effectiveness of Reinforcement Learning. We first compare ATOMICCHECK with a "w/o RL" variant that uses the same protocol-style prompting but does not undergo our RL training. This prompt-only baseline drops to 83.3 overall. To rule out the possibility that the gains come merely from supervised exposure to a structured output format, we add two SFT controls: *Self-Training SFT*, trained on filtered self-generated traces, and *Teacher-Distilled SFT*, trained on Qwen3.5-122B-A10B teacher traces rendered in the same protocol. As summarized in Table 2, these SFT alternatives reach only 82.9 and 84.0 overall. Outcome-only GRPO improves over prompt-only inference (85.3 vs. 83.3), showing that on-policy optimization itself matters. However, it remains 1.3 points below the full method. These results separate three effects: supervised trace fitting is insufficient, generic outcome-only RL is helpful, and the proposed format plus contrastive decomposition rewards provide the remaining gain.

Impact of the Contrastive Decomposition Reward (r_{cd}). The role of r_{cd} is not to reward decomposition unconditionally, but to make decomposition selective. Removing it reduces Overall Macro-F1 from 86.6 to 85.7, with the largest drops on SciFact, LFQA, CoverBench, and ClaimVerify. These subsets stress decomposition in different ways: SciFact and CoverBench often hinge on scientific qualifiers, coverage constraints, or evi-

Models	AggreFact											CoverBench	HoVer	SciFact	Overall
	Agg-CNN	Agg-XSum	Claim Verify	Expert QA	FC-GPT	LFQA	RAG Truth	Reveal	Tofu-MediaS	Tofu-MeetB	WiCE				
ATOMICCHECK	83.5	77.8	**93.2**	**81.5**	**92.4**	**89.9**	90.2	**93.3**	85.7	85.4	85.5	78.9	**86.8**	**93.2**	**86.6**
w/o Decomposition Reward (r_{cd})	87.2	79.8	91.0	80.5	91.3	86.5	89.2	**93.3**	88.5	87.6	85.5	76.0	85.8	87.5	85.7
w/o Format Reward (r_{fmt})	80.2	**79.9**	89.9	79.2	**92.4**	86.5	89.1	**93.3**	84.2	80.7	**88.9**	78.6	86.5	89.8	85.6
w/o Curriculum Stabilization	**90.1**	77.6	89.9	79.6	90.2	88.8	**93.5**	**93.3**	**88.6**	81.8	82.1	**79.0**	85.8	89.8	86.0
w/o Evolving Holistic Baseline Policy ($o^{\text{hb}} \sim \pi_{\text{ref}}$)	88.2	78.3	89.7	79.2	89.1	89.8	90.2	92.2	85.4	**88.9**	87.8	77.2	86.1	89.8	86.0

Table 3: Ablation study of ATOMICCHECK components on Macro-F1 score across all CLEARFACTS datasets. Each row presents the performance of a model trained after removing a specific training phase or component. **Bold** and underline denote best and second-best.

dence combinations, while LFQA and ClaimVerify more often require preserving a specific attribute, caveat, or list element when checking against long evidence. In such cases, the useful signal is not simply to decompose more, but to preserve the original verification target while deciding whether decomposition is helpful. Without the contrastive comparison against a same-backbone holistic path, the policy is guided mainly by final-label correctness and format validity, but receives no direct process-level preference between a faithful split and a split that changes scope or dilutes the failing atom. The gains on some summarization and dialogue subsets suggest that those examples rely more on outcome and format signals than on the decomposition-selection signal. The overall drop indicates that regulated decomposition is especially useful on examples where local target preservation determines the verdict.

Importance of Format Constraints (r_{fmt}). Removing r_{fmt} lowers Overall Macro-F1 to 85.6. The format reward defines a stable, parser-visible output space for RL: it makes verdict extraction reliable and keeps decomposition, verification, aggregation, and conclusion blocks recoverable for downstream reward computation. Without this structural scaffold, rollouts can drift into malformed or partially parseable traces, making both outcome extraction and contrastive comparison less reliable during policy optimization. The drop therefore supports the role of r_{fmt} as a stabilizer for internalizing the structured verification protocol.

Curriculum and Holistic Baseline Stability. The curriculum gate matters because r_{cd} is meaningful only after the model can reliably produce parseable protocol traces. When the gate is disabled, contrastive rewards are applied to early rollouts whose decomposition and aggregation may still be malformed or incomplete, making the GRPO advantage signal noisy. Its lower Overall Macro-F1 (86.0 vs. 86.6) supports the need to delay decomposition-specific rewards until the protocol format is stable. Similarly, replacing the evolving holistic baseline with a fixed reference baseline lowers Overall performance. A fixed reference does not track the policy being trained: after the policy improves, the comparison mainly rewards wins over an outdated holistic checker rather than exposing the policy's current decomposition errors. The evolving same-backbone baseline therefore acts as an adaptive curriculum for deciding when decomposition is actually helpful.

Models	AggreFact	HoVer	CoverBench	SciFact	Overall
$\beta = 0.1$ (Ours)	**87.7**	**86.8**	78.9	**93.2**	**86.6**
$\beta = 0.2$	87.6	84.0	**80.2**	87.4	85.9
$\beta = 0.3$	87.4	86.1	78.9	88.6	86.0
$\beta = 0.5$	86.6	85.0	78.1	89.8	85.3

Table 4: Sensitivity analysis of the contrastive decomposition reward weight β on four major dataset categories (Macro-F1).

Models	AggreFact	HoVer	CoverBench	SciFact	Overall
$T_{\text{warm}} = 5$ (Ours)	**87.7**	86.8	78.9	**93.2**	**86.6**
$T_{\text{warm}} = 25$	87.2	86.8	76.4	**93.2**	85.9
$T_{\text{warm}} = 50$	86.9	**87.5**	78.1	90.9	86.0
$T_{\text{warm}} = 100$	86.5	85.7	**79.4**	90.9	85.6
$T_{\text{warm}} = 150$	86.6	**87.5**	76.0	89.8	85.4

Table 5: Sensitivity analysis of curriculum stabilization duration T_{warm} on major benchmark groups (Macro-F1).

4.4 Impact of Hyperparameters

Impact of Contrastive Decomposition Reward Weight (β). To verify the optimal weight for the contrastive decomposition reward, we experimented with $\beta \in \{0.1, 0.2, 0.3, 0.5\}$. The results in Table 4 show that $\beta = 0.1$ yields the best Overall Macro-F1 (86.6). Larger values reduce the Overall score, confirming our design choice that r_{gated} should serve as an auxiliary shaping signal. A smaller weight allows it to gently guide the model towards acquiring the decomposition capability without overshadowing the primary correctness reward (r_{acc}) or the format constraints. When

β is too large, the model may over-prioritize the decomposition behavior at the expense of outcome accuracy.

Impact of Curriculum Stabilization Duration (T_{warm}). We investigate the effect of the curriculum stabilization duration T_{warm} on model performance. The results in Table 5 indicate that $T_{\text{warm}} = 5$ yields the highest Overall Macro-F1 score (86.6). Longer warmups reduce the Overall score in our sweep. This empirical finding supports the effectiveness of our gating mechanism γ. A minimal warmup phase is sufficient for the model to stabilize its output format in our setting, whereas delaying the contrastive decomposition reward (r_{gated}) appears to withhold useful process guidance. Thus, the default setting of approximately 5 steps offers the best balance between format stability and timely process reward integration in our sweep. Detailed per-subset sensitivity results are provided in Appendix D.2.

4.5 Quantitative Evaluation of Interpretability

We quantitatively evaluate trace quality against rationale-generating specialized verifiers, including ClearCheck and FaithLens, along three trace-inspectability dimensions: (i) *Verification Completeness*, whether all atomic facts in the claim are explicitly checked; (ii) *Localization Precision*, whether the rationale pinpoints the minimal failing sub-claim with evidence (for non-attributable cases); and (iii) *Reasoning Faithfulness*, whether the rationale avoids introducing unsupported details beyond the claim and document.

First, we use DeepSeek-V3.2 (DeepSeek-AI, 2025) for an LLM-as-a-Judge evaluation and strictly select 1163 instances where AtomicCheck, ClearCheck, and FaithLens all predict the correct verdict, focusing the comparison on reasoning quality rather than verdict accuracy. Second, we conduct a three-system blind-ranking human evaluation on 168 balanced examples, sampled as 6 supported and 6 non-attributable cases from each benchmark subset. Human annotators see the same claim, document(s), gold verdict, and three anonymized traces, then rank the traces separately for completeness, localization, and faithfulness. We summarize the rankings using a *ranking score*, which maps first, second, and third place to 2, 1, and 0 points, respectively. Details of the human evaluation are provided in Appendix E, with the annotation guideline shown in Figure 4.

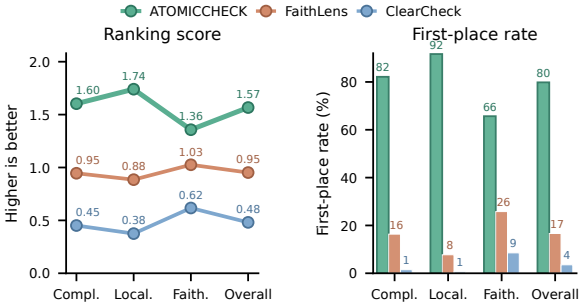


Figure 3: Three-system blind-ranking human evaluation. We report ranking score and first-place rate; Appendix E provides details.

System	Compl.	Local.	Faith.	Avg.
ClearCheck	3.37±1.59	4.40±1.22	4.71±0.97	4.16±0.96
FaithLens	3.98±1.52	4.78±0.81	4.67±0.98	4.48±0.85
AtomicCheck	**4.54**±1.04	**4.89**±0.57	**4.80**±0.80	**4.74**±0.63

Table 6: Three-system LLM-as-a-Judge evaluation of reasoning trace quality (1-5 scale) on instances where all evaluated systems predict the correct verdict. We report mean±std.

As shown in Table 6 and Figure 3, AtomicCheck achieves the highest average score and leads on all three trace-inspectability dimensions against both ClearCheck and FaithLens. The strongest gains appear in completeness and localization, which are exactly the dimensions targeted by the structured verification protocol. The structured trace makes verification obligations explicit by checking claim parts separately and identifying the local source of failure. Appendix F further tests a shared-failure slice where all three systems are wrong, showing that AtomicCheck still provides a more inspectable trace for locating the model's decision basis. Appendix I provides qualitative cases that further illustrate the advantage of AtomicCheck across LLM-AggreFact, HoVer, Cover-Bench, and SciFact.

5 Conclusion

In this work, we proposed **AtomicCheck**, which trains specialized verifiers via *Protocol-Guided Policy Optimization* to follow a structured verification protocol, moving beyond black-box prediction and free-form rationales. By coupling this protocol with a full reward objective featuring a *Contrastive Decomposition Reward*, our approach trains a verifier to reason through structured, locally inspectable intermediate steps and to favor decomposition only when it improves over a holistic

counterfactual. Empirical results show that ATOMICCHECK improves verification accuracy over specialized baselines and produces inspectable traces while approaching strong frontier LLMs.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62476271. We sincerely thank the anonymous reviewers for their valuable feedback and constructive suggestions.

Limitations

Despite ATOMICCHECK's effectiveness, the current work has several limitations. First, our reward design mainly provides explicit process control over decomposition through the contrastive decomposition reward. This keeps reward evaluation rule-based and relatively lightweight, avoiding reliance on costly step-level annotations or additional judge models. However, it leaves finer-grained reward signals for other parts of the structured reasoning trajectory underexplored. Future work can extend protocol-guided RL with denser rewards over the full structured verification trace.

Second, our interpretability evaluation measures whether the structured trace is complete, localizable, and faithful to the provided evidence, but does not establish that the trace is a causally faithful explanation of the model's internal decision process. Verifying step-level faithfulness in the mechanistic sense remains an open direction.

Third, following prior grounded fact-verification work, our experiments focus on textual verification with provided evidence. This setting enables direct comparison with specialized verifiers, but does not cover multimodal evidence or application-specific policies for abstention and error severity. We view these as complementary directions for future end-to-end fact-verification systems.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Rami Aly, Zhijiang Guo, Michael Sejr 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 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Avi Caciularu, Alon Jacovi, Eyal Ben-David, Sasha Goldshtein, Tal Schuster, Jonathan Herzig, Gal Elidan, and Amir Globerson. 2024. Tact: Advancing complex aggregative reasoning with information extraction tools. In *Advances in Neural Information Processing Systems*, volume 37, pages 34775–34799. Curran Associates, Inc.

Hung-Ting Chen, Fangyuan Xu, Shane A. Arora, and Eunsol Choi. 2023. Understanding retrieval augmentation for long-form question answering. *CoRR*, abs/2310.12150.

Jianghao Chen, Wei Sun, Qixiang Yin, Lingxing Kong, Zhixing Tan, and Jiajun Zhang. 2025. ACE-RL: adaptive constraint-enhanced reward for long-form generation reinforcement learning. *CoRR*, abs/2509.04903.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020a. Tabfact: A large-scale dataset for table-based fact verification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, Findings of ACL, pages 1026–1036. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Kenneth Huang, Bryan R. Routledge, and William Yang Wang. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3697–3711. Association for Computational Linguistics.

I.-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. FacTool: Factuality Detection in Generative AI – A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios. *arXiv preprint arXiv:2307.13528*.

Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. 2025. Capture the key in reasoning to enhance cot distillation generalization. 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 441–465. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *CoRR*, abs/2512.02556.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-Based Factual Consistency Evaluation for Summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Arnav Gudibande, Eric Wallace, Charlie Victor Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2024. The false promise of imitating proprietary language models. In *The Twelfth International Conference on Learning Representations*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638.

Qisheng Hu, Quanyu Long, and Wenya Wang. 2025. 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.

Alon Jacovi, Moran Ambar, Eyal Ben-David, Uri Shaham, Amir Feder, Mor Geva, Dror Marcus, and Avi Caciularu. 2024a. CoverBench: A Challenging Benchmark for Complex Claim Verification. *arXiv preprint arXiv:2408.03325*.

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024b. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4615–4634. 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.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2567–2577. Association for Computational Linguistics.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7561–7583. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the Factual Consistency of Abstractive Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Deren Lei, Yaxi Li, Siyao Li, Mengya Hu, Rui Xu, Ken Archer, Mingyu Wang, Emily Ching, and Alex Deng. 2025. FactCG: Enhancing Fact Checkers with Graph-Based Multi-Hop Data. 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 5002–5020, Albuquerque, New Mexico. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pages 7001–7025. Association for Computational Linguistics.

Weize Liu, Guocong Li, Kai Zhang, Bang Du, Qiyuan Chen, Xuming Hu, Hongxia Xu, Jintai Chen, and Jian Wu. 2024a. Mind's mirror: Distilling self-evaluation capability and comprehensive thinking from large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*,

pages 6748–6763. Association for Computational Linguistics.

Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. 2024b. Are llms capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pages 9215–9235. Association for Computational Linguistics.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. ExpertQA: Expert-Curated Questions and Attributed Answers. 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 3025–3045, Mexico City, Mexico. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Dasha Metropolitansky and Jonathan Larson. 2025. Towards Effective Extraction and Evaluation of Factual Claims. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6996–7045, Vienna, Austria. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10862–10878. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.

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.

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.

Zixuan Ren, Yang Zhao, and Chengqing Zong. 2023. Towards informative open-ended text generation with dynamic knowledge triples. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3189–3203, Singapore. Association for Computational Linguistics.

Wooseok Seo, Seungju Han, Jaehun Jung, Benjamin Newman, Seungwon Lim, Seungbeen Lee, Ximing Lu, Yejin Choi, and Youngjae Yu. 2025. Verifying the verifiers: Unveiling pitfalls and potentials in fact verifiers. In *Second Conference on Language Modeling*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *Preprint*, arXiv:2402.03300.

Shuzheng Si, Qingyi Wang, Haozhe Zhao, Yuzhuo Bai, Guanqiao Chen, Kangyang Luo, Gang Chen, Fanchao Qi, Minjia Zhang, Baobao Chang, and Maosong Sun. 2025. FaithLens: Detecting and Explaining Faithfulness Hallucination. *Preprint*, arXiv:2512.20182.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.

Andreas Soularidis, Dimitrios Doumanas, Konstantinos Kotis, and George A. Vouros. 2025. Automating agentic collaborative ontology engineering with role-playing simulation of llm-powered agents and RAG technology. *Knowl. Eng. Rev.*, 40.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023a. Understanding Factual Errors in Summarization: Errors,

Summarizers, Datasets, Error Detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024a. MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Liyan Tang, Igor Shalyminov, Amy Wing-mei Wong, Jon Burnsky, Jake W. Vincent, Yuan Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024b. Tofueval: Evaluating hallucinations of llms on topic-focused dialogue summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 4455–4480. Association for Computational Linguistics.

Rongchuan Tang, Yang Zhao, Chengqing Zong, and Yu Zhou. 2023b. Multilingual knowledge graph completion with language-sensitive multi-graph attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10508–10519, Toronto, Canada. Association for Computational Linguistics.

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.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and Answering Questions to Evaluate the Factual Consistency of Summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

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.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2024. Factcheck-Bench: Fine-Grained Evaluation Benchmark for Automatic Fact-checkers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14199–14230, Miami, Florida, USA. Association for Computational Linguistics.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 80756–80827. Curran Associates, Inc.

Lu Xiang, Yang Zhao, Yaping Zhang, Zixuan Ren, Xingquan Zhang, Zhiyuan Chen, and Chengqing Zong. 2026. An empirical study on reasoning and generalization in large language models. *Computational Linguistics*, pages 1–56.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the AI ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6588–6600. Association for Computational Linguistics.

Junnan Zhu, Min Xiao, Yining Wang, Feifei Zhai, Yu Zhou, and Chengqing Zong. 2025. TROVE: A challenge for fine-grained text provenance via source sentence tracing and relationship classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11755–11771, Vienna, Austria. Association for Computational Linguistics.

A Additional Related Work

Verification-Oriented Decomposition. Decomposition has been used at different granularities and for different verification objects. Long-form factuality metrics split generated answers into atomic or verifiable claims (Min et al., 2023; Wei et al., 2024; Song et al., 2024), summarization metrics use question-answer based consistency checks (Wang et al., 2020; Fabbri et al., 2022), and complex-claim fact-checkers decompose human claims into subclaims, programs, logical predicates, or graph units (Pan et al., 2023; Wang and Shu, 2023; Pham et al., 2025). These methods differ in surface form, but share the same motivation: expose smaller verification targets that can be checked against evidence.

Recent work also shows that decomposition must be controlled rather than applied blindly. Claim-extraction studies emphasize coverage, decontextualization, and faithfulness of the extracted units (Metropolitansky and Larson, 2025), while analyses of Decompose-Then-Verify pipelines show that over-fragmentation, omitted context or key details, and altered meaning can reduce downstream accuracy (Hu et al., 2025). This points to a verification-specific control problem: the system must decide not only how to split a claim, but whether the split preserves the original verification target and improves the final grounded verdict.

ATOMICCHECK is closest in spirit to this selective-decomposition view. It treats subclaims as local audit units and trains a compact verifier to follow a fixed parser-visible protocol. Its contrastive decomposition reward compares decomposed and holistic trajectories from the same backbone, favoring decomposition when it improves verification quality and allowing the model to learn an appropriate level of granularity from the verification process itself.

Viewed in this way, the protocol connects two related lines of work: methods that densify or evaluate intermediate reasoning signals, and systems that organize, select, or trace the external knowledge supporting a generated output (Chen et al., 2025; Xiang et al., 2026; Ren et al., 2023; Tang et al., 2023b; Soularidis et al., 2025; Zhu et al., 2025). ATOMICCHECK specializes this shared design principle to claim-level fact verification by linking decomposition, document evidence, local judgments, and the final verdict within one parser-visible protocol.

B Evaluation Dataset Details

CLEARFACTS is the cleaned fact-verification evaluation set introduced by Seo et al. (2025). It is constructed from a heterogeneous collection of fact-verification benchmarks after filtering unverifiable or near-verbatim instances and correcting detected label errors. The resulting evaluation set keeps a common grounded-verification interface: a statement, one or more reference documents, and an attributable/not-attributable verdict. This standardization enables direct model comparison while retaining the source diversity of the original benchmarks.

Table 7 summarizes the evaluation coverage. Following MiniCheck's description of LLM-AggreFact (Tang et al., 2024a), the first eleven subsets cover fixed-document summarization, retrieval-augmented question answering, post-hoc grounding of answers generated before evidence is supplied, open-domain QA reasoning-chain attribution, and human-written Wikipedia claims with citations. CLEARFACTS further adds CoverBench (Jacovi et al., 2024a), HoVer (Jiang et al., 2020), and SciFact (Wadden et al., 2020), extending the evaluation to complex claims over structured and semi-structured evidence, multi-hop Wikipedia evidence, and scientific abstracts.

The benchmark therefore evaluates more than topical variation. Its evidence ranges from narrative articles and dialogue transcripts to cited web pages, Wikipedia passages, scientific abstracts, and structured or semi-structured sources. The CoverBench portion draws on TabFact (Chen et al., 2020a), FEVEROUS (Aly et al., 2021), FinQA (Chen et al., 2021), PubMedQA (Jin et al., 2019), MultiHiertt (Zhao et al., 2022), HybridQA (Chen et al., 2020b), QRData (Liu et al., 2024b), and TACT (Caciularu et al., 2024), where verification may require reading tables, financial records, structured records, or hybrid text-table contexts. While all subsets are normalized to the same grounded-verification protocol, their original task settings require verifiers to handle different evidence organizations and application domains.

C Implementation Details

C.1 Baseline Details

We compare ATOMICCHECK against a broad range of baselines, classified into General LLMs and Specialized Verifiers.

Subset	Task setting	Domain / evidence form
LLM-AggreFact subsets (Tang et al., 2024a)		
Agg-CNN (Tang et al., 2023a)	News summarization	CNN/DailyMail source articles and generated summaries
Agg-XSum (Tang et al., 2023a)	News summarization	XSum source articles and generated summaries
ClaimVerify (Liu et al., 2023)	Generative search grounding	Generative search-engine responses with associated cited documents
ExpertQA (Malaviya et al., 2024)	Expert-domain QA and attribution	Expert-curated questions across 32 fields; answer claims with verified attributions to supporting evidence
FC-GPT (Wang et al., 2024)	Search-query response checking	Decontextualized atomic facts from LLM responses to search queries with evidence documents
LFQA (Chen et al., 2023)	Long-form question answering	LLM-generated answers to ELI5 questions evaluated against retrieved or selected documents
RAGTruth (Niu et al., 2024)	Retrieval-augmented generation	RAG responses with span-level hallucination annotations across summarization, QA, and data-to-text generation
Reveal (Jacovi et al., 2024b)	Reasoning-chain attribution	Open-domain QA reasoning-chain steps with attribution labels against retrieved Wikipedia passages
Tofu-MediaS (Tang et al., 2024b)	Dialogue summarization	Topic-focused summaries over MediaSum interview dialogues
Tofu-MeetB (Tang et al., 2024b)	Dialogue summarization	Topic-focused summaries over MeetingBank meeting transcripts
WiCE (Kamoi et al., 2023)	Wikipedia citation grounding	Naturally occurring Wikipedia claims with cited documents
Additional benchmarks introduced by CLEARFACTS		
CoverBench (Jacovi et al., 2024a)	Complex claim verification	Diverse complex claims over long, structured, tabular, financial, biomedical, statistical, and hybrid evidence
HoVer (Jiang et al., 2020)	Multi-hop claim verification	Multiple Wikipedia documents per claim
SciFact (Wadden et al., 2020)	Scientific claim verification	Expert-written scientific claims paired with evidence-containing research abstracts

Table 7: Task settings and evidence forms covered by the CLEARFACTS evaluation set.

General LLMs. We evaluate the following models using their specific versions: GPT-4o (`gpt-4o-2024-08-06`), GPT-5.4 (`gpt-5.4`), o1 (`o1-2024-12-17`), o3-mini (`o3-mini-2025-01-31`), Gemini 2.0-Flash (`gemini-2.0-flash`), Claude 3.7-Sonnet (`claude-3-7-sonnet-20250219`), and Claude 3.5-Haiku (`claude-3-5-haiku-20241022`). Following the CLEARFACTS protocol used by ClearCheck (Seo et al., 2025), we use the standard CLEARFACTS prompt for these general LLM baselines.

Specialized Verifiers. We compare against key specialized methods:

- **MiniCheck.** We employ **MiniCheck-7B**[1] (Tang et al., 2024a). Fine-tuned on 35K private synthetic samples generated by Llama-3.1-405B-Inst, it outperforms models of comparable size and reaches GPT-4-level performance.
- **FactCG.** We use **FactCG-DBT**[2] (Lei et al., 2025), the only released version of FactCG. Based on DeBERTa-v3-Large, it is trained on 52K data points including graph-based synthetic claims. Following the official setting, we use a prediction threshold of 0.5.
- **FaithLens.** We evaluate the officially released **FaithLens-8B** checkpoint[3] (Si et al., 2025), a specialized faithfulness hallucination detector trained with supervised fine-tuning and rule-based reinforcement learning. FaithLens starts from 52,268 training candidates: 35,554 SFT candidates synthesized by DeepSeek-V3.2-Think with CoTs and explanations, plus 16,714 verified high-quality challenging examples reserved for RL. The SFT candidates are further filtered to 11,929 examples for cold-start fine-tuning. We use the released

[1] https://huggingface.co/bespokelabs/Bespoke-MiniCheck-7B

[2] https://huggingface.co/yaxili96/FactCG-DeBERTa-v3-Large

[3] https://huggingface.co/ssz1111/FaithLens

Variant	AggreFact														
	Agg-CNN	Agg-XSum	Claim Verify	Expert QA	FC-GPT	LFQA	RAG Truth	Reveal	Tofu-MediaS	Tofu-MeetB	WiCE	Cover Bench	HoVer	SciFact	Overall
w/o RL	82.7	77.0	86.5	76.1	89.0	84.4	86.0	91.1	87.6	82.6	79.8	76.9	82.1	88.5	83.3
Self-Training SFT	79.5	78.5	85.4	76.4	87.9	**90.4**	91.4	91.1	88.1	84.0	78.8	76.2	78.5	84.9	82.9
Teacher-Distilled SFT	**87.6**	67.5	92.1	76.9	83.1	86.7	**92.4**	91.1	**92.4**	**87.0**	80.2	74.7	83.6	87.2	84.0
Outcome-only GRPO	85.8	**80.6**	87.6	78.5	90.1	87.7	90.3	92.2	86.1	84.4	**87.7**	76.9	84.6	88.6	85.3
AtomicCheck	83.5	77.8	**93.2**	**81.5**	**92.4**	89.9	90.2	**93.3**	85.7	85.4	85.5	**78.9**	**86.8**	**93.2**	**86.6**

Table 8: Full optimization-isolation comparison across all ClearFacts datasets (Macro-F1). Self-Training SFT and Teacher-Distilled SFT use correctness- and format-filtered supervised traces; Outcome-only GRPO is trained with r_{acc} but without r_{fmt} and r_{cd}.

Baseline	Generated	Correct	Retained	Overall
Self-Training SFT	53,725	37,650	37,649	82.9
Teacher-Distilled SFT	53,725	35,920	34,856	84.0

Table 9: Construction statistics for the supervised protocol baselines. "Correct" counts trajectories whose parsed verdict matches the gold label; "Retained" further requires valid protocol formatting and is the data actually used for SFT. Overall reports Macro-F1 on ClearFacts.

checkpoint together with the official FaithLens benchmark inference and prediction aggregation pipeline. Since the original FaithLens evaluation focuses on cleaned LLM-AggreFact and HoVer, we replace those benchmark inputs with ClearFacts examples and evaluate the released FaithLens checkpoint under our ClearFacts protocol, extending the comparison to all 14 subsets, including CoverBench and SciFact.

- **ClearCheck.** We evaluate **ClearCheck-8B**[4] (Seo et al., 2025). It is fine-tuned on CoT reasoning traces generated by Llama-3.1-405B-Inst on 57K ANLI examples and 25.2K private synthetic multi-hop data. Seo et al. (2025) report that CoT and direct prediction yield similar performance, but CoT makes the output auditable; we therefore use ClearCheck's CoT outputs for interpretability comparisons.

Supervised Protocol Baselines. Table 8 expands the compact comparison in Table 2. The SFT rows use the same 53,725 prompts as the RL training mixture in Table 12; Table 9 reports the generated, correct, and retained trajectory counts for these supervised baselines. For each prompt, we collect one protocol-style reasoning trajectory, parse the final verdict, and retain only trajectories whose verdict matches the gold label and whose `<decompose>`, `<verify>`, `<aggregate>`, and `<conclusion>` tags form a valid protocol instance. This filtering keeps the SFT data comparable to the structured output space used by AtomicCheck, while removing trajectories that are either incorrect or malformed.

The retained Self-Training SFT set contains 12,974 ANLI, 10,610 MiniCheck-Syn, and 14,065 HoVer trajectories. The retained Teacher-Distilled SFT set contains 12,350 ANLI, 9,057 MiniCheck-Syn, and 13,449 HoVer trajectories; these teacher trajectories are generated by Qwen3.5-122B-A10B. Both baselines use full-parameter SFT for one epoch on the retained trajectories. At evaluation time, both use the same structured protocol template and think-mode decoding as the full model.

C.2 Training Efficiency

The contrastive decomposition reward introduces only a modest training overhead. For each training sample, we generate a single greedy holistic continuation. Under the same single-node 8×H100 setting, the full AtomicCheck RL run takes 4.90 hours, compared with 4.05 hours for the Outcome-only GRPO variant.

Software versions. RL training used Python 3.10 with PyTorch 2.6.0, Transformers 4.57.1, Datasets 4.0.0, FlashAttention 2.8.3, and Verl 0.4.0; final evaluation and inference used vLLM 0.11.0.

D Additional Experiment Results

D.1 Impact of Training Datasets

Table 12 reports the training mixture used for AtomicCheck. The ANLI and MiniCheck-Syn portions follow the training data used by MiniCheck (Tang et al., 2024a); ANLI is originally introduced by Nie et al. (2020). The HoVer

[4] https://huggingface.co/just1nseo/ClearCheck-8B

Models	AggreFact														
	Agg-CNN	Agg-XSum	ClaimVerify	ExpertQA	FC-GPT	LFQA	RAGTruth	Reveal	Tofu-MediaS	Tofu-MeetB	WiCE	CoverBench	HoVer	SciFact	Overall
Full	83.5	77.8	**93.2**	**81.5**	**92.4**	**89.9**	90.2	93.3	85.7	85.4	**85.5**	78.9	**86.8**	**93.2**	**86.6**
w/o MINICHECK-SYN	86.3	**81.2**	89.6	77.9	90.2	88.7	87.0	92.2	84.2	84.2	83.3	**80.1**	86.8	91.0	85.9
w/o ANLI	**86.8**	79.7	88.7	78.7	90.2	**89.9**	90.3	**95.5**	85.4	**88.0**	78.9	79.4	86.5	89.8	86.0
w/o HoVer	84.3	76.2	91.0	80.9	91.3	88.8	**93.5**	**95.5**	**91.2**	85.2	84.4	79.8	84.0	86.3	86.1

Table 10: Detailed ablation analysis of the impact of removing specific training datasets (MINICHECK-SYN, ANLI, HoVer). In each ablation run, the corresponding training component was removed, and the model was trained on the remaining mixture using the same hyperparameters and total training steps as the full model. **Bold** and underline denote best and second-best.

Models	AggreFact														
	Agg-CNN	Agg-XSum	Claim Verify	Expert QA	FC-GPT	LFQA	RAG Truth	Reveal	Tofu-MediaS	Tofu-MeetB	WiCE	Cover Bench	HoVer	SciFact	Overall
$\beta = 0.1$ (Ours)	83.5	77.8	**93.2**	81.5	92.4	89.9	90.2	93.3	**85.7**	85.4	**85.5**	78.9	**86.8**	**93.2**	**86.6**
$\beta = 0.2$	82.9	78.8	86.4	**84.3**	90.1	**91.0**	**93.5**	**94.4**	**85.7**	**86.5**	82.2	**80.2**	84.0	87.4	85.9
$\beta = 0.3$	83.5	**84.4**	92.0	80.7	**94.6**	88.7	88.0	94.4	85.4	79.5	82.2	78.9	86.1	88.6	86.0
$\beta = 0.5$	**85.8**	76.2	89.8	79.6	90.2	88.8	90.3	92.2	**85.7**	82.7	84.4	78.1	85.0	89.8	85.3

Table 11: Detailed sensitivity analysis of the contrastive decomposition reward weight β across all evaluation subsets (Macro-F1).

Source	Split / origin	Examples
ANLI	MiniCheck-released ANLI train subset	21,159
MINICHECK-SYN	MiniCheck synthetic train subset	14,395
HoVer	Original HoVer train split	18,171
Total		53,725

Table 12: Training data composition.

Datasets	AggreFact	HoVer	CoverBench	SciFact	Overall
Full	87.7	**86.8**	78.9	**93.2**	**86.6**
w/o MINICHECK-SYN	86.4	**86.8**	**80.1**	91.0	85.9
w/o ANLI	87.1	86.5	79.4	89.8	86.0
w/o HoVer	**88.1**	84.0	79.8	86.3	86.1

Table 13: Ablation analysis of training datasets on four major categories (Macro-F1). Each row presents the performance of a model trained on the remaining datasets after removing the specified component. Detailed per-subset results are provided in Appendix D.2.

portion is drawn from the original HoVer training set (Jiang et al., 2020).

Split-hygiene audit. After standard text normalization, we check exact overlaps between the training set and CLEARFACTS at the statement level and at the statement–document-pair level; for pair matching, multiple reference documents are concatenated before comparison. We find no exact overlap at either level, indicating that evaluated claims and evaluated claim-evidence pairs are not present in training. The training and evaluation distributions also differ substantially: training documents are mostly short passages (median 113 words; 90th percentile 234 words), whereas CLEARFACTS contains longer and more heterogeneous evidence (median 309 words; 90th percentile 1,343 words), including table-like and multi-line evidence formats that are largely absent from training.

To evaluate the contribution of each component in our composite training dataset (MINICHECK-SYN (Tang et al., 2024a), ANLI (Nie et al., 2020), HoVer (Jiang et al., 2020)), we conducted an ablation study by removing one dataset at a time. In each setting, the model was trained on the remaining mixture while keeping all training hyperparameters and the total number of training steps identical to the full model run.

Table 13 shows that each component dataset contributes to overall generalization: removing MINICHECK-SYN, ANLI, or HoVer reduces Overall Macro-F1 (86.6 $\rightarrow$ 85.9–86.1). The most striking transfer effect comes from HoVer: excluding HoVer causes a large degradation on SciFact (93.2 $\rightarrow$ 86.3, **-6.9**), indicating that multi-hop verification patterns learned from HoVer transfer strongly to scientific claim verification.

D.2 Detailed Experiment Results

This section reports the per-subset breakdown of the experimental results.

Table 10 presents the detailed ablation study, showing the impact of removing specific training datasets (MINICHECK-SYN, ANLI, HoVer) across all 14 evaluation subsets.

Table 11 details the sensitivity of the model to the contrastive decomposition reward weight β,

reporting Macro-F1 for each subset under different β values.

Table 14 shows the detailed impact of the curriculum stabilization duration T_{warm} on the model's performance across the benchmark.

E Human Evaluation Protocol

To strictly assess the interpretability of the reasoning traces, we recruited three student volunteers who are currently engaged in NLP research. Two hold Bachelor's degrees in Computer Science, and one holds a Master's degree. For the three-system ranking study, we sampled 168 instances from the subset where ATOMICCHECK, ClearCheck, and FaithLens all predict the correct verdict: for each of the 14 benchmark subsets, we selected 6 supported and 6 non-attributable examples. Each annotation item contains the claim, reference document(s), gold label, and three anonymized reasoning traces, denoted as Model A, Model B, and Model C. The correspondence between anonymous identifiers and systems is independently randomized for each item to prevent position and identity bias.

Human annotators independently rank the three traces for each evaluation dimension. Rankings may include ties, e.g., Model A $>$ Model C $>$ Model B or Model A $=$ Model C $>$ Model B. Since all systems in this study predict the correct verdict, annotators are explicitly instructed to evaluate the quality of the audit trail. The full annotation guideline is shown in Figure 4.

Aggregation. We convert each dimension-level ranking into a ranking score. A strict three-way ranking assigns 2, 1, and 0 points to the first-, second-, and third-ranked traces, respectively, while tied systems receive the average score of their tied positions. For item i, dimension d, annotator j, and system m, let $s_{i,d,j}(m)$ denote this score. We first average scores across annotators:

$$\bar{s}_{i,d}(m) = \frac{1}{|\mathcal{J}_{i,d}|} \sum_{j \in \mathcal{J}_{i,d}} s_{i,d,j}(m).$$

The reported ranking score for system m on dimension d is

$$\text{RankScore}_d(m) = \frac{1}{N_d} \sum_{i=1}^{N_d} \bar{s}_{i,d}(m).$$

Let $\mathcal{T}_{i,d} = \arg\max_{m'} \bar{s}_{i,d}(m')$ be the set of systems tied for the highest averaged score on item i. The first-place rate is computed as

$$\text{FirstPlaceRate}_d(m) = \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{\mathbb{I}[m \in \mathcal{T}_{i,d}]}{|\mathcal{T}_{i,d}|}.$$

Thus, if two systems are tied for first on an item, each receives 0.5 first-place credit for that item. The overall score is computed as the mean of the three dimension-level scores.

Inter-Annotator Agreement. We report Kendall's τ_b, a rank-correlation coefficient that adjusts for tied rankings. For each pair of annotators, we compute τ_b over the three anonymous systems for each item and metric, excluding pairs where both annotators mark a tie because they contain no directional ordering information. Table 15 reports the mean pairwise τ_b.

F Error-Inclusive Audit-Utility Evaluation

The main trace-quality evaluation conditions on examples where all systems predict the correct verdict. To test whether traces remain useful when verifiers fail, we additionally evaluate the exhaustive *shared-failure* slice of CLEARFACTS, where ATOMICCHECK, ClearCheck, and FaithLens all predict the wrong final verdict. This slice contains 101 examples.

We use DeepSeek-V3.2 as an audit-utility judge. The judge is instructed not to reward final-verdict correctness, but to score whether a trace exposes an inspectable audit surface, localizes the model's decision basis, and reduces the effort needed for a human to verify or repair the decision. The prompt is shown in Figure 11.

Results. Table 16 shows that ATOMICCHECK receives the highest score on every audit-utility dimension in the shared-failure setting. Although all three systems reach an incorrect final verdict, ATOMICCHECK's structured trace is judged easier to inspect, localize, and repair (4.63 average vs. 3.75 for ClearCheck and 4.12 for FaithLens).

G Formal Protocol Constraints (r_{fmt})

We employ a binary format-compliance reward $r_{\text{fmt}}(y) \in \{0, 1\}$ to enforce the structural validity of the verifier's output. The reward is deterministic: $r_{\text{fmt}}(y) = 1$ if and only if the response passes the validation checks described below. The validator focuses on structure and parsability. Table 17 summarizes the key paper-specific notation used for

Models	AggreFact														
	Agg-CNN	Agg-XSum	Claim Verify	Expert QA	FC-GPT	LFQA	RAG Truth	Reveal	Tofu-MediaS	Tofu-MeetB	WiCE	Cover Bench	HoVer	SciFact	Overall
$T_{\text{warm}} = 5$ (Ours)	83.5	77.8	**93.2**	81.5	92.4	**89.9**	90.2	93.3	85.7	85.4	85.5	78.9	86.8	**93.2**	**86.6**
$T_{\text{warm}} = 25$	84.9	76.5	88.6	79.2	**94.6**	89.8	89.1	94.4	81.9	**86.8**	**86.7**	76.4	86.8	**93.2**	85.9
$T_{\text{warm}} = 50$	84.0	**80.9**	87.6	79.4	93.5	86.5	91.3	93.3	86.8	86.1	80.0	78.1	**87.5**	90.9	86.0
$T_{\text{warm}} = 100$	78.2	77.2	91.0	76.8	91.3	86.6	**94.6**	**94.4**	**87.3**	83.7	83.3	**79.4**	85.7	90.9	85.6
$T_{\text{warm}} = 150$	**86.8**	76.4	89.8	**81.8**	91.3	86.4	91.4	93.3	83.0	84.2	82.2	76.0	**87.5**	89.8	85.4

Table 14: Detailed sensitivity analysis of curriculum stabilization duration T_{warm} across all evaluation subsets (Macro-F1).

Human Evaluation Guidelines

Task overview. Given a claim, its reference document(s), the gold verdict, and three anonymized verification traces, rank Model A, Model B, and Model C by the quality of their audit trail. A better trace should be complete, precisely localized when needed, faithful to the provided evidence, and clear enough for a human auditor to verify the decision. Do not reward a trace merely for predicting the correct verdict, since all evaluated systems are correct on the selected examples. Do not use verbosity or formatting style as the primary criterion unless the substantive quality of two traces is otherwise tied.

Evaluation metrics.

1. **Verification Completeness**: Does the trace verify every atomic detail in the claim?
   - Rank higher the trace that explicitly checks all material entities, dates, quantities, relations, conditions, comparisons, and conjunctions.
   - Rank lower the trace that validates only an easy portion of a compound claim, omits a material qualifier, or provides generic evidence without checking claim-specific details.
   - Mark a tie when two traces cover or omit the same substantive facts with comparable specificity.
2. **Localization Precision**: Does the trace pinpoint the minimal failing part of the claim?
   - Rank higher the trace that identifies the exact unsupported or contradicted sub-claim and explains why it fails.
   - Prefer traces that distinguish contradiction from insufficient evidence when the distinction is supported by the document.
   - Rank lower the trace that only states that the claim is not supported, flags the wrong sub-claim, or gives an overly broad failure explanation.
   - For supported claims, mark this dimension as not applicable (N/A) rather than forcing a ranking.
3. **Reasoning Faithfulness**: Is the reasoning fully grounded in the claim and the provided document(s)?
   - Rank higher the trace that uses only document-supported facts and does not overstate what the evidence proves.
   - Rank lower the trace that fabricates evidence, introduces unsupported entities or values, quotes the document inaccurately, or makes intermediate claims that do not follow from the evidence.
   - Generic procedural text, such as stating that the model will check the claim, is not considered a hallucination unless it introduces unsupported factual content.
   - Mark a tie when two traces are comparably faithful or comparably unfaithful.

Figure 4: Annotation guideline for the three-system blind-ranking human evaluation.

Dimension	Kendall's τ_b
Verification Completeness	0.583
Localization Precision	0.566
Reasoning Faithfulness	0.453
Average	0.541

Table 15: Inter-annotator agreement for the three-system blind-ranking human evaluation.

Metric	AtomicCheck	ClearCheck	FaithLens
Audit surface	**4.58**±0.87	3.39±1.41	4.04±1.39
Debug localizability	**4.66**±0.93	3.90±1.59	4.14±1.49
Correction effort	**4.64**±1.04	3.97±1.62	4.17±1.52
Average	**4.63**±0.89	3.75±1.41	4.12±1.41

Table 16: Error-inclusive audit-utility evaluation on the exhaustive shared-failure slice, where all three systems predict the wrong final verdict. Scores are mean±std on a 1–5 scale; bold indicates the best system in each row.

the structured protocol, reward design, and human evaluation.

Allowed Tags. The validator implements a strict whitelist. Any angle-bracket token matching `r"<([^> \s]+)>"` must be one of: `think`, `decompose`, `verify`, `aggregate`, `conclusion`, or their corresponding closing tags (e.g., `/decompose`). Tags are case-sensitive (lowercase required). The presence of any other tag (e.g., `<ref>`, `<Claim1>`) results in rejection ($r_{\text{fmt}} = 0$).

Required Sections. The response must contain at least one non-empty instance of the following blocks: <decompose>, <verify>, <aggregate>, and <conclusion>.

Verdict Normalization. The content of the final <conclusion> block is normalized (stripping brackets, quotes, whitespace, and lowercasing) and mapped to a canonical label. Accepted aliases include:

- **Attributable**: "attributable", "supported", "yes", "true".
- **Contradictory**: "contradictory", "refuted", "false", "no".
- **Not Attributable**: "not attributable", "unsupported", "unverifiable".

If the normalized verdict does not match any alias, the response is rejected.

Flatness Constraint. To ensure parsing robustness, we enforce a strict non-nesting rule. The response must be a sequence of top-level blocks. Specifically: (1) For every allowed tag type, opening and closing tags must be balanced. (2) In the sequence of tags extracted from the text, every opening tag must be immediately followed by its corresponding closing tag. This prohibits nesting (e.g., <verify>...<decompose>...</decompose>...</verify> is invalid).

Reference Implementation. The exact decision procedure is summarized in Algorithm 1.

Algorithm 1 Format-Compliance Validator (r_{fmt})

input: response string y
$allowed \leftarrow$ {think, ..., conclusion} (and closing tags)
$tags \leftarrow$ extract_tags(y)
if has_disallowed_tag($tags, allowed$) **then**
  return 0
end if
$required \leftarrow$ required_blocks()
for $block \in required$ **do**
  $content \leftarrow$ extract_block_contents($y, block$)
  if len($content$) $= 0$ **then**
    return 0
  end if
  if $\forall c \in content$, strip(c) $= \epsilon$ **then**
    return 0
  end if
end for
$verdict \leftarrow$ normalize(last_conclusion(y))
if $\neg$ is_canonical($verdict$) **then**
  return 0
end if
$seq \leftarrow$ extract_all_tags(y)
for $i = 0$ **to** len(seq) $- 1$ **step** 2 **do**
  if is_closing($seq[i]$) **then**
    return 0
  end if
  if $seq[i+1] \neq$ closing_tag($seq[i]$) **then**
    return 0
  end if
end for
return 1

H Prompt Details

A core contribution of ATOMICCHECK is the transition from free-form Chain-of-Thought (CoT) to a strict *Parsable Verification Protocol*. To enable this, we design a structured prompt that enforces an atomic operation workflow. The prompt explicitly defines three XML-like tags—<decompose>, <verify>, and <aggregate>/<conclusion>—which serve as delimiters for the corresponding reasoning stages. This structure enhances the interpretability of the verification process for human auditors.

Below, we provide the full prompt used for ATOMICCHECK and the prompts used for the LLM-as-a-Judge and audit-utility evaluations.

I Qualitative Case Studies

Presentation protocol. For readability, the figures provide source-audited, author-abridged case displays rather than verbatim full inputs and model outputs. These display edits do not change the model predictions, gold labels, cited source values, output structure, or substantive reason why each system succeeds or fails. We apply only presentation-level transformations: (i) we omit any reasoning before </think> and display only the visible answer content after that marker; (ii) on the input side, we excerpt long document or table snippets; (iii) for ATOMICCHECK, we keep the original structured output organization and stage order, reproduce the original visible <decompose>, <aggregate>, and <conclusion> fields, and condense only <verify> to the locally relevant evidence checks; (iv) for ClearCheck and FaithLens, we summarize the visible rationale in the model's own failure pattern rather than reproducing the entire trace; and (v) italicized *Error* or *Analysis* text is our author annotation, not model-generated text.

Temporal alias reversal. As shown in Figure 5, this HoVer case interleaves several supported facts about John Morgan with a reversed naming-history

relation. FaithLens and ClearCheck both verify the surface facts—that John Morgan is a lawyer, that he is associated with the museum, and that the birth date is supported—but they treat the document statement that the museum is "now operated as Alcatraz East" as if it supported the claim that Alcatraz East was a previous name. This is a partial-verification failure: correct atoms mask a local contradiction. In contrast, ATOMICCHECK applies its structured verification protocol to separate the identity, founder, birth-date, and alias-history atoms. The verification step localizes the alias-history atom as contradicted, preventing the supported atoms from dominating the final synthesis and yielding the correct contradictory verdict.

Derivable supported value. Figure 6 presents a CoverBench case with a supported derived value to show that the protocol does not merely make the verifier stricter. The document states that the August 2005 redemption of the 7.875% senior unsecured notes had a total cost of $258.6, composed of the $250.0 principal, accrued interest, and a $1.4 prepayment penalty. The accrued interest is therefore exactly $\$258.6 - \$250.0 - \$1.4 = \$7.2$, which matches the claim's interest-expense component. FaithLens lists the total, principal, and penalty values but concludes that the document "does not break down the exact components"; ClearCheck extracts the total-cost decomposition and then asserts that the document "does not provide" the interest expense. Both therefore reject a supported claim because the target value is not stated literally. ATOMICCHECK performs the implicit subtraction inside the verification block and labels the derived-interest atom as supported, producing the correct verdict. This case shows that the protocol also prevents derivable supported claims from being rejected when the local arithmetic is unambiguously determined by the cited values.

Structured field constraints. Figure 7 highlights partial verification in a LLM-AggreFact structured-field setting. FaithLens collapses the weekday and weekend hours, and ClearCheck extracts the relevant hours but still accepts the statement. The error is not retrieval alone; it is the failure to bind each structured field to the corresponding claim component. ATOMICCHECK decomposes the statement into open-day, weekday-hour, and weekend-hour obligations. This protocol makes the supported open-day component separable from the contradicted weekend time range, so the final decision can be traced to the exact hours field that invalidates the claim.

Scientific causal polarity. Figure 8 covers a SciFact case where the central difficulty is not surface retrieval but scientific polarity. FaithLens matches the terms PKCζ, tumorigenesis, and glutamine metabolism, then treats the document statement that *loss* of PKCζ enhances tumorigenesis as support for the claim that PKCζ itself causes tumor enhancement. ATOMICCHECK decomposes the claim into the causal subject and the metabolic mechanism: the glutamine-metabolism mechanism is supported, but the causal direction is contradicted because the document characterizes PKCζ as a metabolic tumor suppressor. The final verdict is therefore tied to the precise local scientific relation that invalidates the claim.

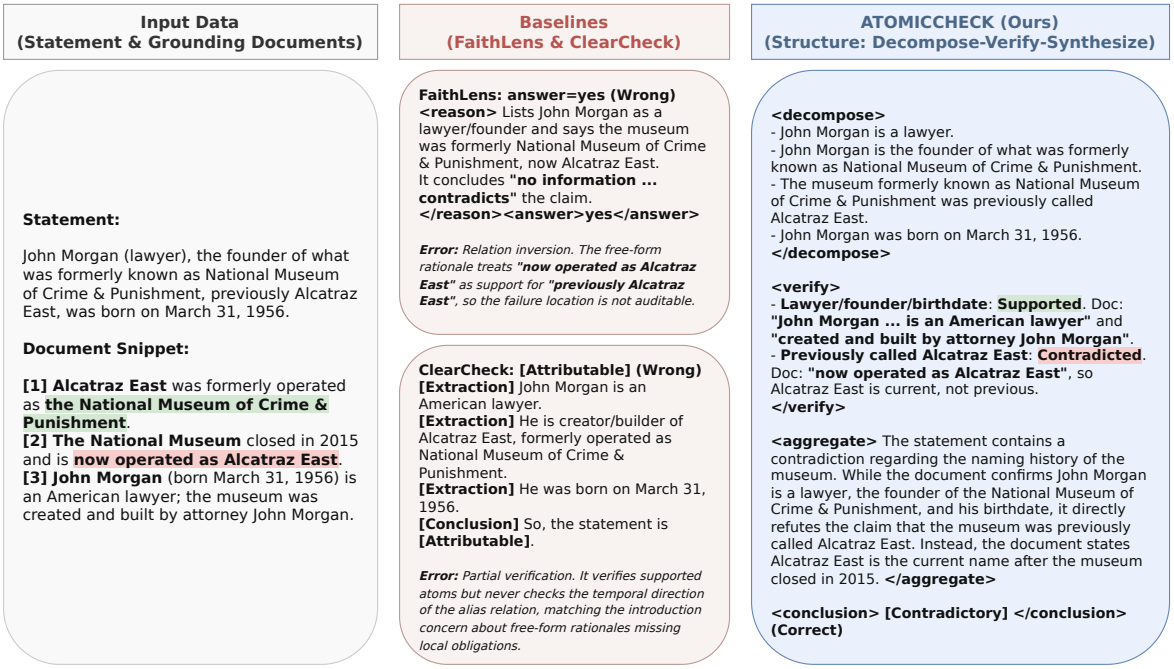


Figure 5: HoVer case: temporal alias reversal. The claim combines several supported facts about John Morgan with an incorrect naming-history relation. ClearCheck and FaithLens verify the supported atoms and accept the claim, while ATOMICCHECK isolates the alias-history atom and identifies that Alcatraz East is the current name rather than a previous one.

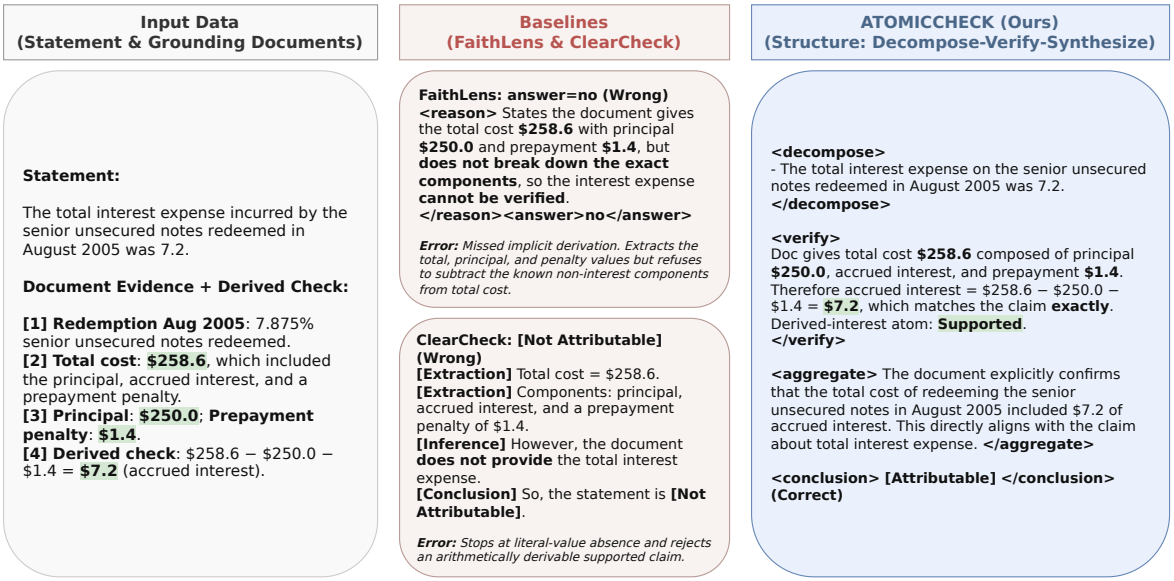


Figure 6: CoverBench case: derivable supported value. The document gives the total cost ($258.6), principal ($250.0), and prepayment penalty ($1.4) for the August 2005 redemption of the 7.875% senior unsecured notes, so the accrued interest is unambiguously $\$258.6 - \$250.0 - \$1.4 = \$7.2$. ATOMICCHECK performs this subtraction inside the verification block and labels the derived-interest atom as supported, while ClearCheck and FaithLens reject the claim because the target value is not stated literally.

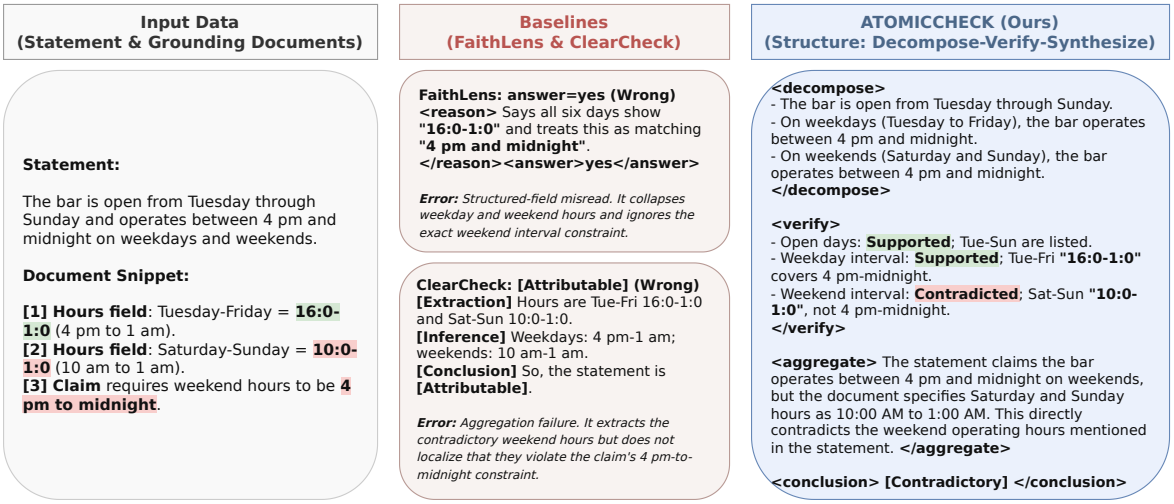


Figure 7: LLM-AggreFact case: structured field constraints. The open-day portion of the claim is supported, but the weekend time range is contradicted by the structured hours field. ATOMICCHECK separates the supported and contradicted sub-claims before producing the final verdict.

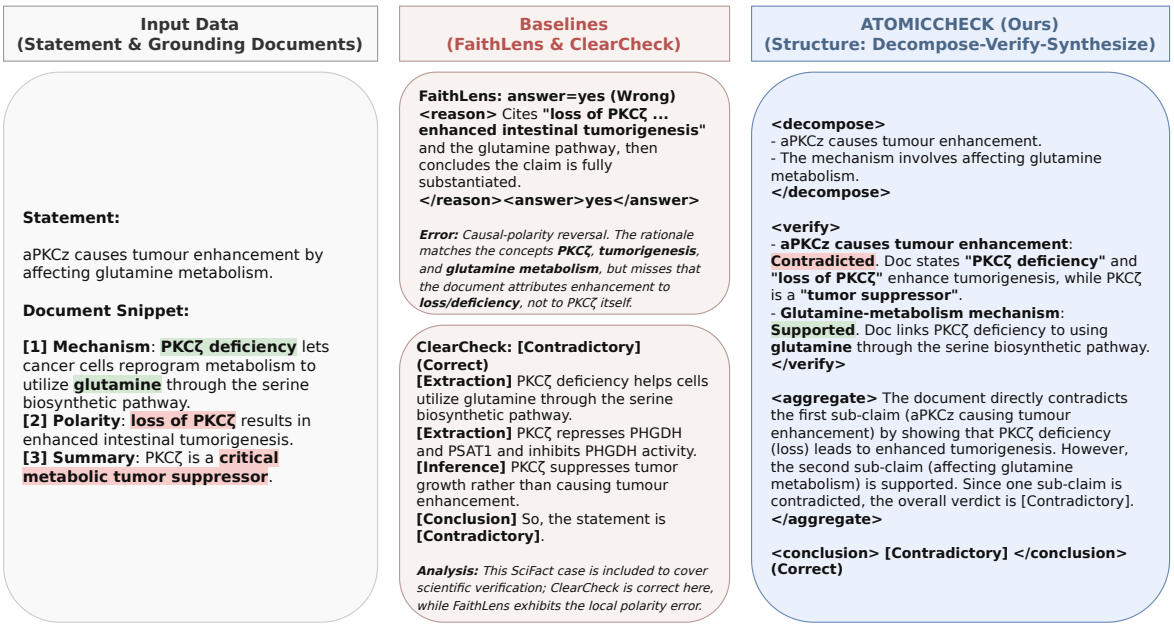


Figure 8: SciFact case: scientific causal polarity. The document links glutamine metabolism to PKCζ deficiency and states that PKCζ is a tumor suppressor. FaithLens matches the overlapping scientific terms but reverses the causal polarity, while ATOMICCHECK localizes the contradicted causal subject and direction.

Prompt used for ATOMICCHECK training and inference

ROLE DEFINITION
You are a meticulous fact verifier. You must base your entire analysis exclusively on the provided **DOCUMENT**. Your reasoning must follow a structured process using these atomic operations:

Step 1: Decompose
Break the STATEMENT into a list of small, independently verifiable sub-claims. Write the output inside **<decompose>** and **</decompose>**. Each sub-claim must be faithful, atomic, and self-contained.

Step 2: Verify
For each sub-claim, analyze the DOCUMENT to determine if it is supported, contradicted, or unverifiable. Write your analysis inside **<verify>** and **</verify>** tags.

Step 3: Synthesize
Once all sub-claims have been verified, construct the judgment:
- First, synthesize the results into a final explanation inside **<aggregate>** tags.
- Then, provide the final verdict inside **<conclusion>** tags.

VERDICT RULES
The verdict must be one of the following:
1. **[Attributable]**: IF AND ONLY IF all sub-claims are supported.
2. **[Contradictory]**: IF any sub-claim is contradicted.
3. **[Not Attributable]**: IF no contradictions, but at least one sub-claim is unverifiable.

INPUT DATA
DOCUMENT: **{document}**
STATEMENT: **{statement}**

Figure 9: Prompt used for ATOMICCHECK training and inference.

Prompt used for all-correct trace-quality judge evaluation

ROLE DEFINITION
You are an expert fact-verification auditor. Evaluate the model's verification trace against the provided document and claim. Focus on trace quality, auditability, and evidence alignment.

METRICS
1. **Verification Completeness**: Did the trace verify every atomic fact in the claim? For this metric, treat an "atomic fact" as a checkable unit in the claim (e.g., entities, relations, numbers, dates, locations, negations, and any conditions/clauses). A generic statement such as "all details match" does NOT count as verifying unaddressed sub-claims.
    - 5 = Explicitly covers all atomic sub-claims with a clear chain (sub-claim $\rightarrow$ document evidence $\rightarrow$ judgment)
    - 3 = Verifies the main proposition but skips secondary qualifiers/conditions or other atomic details
    - 1 = Partial verification error: ignores a key sub-claim that could change the verdict
2. **Localization Precision** (Not Attributable only; label = NS): For NS cases, does the trace pinpoint the minimal failing sub-claim and whether it is contradicted by the document or simply lacks supporting evidence?
    - 5 = Isolates the minimal failing atomic statement, states whether it is contradicted or lacks supporting evidence, and cites auditable evidence with a short quote from the document
    - 3 = Identifies a general area of failure but does not isolate the minimal failing sub-claim and/or lacks an auditable quote
    - 1 = Generic "not attributable/contradicted" conclusion, wrong localization, or no localization
3. **Reasoning Hallucination**: Does the trace introduce incorrect information beyond the claim and the document? Consider two types of hallucination:
    - Claim-side hallucination: introduces details or "error points" that are not part of the claim, treating them as if the claim asserted them
    - Document-side hallucination: fabricates evidence or details not in the document, or misrepresents/overstates what the document supports
    - 5 = No hallucination. Any added context is clearly framed as non-factual (e.g., a checking plan, definitions, uncertainty such as "may/might") and is not used as factual justification
    - 3 = Minor hallucination. States a small number of unsupported factual details, but they are peripheral and not used to justify the verdict; they do not add key atomic facts (numbers/dates/entities/relations/negations/conditions)
    - 1 = Severe hallucination. Adds unsupported or incorrect key facts, fabricates/misrepresents evidence, or relies on hallucinated details to justify the verdict

IMPORTANT
- Use ONLY the provided document; do not use outside knowledge
- Focus on reasoning quality, not the final label; even if the prediction matches the ground-truth, penalize missing sub-claims or hallucinated content
- If the ground-truth label is Supported (S), set localization to 5 and write "N/A (Supported)" in the reason
- Keep the reason brief (1-2 sentences)

INPUT
- Document: {document}
- Claim: {claim}
- Ground-truth Label: {label}
- Model Verification Trace: {trace}

OUTPUT JSON ONLY
```
{ "completeness": int, "localization": int, "faithfulness": int, "reason": "brief explanation" }
```

Figure 10: Prompt used for the all-correct trace-quality judge in Table 6. This judge evaluates completeness, localization, and faithfulness while holding final-verdict correctness fixed.

Prompt used for error-inclusive audit-utility judge evaluation

ROLE DEFINITION
You are an expert fact-checking audit evaluator. Your task is to evaluate whether a model's verification trace is useful as an audit/debugging artifact for a human reviewer.

IMPORTANT DISTINCTION
- Do **not** simply reward the trace for having the correct final verdict.
- A trace with a wrong final verdict can still receive a high audit score if it exposes the model's own decomposition, local evidence checks, decisive local judgment, and likely failure point clearly enough for a human to inspect or correct it.
- Conversely, a trace with a correct final verdict can receive a low audit score if it is opaque, generic, or hard to localize.
- Use only the provided document and claim. Do not use outside knowledge.

METRICS
1. **Audit Surface**: Does the trace expose inspectable intermediate structure?
   - 5 = clear units/subclaims, local checks, final synthesis/conclusion, and an auditor can identify what was checked
   - 3 = some reasoning is visible, but units/local checks are incomplete or hard to separate
   - 1 = mostly opaque, generic, or only a final answer
2. **Debug Localizability**: If a human disagrees with the model's final verdict, can they quickly locate the exact trace step or local claim that caused the decision?
   - 5 = decisive local step is explicit and easy to inspect, even if it is wrong
   - 3 = broad reason is visible but the decisive local step is ambiguous
   - 1 = no clear way to locate the model's failure or decision basis
3. **Correction Effort**: How easy would it be for a human to correct or verify the trace?
   - 5 = small, localized edit/check would repair or confirm the decision
   - 3 = requires reading much of the trace/document
   - 1 = requires reconstructing the reasoning from scratch

OUTPUT JSON ONLY
```
{"audit_surface": int, "debug_localizability": int, "correction_effort": int,
"reason": "brief 1-2 sentence explanation"}
```

INPUT
- Document: {document}
- Claim: {claim}
- Ground-truth Label: {label}
- Model Verification Trace: {trace}

Figure 11: Prompt used for the error-inclusive audit-utility judge in Table 16. This judge evaluates audit surface, debug localizability, and correction effort without rewarding final-verdict correctness.

Symbol	Description
	Structured Verification Protocol
o	A generated structured verification trajectory.
`<decompose>`	Protocol block that splits the original claim into checkable sub-claims.
`<verify>`	Protocol block that verifies each sub-claim against the provided evidence.
`<aggregate>`	Protocol block that combines local verification results.
`<conclusion>`	Protocol block that emits the final parsed verdict.
	Reward Functions
r_{acc}	Outcome reward indicating whether the parsed verdict matches the gold verdict.
r_{fmt}	Format reward indicating whether the trajectory satisfies the structured protocol constraints.
r_{cd}	Contrastive decomposition reward comparing a decomposed trajectory with a holistic baseline.
o^{hb}	The same-backbone holistic baseline trajectory that checks the original claim without decomposition.
$\bar{r}_{\text{fmt}}^{(t)}$	The average format reward over rollouts at training step t.
γ	The gate indicator that activates contrastive decomposition reward after format stabilization.
r_{gated}	The gated contrastive decomposition reward, $\gamma \cdot r_{\text{cd}}$.
R	The final weighted reward used for policy optimization.
	Reward and Curriculum Hyperparameters
λ	The coefficient balancing outcome accuracy and format adherence in the total reward.
β	The coefficient controlling the strength of the contrastive shaping signal.
τ	The threshold used to activate the curriculum gate γ.
T_{warm}	The empirical warmup duration before decomposition-specific reward is activated.
	Human Evaluation
$s_{i,d,j}(m)$	The ranking score assigned to system m for item i, dimension d, and annotator j.
$\bar{s}_{i,d}(m)$	The annotator-averaged ranking score for system m on item i and dimension d.
$\text{RankScore}_d(m)$	The mean ranking score of system m on evaluation dimension d.
$\text{FirstPlaceRate}_d(m)$	The fraction of examples where system m is tied for or assigned first place on dimension d.
τ_b	Kendall's rank-correlation coefficient used to report inter-annotator agreement.

Table 17: Summary of key paper-specific notation.