Think Like Human: Enhancing Large Language Models via Multi-reward Reinforcement Learning for Fact Verification

Anonymous ACL submission

Abstract

Fact verification aims to verify the truthfulness of claims and statements. Recent LLM-based fact verification methods employ prompt-tuning or supervised fine-tuning mechanisms to achieve this goal. However, fact-checking tasks necessitate that models comprehend and analyze the complex relationships between claims and evidence to make veracity judgments, yet LLMs lack such reasoning capabilities to a certain extent, leading to suboptimal performance in handling fact-checking tasks. In this paper, we propose a novel **T**hought-enhanced **F**act **V**erification framework via Reinforcement Learning (TFV-RL) for fact verification by aligning LLM to fact-checkers' thinking process via reinforcement learning. We design a novel Multi-Reward mechanism (Multi-R) that designs four rewards to integrate fact-checking objectives into the reinforcement learning process and align LLMs with fact-checkers' thinking process better. The experimental results on three datasets demonstrate that our method has obtained the best performance. Besides, we analyze the impact of different backbones and different training methods, discovering that TFV-RL can align LLMs with fact-checkers' thinking process better. It enables the model to simulate the thinking process of fact-checkers during fact-checking, making more accurate judgments and generating reasoning.

1 Introduction

With the rapid development of social media, the widespread dissemination of misinformation has become a critical problem for all internet users. Thus, fact verification is designed to tackle this challenge, aiming to verify the truthfulness of claims and statements with relevant evidence (Kotonya and Toni, 2020; Cao et al., 2025).

Traditional fact-checking methods utilize small-scale models and integrate semantic information via attention-based mechanisms to assess the veracity of claims (Wang, 2017; Chen et al., 2021; Gu et al., 2022; Wang et al., 2021). These traditional approaches have achieved good performance in the field of fact verification. Along with the rapid advancement of Large Language Models (LLMs), LLMs demonstrate a strong capability of understanding and thinking, which encourages fact verification methodologies to shift from small model-based approaches to those leveraging LLMs. Recent LLM-based studies follow primarily two directions: the first improves fact verification performance by retrieved-augmented Generation mechanisms that retrieve and filter more useful evidence (Yue et al., 2024; Zheng et al., 2024); the second improves fact-checking capabilities during post-training inference through prompt engineering techniques (Cao et al., 2025; Kanaani, 2024).

However, both approaches merely enable models to mimic examples during inference, rather than genuinely learning the thinking processes of fact-checkers, thus leading to overly brief responses or logical thinking errors. As shown in Table 1, compared to human-annotated explanation and thinking processes, the thinking processes generated by LLMs are significantly disappointing in terms of both length and complexity. Even provided with relatively complex examples in question prompts, LLMs can address only a single point of contradiction by mimicking the form of the given example and have the problem of logical redundancy, whereas human-annotated thinking processes can comprehensively address all points of contradiction. Hence, how to align large language models with human fact-checkers' reasoning processes poses a significant challenge in the field of LLM-based fact verification.

To tackle this problem, we propose a novel **T**hought-enhanced **F**act **V**erification framework via **R**einforcement **L**earning (TFV-RL) for fact verification by aligning LLM to fact-checkers' thinking process via reinforcement learning (RL). Specifically, we first collect human-annotated

	Thinking	Length	Evaluation
fact-checker	Jeffrey Epstein reached a plea deal with federal prosecutors in 2007 when George W. Bush was president and Alex Acosta was the ... Epstein agreed to plead guilty to state prostitution charges, register as a sex offender, serve 18 months in county jail and pay monetary damages to his victims. In exchange, ... A judge ordered the agreement unsealed in 2009.	61	Provides comprehensive historical context and specific dates spanning 2007-2009, establishing robust chronological conflict.
LLM	The claim conflicts with the given evidence. Specifically, the claim conflicts with the evidence. Specifically, Jeffrey Epstein reached a plea agreement in 2008, which was during the George W. Bush administration (2001–2009), not under the Obama administration (2009–2017).	38	Detailed factual correction with specific dates and administrative context, despite *logical redundancy*.

Table 1: An example of a human-annotated thinking process and LLM-annotated thinking process. The claim we used is **The Jeffrey Epstein plea agreement "was a sweetheart plea deal that was made**. *Italic texts* demonstrate the problems in thinking processes.

claim-evidence pairs with fact-checkers' thinking process. Then, to construct thought-augmented training data, we utilize GPT-4 (OpenAI, 2023) as the thinking decomposer and filter to decompose the whole thinking process into multiple steps via logical relations. For the RL process, we design a novel Multi-Reward mechanism (Multi-R) to modify the traditional RL method and align it with the objectives of fact verification tasks. By incorporating prediction reward, semantic reward, structural reward, and strong-to-weak(S2W) transfer reward, we not only preserve the capability of enhancing the diversity of language generation but also integrate fact-checking objectives into the RL process.

We conduct experiments in three public datasets, including FEVER (Thorne et al., 2018), HOVER (Jiang et al., 2020), and SciFact (Wadden et al., 2020). The main results demonstrate that TFV-RL outperforms other methods, both SLM-based methods and LLM-based methods, in fact verification tasks. Additionally, we observe that different backbone models exhibit variations in comprehension capabilities. However, after training with the TFV-RL framework, its capability in handling fact-checking tasks has been significantly improved. Compared to the DPO method, the TFV-RL approach better aligns the thinking processes of LLMs with those of human fact-checkers. Besides, experiments between traditional reinforcement learning and our Multi-R approach demonstrate that Multi-R enables better alignment of LLMs with fact-checkers' thinking processes, making it better adapted to fact verification tasks.

Our major contributions are as follows: (1) We propose a novel TFV-RL framework designed to align the thinking processes of LLMs with those of human fact-checkers, thereby enhancing the models' comprehension of the relationship between claims and evidence and improving their fact-verification capabilities. (2) We propose a novel Multi-Reward mechanism that leverages reinforcement learning with fused reward functions to better align LLMs with human fact-checkers' reasoning processes. (3) The experiment results on three datasets demonstrate that our model outperforms the traditional fact verification methods. Further analysis on backbones and training methods demonstrates that our proposed method enables better alignment of LLMs with fact-checkers' thinking processes.

2 Related Work

2.1 Fact Verification

Research on fact verification typically involves verifying text-only claims using textual evidence, such as metadata of the claim, documents retrieved from knowledge bases, or tabular evidence (Wang, 2017; Chen et al., 2021; Gu et al., 2022; Wang et al., 2021; Gong et al., 2024; Zhang et al., 2024; Wu

et al., 2025; Barik et al., 2025). Wang (2017) incorporated additional metadata as external evidence to verify claims. Chen et al. (2021) constructed entity graphs to acquire more detailed data representations. Zhang et al. (2024) focused on the multi-hop problem and proposed a causal walk network to prevent the model from being degraded by shortcuts. Gu et al. (2022) converted tabular evidence into a sequential format through serialisation and integrated it with the claim for validity determination. Gong et al. (2024) constructed a heterogeneous graph to integrate textual and tabular evidence. With the development of LLMs, LLM-based fact verification has become a hotspot research field (Cao et al., 2025; Zheng et al., 2025; Yue et al., 2024). These approaches leverage various claim-evidence interaction methods to deal with text-only fact verification and demonstrate satisfactory performance on unimodal fact verification. However, these methods overlook the divergence in thinking processes between LLMs and human fact-checkers. This oversight can lead to logical or factual errors in the fact-checking tasks, thereby compromising the model's performance.

2.2 Reinforcement Learning

Reinforcement Learning from Human Feedback (RLHF) methods tailored for LLMs were introduced. This approach enables LLMs to align their final outputs with fact-checkers behaviour. This alignment is manifested in linguistic style, grammatical format, and certain thinking capabilities (Schulman et al., 2017; Rafailov et al., 2023; Wu et al., 2024; Zeng et al., 2024). Derived from traditional reinforcement learning methods, the PPO method introduced by Schulman et al. (2017) utilises a reward model to score positive and negative samples separately, thereby aligning the model with fact-checkers' preferences. Building upon PPO, Rafailov et al. (2023) proposed the DPO method, which treats the LLM itself as an implicit reward model, eliminating the need for an explicit reward model. Zeng et al. (2024) focused on token-level preference and they proposed a TDPO method.

These RL methods improve the performance of LLMs. However, they are designed for generative tasks like question answering, multi-round dialogue, which cannot be directly applied to the fact verification task, which demands more specialized and meticulous thinking processes. Therefore, traditional RL methods are incapable of effectively aligning with the reasoning processes of human fact-checkers, resulting in suboptimal performance on fact verification tasks.

3 Methodology

Task Definition Fact verification aims to verify the truthfulness of the given claim or statement. Specifically, given a claim or statement C and its relevant evidence E, the final goal is to find a function $f : (C, E) \rightarrow (P, I)$ that can obtain the prediction of the label P and the interpretable text of the thinking process I.

To tackle the aforementioned problems, we propose a novel method named TFV-RL to combine different types of rewards by employing reinforcement learning methods on fact verification tasks. As shown in Figure 1, TFV-RL has two stages: (1) the thinking-augmented training data construction stage that collects and preprocesses data for model training, and (2) the reinforcement learning with multi-reward stage that aligns the model with fact-checkers' thinking process through 4 types of rewards. In this section, firstly, we aim to offer the task definition. Then, we will introduce these two components in detail.

3.1 Thinking-Augmented Training Data Construction

In the data construction phase, we primarily collect and preprocess relevant training data, involving the following procedures: data collection, thinking process decomposition, and consistency filter.

Data Collecting This process aims to collect available fact verification data with human-annotated labels and thinking explanations. Specifically, we collect fact-checking data from public website PolitiFact [1] for the period 2023-2025 using open-source tools. Each data point contains a claim that requires verification, its relevant evidence, and a human-written thinking process. To ensure the high quality of training data, we perform subsequent filtering of the compiled dataset. Inspired by Aly et al. (2021), we remove claims whose length is shorter than 10 words, for they contain little information to be fact-checked. Besides, we abandon those claims with only one piece of evidence, because they can be easily predicted even if the model doesn't have fact-checker-like thinking capability.

[1] https://www.politifact.com/

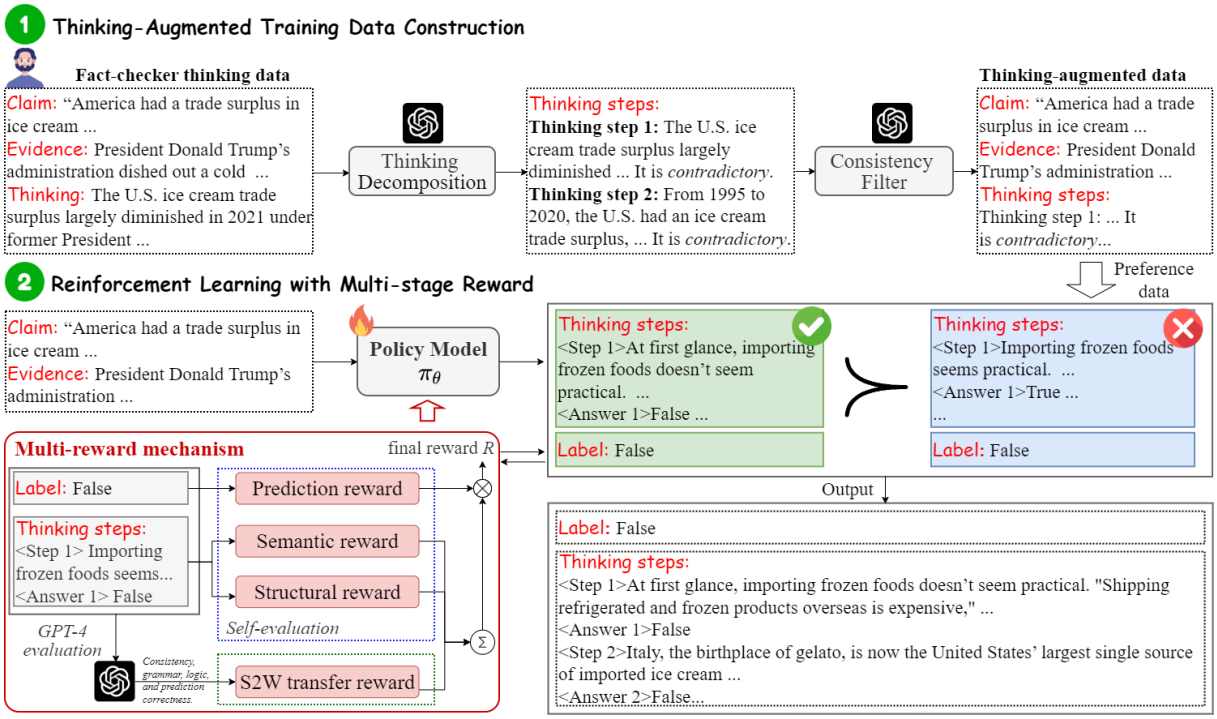


Figure 1: The framework of our proposed TFV-RL method. It has two components, the data construction component and the reinforcement learning component.

Thinking Process Decomposition This process is designed to facilitate the calculation of the semantic reward. Specifically, we leverage the highly capable GPT-4 (OpenAI, 2023) to decompose all human-annotated thinking processes within the dataset at the sentence-level decomposition. Specifically, the LLM first extracts atomic facts from the claim. The LLM divides each sentence of the thinking process into multiple reasoning steps based on the extracted atomic facts, and assigns fine-grained veracity labels to the atomic facts involved in each step. After this process, we obtain a dataset that contains both complete thinking text and decomposed thinking steps.

Consistency Filter Since we decompose the complete thinking text into several steps, it is hard to ensure the consistency of the decomposed step set. Hence, we propose a Consistency Filter module to verify alignment between the decomposed text and the source. We employ the GPT-4 (OpenAI, 2023) as the filter. The module assigns consistency scores to each decomposition, enabling iterative refinement based on these metrics. Decompositions are deemed correct when their consistency score exceeds a predefined threshold θ; otherwise, the system performs automated re-decomposition until it satisfies θ. After these three procedures, we obtain the training data for both the RL stage.

3.2 Reinforcement Learning with Multi-Reward

In this process, we train models via RL to acquire fact-checker-like thinking patterns, thereby achieving model alignment. Specifically, we utilize the collected thinking-augmented training data as the training input to the the policy model π_θ. π_θ is expected to generate the thinking process $S^m = S_1^m, S_2^m, \cdots, S_{|S^m|}^m$ and its prediction $A^m = A_1^m, A_2^m, \cdots, A_{|A^m|}^m$. Based on the output, we calculate the rewards via reward model, and the parameters are updated by:

$$\mathcal{J}(\theta) = \mathbb{E}_{(x,y)\sim\pi_\theta}[r(x,y)] - \beta\mathbb{D}_{KL}[\pi_\theta(y|x)||\pi_{ref}(y|x)]. \quad (1)$$

π_θ denotes the model to be optimized, and π_{ref} denotes the reference model. $r(x,y)$ demonstrates the reward function.

During the RL phase, we propose the novel Multi-R mechanism to construct pair-wise preference data and facilitate its application in fact-checking tasks. First, we allow LLM to generate M samples, each of which contains thinking steps and predicted label. Then we design four types of rewards to reconstruct the traditional reward function and select chosen and refused training data

from the samples. The composite reward function integrates four components: (1) prediction reward, (2) semantic reward, (3) structural reward, and (4) S2W transfer reward.

Prediction Reward This is designed due to the ultimate goal of the fact verification task. Inspired by Jin et al. (2025), we calculate the prediction reward R_p of each output by:

$$R_p = \sum_{c=1}^{C} P_{i,c} \log(\hat{P}_{i,c}) = \begin{cases} 1, \text{if prediction is correct} \\ 0, \text{else} \end{cases} . \quad (2)$$

C denote the number of categories. If the R_p value is 0, directly mark this sample as refused data. If the prediction result is true, then calculate the subsequent reward score for this sample.

Semantic Reward The semantic reward primarily aligns model capabilities with fact-checkers' thinking processes. We first compute the weighted similarity between each step of the model's thinking trajectory and the ground-truth thinking chain. These per-step similarities are aggregated via summation and normalisation to yield the step-wise semantic reward R_{sem}:

$$R_{sem} = \log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_i^+)/\tau)}{\sum_{j=1}^{K} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j^-)/\tau)}. \quad (3)$$

$\mathbf{z}_i$ denotes the output thinking process of π_θ. $\mathbf{z}_i^+$ and $\mathbf{z}_i^-$ represent the ground-truth thinking process and other thinking process in the same batch. τ denotes the model temperature.

Structural Reward Due to the heterogeneous stylistic patterns in pretraining corpora, LLMs employ diverse syntactic structures during generation. To regularise output formats, we design a structural reward R_{str} defined as:

$$R_{str} = \frac{1}{|S^m|} \sum_{i=1}^{|S^m|} \left(\frac{\text{SentSim}(S_i^m \cdot S_i^h)}{d_i^m d_i^h} \right) + \frac{1}{|A^m|} \sum_{i=1}^{|A^m|} \sum_{c=1}^{C} A_{i,c} \log(\hat{A}_{i,c}). \quad (4)$$

$|S^m|$ and A^m are the length of the decomposed step set and the length of the decomposed answer set, respectively [2]. S_i^m and S_i^h are model-generated and human-written i-th step. d_i^m and d_i^h are word length of model generation and human annotation. SentSim is the sentence similarity obtained by SentenceBert (Reimers and Gurevych, 2019).

[2]It is easy to prove that $|S^m|$ equals to A^m

Dataset	Train	Dev	Test
FEVER	145,449	19,998	19,998
2-hop HOVER	9,052	1,126	1,333
3-hop HOVER	6,084	1,835	1,333
4-hop HOVER	3,035	1,039	1,333
SciFact	809	300	300

Table 2: The statistics of FEVER and HOVER datasets.

S2W Transfer Reward Given that the capabilities of the policy model are inherently constrained, the performance of models trained solely based on it remains limited. To address this, we designed an S2W transfer reward R_{S2W} to transfer knowledge from more capable models to the policy model, thereby enhancing its performance. Specifically, the output of the model to be trained is fed as input into the GPT-4 (OpenAI, 2023) and GPT-4 generate a comprehensive feedback. Each evaluation encompasses four dimensions: CONSISTENCY, GRAMMAR, LOGIC, and PREDICTION CORRECTNESS. The evaluations from the two models are processed through a multi-layer perceptron (MLP) to balance the performance across these four metrics, yielding the corresponding S2W transfer reward.

After obtaining these four types of rewards, the final reward R is calculated by:

$$R = \frac{1}{3}(R_{sem} + R_{str} + R_{S2W}). \quad (5)$$

After calculating the final reward scores R, we set all samples that exceed the threshold α as chosen data, and all other data as refused data. We then combine each piece of chosen data with each piece of refused data to construct a preference dataset. For the RL phase, we modify the reward function by the four types of rewards and integrate it with the DPO (Rafailov et al., 2023) approach and employ it to train LLMs. With the Multi-R process, the preference data pairs becomes better aligned with the objectives of fact-checking, effectively facilitating the alignment between the thinking processes of LLMs and fact-checkers.

4 Experiment Settings

Datasets We experiment on three public fact verification datasets, FEVER (Thorne et al., 2018),

Model	FEVER		2-hop HOVER		3-hop HOVER		4-hop HOVER		SciFact	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SLM-based methods										
DeBERTa	65.32	63.17	62.60	60.42	61.14	59.03	60.88	58.84	78.84	79.02
EvidenceNet	72.87	69.84	71.00	68.84	70.08	67.91	68.92	66.98	81.95	82.40
SaGP	74.18	71.63	73.92	70.77	73.46	70.24	73.09	69.87	83.27	84.51
CO-GAT	75.90	72.85	74.25	70.81	73.63	69.97	72.68	69.23	82.36	82.97
Causal walk	76.92	74.28	75.90	72.61	74.16	71.50	73.74	70.69	84.88	85.47
LLM-based methods										
CHECKWHY	73.05	70.58	73.52	70.87	72.94	69.83	72.66	69.31	83.25	83.90
HiSS	74.97	71.32	74.28	71.44	73.75	70.64	72.86	70.03	81.07	82.43
RAFTs	77.05	75.19	76.54	74.25	75.38	73.67	74.69	72.18	84.85	85.38
TFV-RL	**81.93**	**80.15**	**78.87**	**75.13**	**77.75**	**74.39**	**76.88**	**73.79**	**85.49**	**86.85**

Table 3: Results of the fact verification task. Acc and F1 denote the Accuracy and MACRO F1 scores, respectively. 2-hop, 3-hop, and 4-hop HOVER denotes that we report the result of the performance on 2-hop, 3-hop and 4-hop HOVER datasets. **Bold** denotes the best performance and Underline denotes the second-best performance.

HOVER (Jiang et al., 2020), and SciFact (Wadden et al., 2020). The statistics are shown in Table 2 [3].

Baselines We compare it to several fact verification approaches to assess the performance of TFV-RL, including five small-model-based methods and three LLM-based methods. **DeBERTa** (He et al., 2021) utilises the pre-trained DeBERTa model for textual feature extraction to generate predictions. **EvidenceNet** (Chen et al., 2022) selects salient sentences from document-level evidence, employing attention mechanism for label prediction. **SaGP** (Si et al., 2023) employs an adversarially perturbed graph neural network to identify rational subgraphs for joint prediction and explanation generation. **CO-GAT** (Lan et al., 2024) leverages sentence-level evidence aggregation coupled with GAT mechanisms to substantiate claims. **Causal walk** (Zhang et al., 2024) applies a two-stage graph fusion method to reduce the impact of shortcuts and make predictions. **CHECKWHY** (Si et al., 2024) use CoT-based methods to prompt-tune LLMs for fact verification. **HiSS** (Zhang and Gao, 2023) leverages prompt-optimised LLMs for claim decomposition and cascaded subclaim validation to generate predictions. **RAFTs** (Yue et al., 2024) utilise a RAG-based method to retrieve more relevant evidence and an argument-based verification mechanism to make predictions based on GPT-3.5 (Ouyang et al., 2022).

Implementation Details We use 2 Tesla V100-PCIE GPUs with 32GB memory for all experiments and implement our model via the Pytorch framework. We choose Llama3.1-8B (Grattafiori et al., 2024), Qwen2.5-7B (Team, 2024), and Qwen3-8B (Team, 2025) as our backbones. The threshold θ is set to 0.8. The smoothing hyperparameter δ is set to -0.5. The threshold α is set to 0.8. The batch size is 4. The number of training epochs of SFT is 3, and that of RL is 3. We set the learning rate to 1e-5. Significantly, we train LLMs only utilising the data we collected from PolitiFact. To make the comparison fair enough, we also use this data to train the baseline models.

Evaluation metrics For the fact verification tasks, we use Accuracy (Acc) and Macro F1 (F1) scores as the evaluation metrics.

5 Results and Discussion

5.1 Overall Results

We conduct experiments on FEVER, HOVER, and SciFact datasets, and the results are demonstrated in Table 3. Rows 1-5 present benchmark results from small-scale models, rows 6-8 display large-scale model verification outcomes, and the final three rows demonstrate the performance of our proposed TFV-RL approach. Compared to small-model-based approaches, TFV-RL achieves significant performance. This indicates that aligning models with human fact-checkers' thinking processes effectively unlocks their potential for fact verification tasks to some extent. Moreover, TFV-RL demonstrates significant performance superiority over other LLM-based approaches. This further demonstrates that, for fact-checking tasks, reinforcement learning-based alignment delivers stronger empirical results than prompt-tuning or supervised fine-tuning (SFT) methods. Overall, TFV-RL outperforms the baseline models, demonstrating its effectiveness and superiority.

[3] More information of datasets can be seen in Appendix A

Model	FEVER		2-hop HOVER		3-hop HOVER		4-hop HOVER		SciFact	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
TFV-RL	**81.93**	**80.15**	**78.87**	**75.13**	**77.75**	**74.39**	**76.88**	**73.79**	**85.49**	**86.85**
–w/o prediction reward	79.05	76.28	76.85	73.88	75.24	72.50	74.38	71.14	82.00	82.85
–w/o structural reward	81.17	80.01	78.49	74.71	77.10	74.68	76.76	73.18	83.59	84.15
–w/o semantic reward	80.47	79.64	77.84	74.43	76.42	74.00	76.05	73.23	84.33	85.08
–w/o S2W transfer reward	80.69	79.97	78.16	74.63	76.95	73.85	76.49	73.48	85.00	85.87

Table 4: Results of the ablation study. Acc and F1 denote the Accuracy and MACRO F1 scores, respectively. **Bold** denotes the best performance. We exclusively present ablation studies based on Qwen3-8B.

Model	FEVER		Avg. HOVER		SciFact	
	Acc	F1	Acc	F1	Acc	F1
Qwen2.5-7B	74.20	73.75	70.84	68.94	79.92	80.68
–w DPO	76.11	75.68	74.37	71.79	82.88	83.41
–w Multi-R	80.00	78.66	76.52	73.64	84.36	85.58
Llama3.1-8B	74.93	73.85	70.89	69.15	80.00	80.95
–w DPO	78.81	76.19	74.90	71.20	84.17	85.00
–w Multi-R	80.93	80.01	77.26	73.53	85.29	86.53
Qwen3-8B	76.33	74.32	72.05	69.93	80.08	81.12
–w DPO	79.63	78.20	74.58	72.05	83.80	84.47
–w Multi-R	**81.93**	**80.15**	**77.57**	**74.14**	**85.49**	**86.85**

Table 5: The result of module analysis on the FEVER and 4-hop HOVER dataset. Avg. HOVER denotes that we report the average result based on the performance on 2-hop, 3-hop and 4-hop HOVER datasets. *w/o Training*, *w DPO*, and *w Multi-R* represent without training, training by DPO, and training by Multi-R, respectively.

Model	Time	Improvement
Qwen3-8B	$\times 1$	0
TFV-RL–Qwen3-8B	$\times 1.2$	$+5.09$
DeepSeek-R1	$\times 25$	$+4.42$

Table 6: Results of the time cost. To facilitate comparison, we normalised inference times relative to the vanilla model's latency (set as 1 unit of time). We also benchmarked the performance of the other two models against Qwen3-8B model as the baseline. Results reflect mean values across trials.

5.2 Ablation Study

We also conduct ablation experiments, and the results are shown in Table 4. Experimental results demonstrate that all four designed rewards impact model performance, with the prediction reward exerting the most substantial influence. This predominance stems from the core objective of RL and fact-checking: accurate verification of claim veracity labels. Consequently, the prediction reward constitutes the most critical component. Removing the structural reward yields only marginal performance degradation, indicating that enforcing output format constraints has a limited impact on overall capability. The model's performance also degraded after removing the S2W transfer reward, suggesting that the S2W transfer reward may help mitigate the discrepancy between traditional RL and the fact-checking task to some extent. According to the results, four types of TFV-RL rewards are crucial to improve the performance of LLMs.

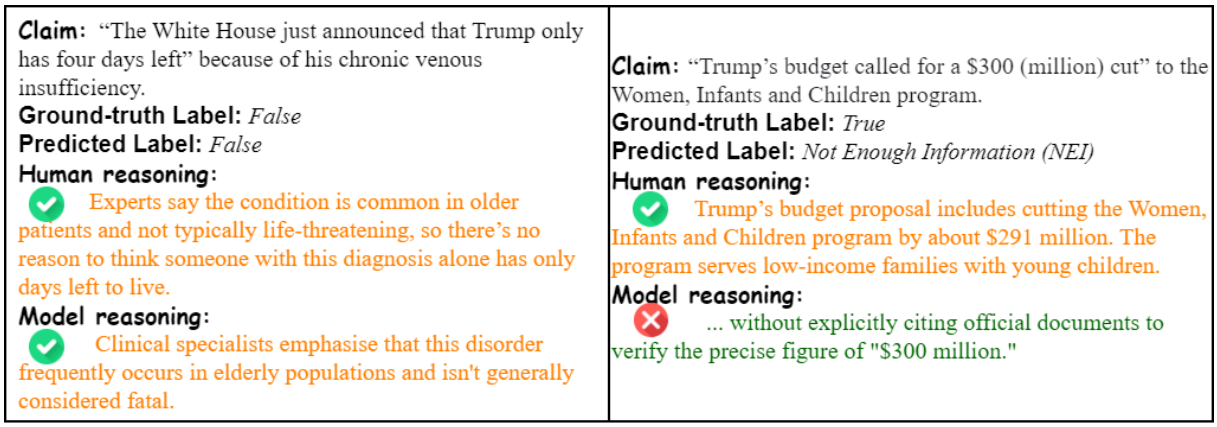
5.3 Thinking Alignment Analysis

We conduct more experiments to further investigate the effectiveness of the Multi-R mechanism of TFV-RL.

Impact of LLM backbones We first compared the performance differences between base models trained with the Multi-R method and those without. Experimental results are presented in Table 5. The results reveal that untrained models exhibit poor performance, with Table 3 indicating they underperform even some small-model-based approaches. This suggests that during pre-training, the fact verification capability of LLMs remains underutilised and underdeveloped. After Multi-R training, all LLMs demonstrated significant performance improvements. This indicates that the Multi-R method effectively elicits and enhances the fact verification capabilities previously lacking in LLMs.

Impact of training methods Concurrently, we compared the performance of conventional DPO and Multi-R. The results are also presented in Table 5. The experimental results reveal that while the conventional DPO method improves LLM performance on the fact verification task, its enhancement over the baseline is limited compared to Multi-R. In addition, the results indicate substantial variations in performance gains between different training methods when applied to the same model on the same task. Consequently, the effective integration of existing advanced RL methods into specific task frameworks poses a major challenge for leveraging LLMs across diverse downstream applications [4].

[4]We also conduct experiments to investigate the impact of data scale. Due to the page limit, we report the results in



(a) Correctly classified example (b) Wrongly classified example

Figure 2: Two cases from PolitiFact correctly predicted by TFV-RL. The orange part demonstrates the fact-checker-like thinking process. The green part shows a case of hallucination.

5.4 Efficiency analysis

Given our constraint requiring models to output thinking steps, we experimentally compared the inference latency of: (1) the vanilla model, (2) the model fine-tuned with TFV-RL, and (3) the R1 (DeepSeek-AI et al., 2025) model with advanced thinking capabilities. Results are presented in Table 6. Multi-R maintains near-identical inference latency to the vanilla model while achieving a 20x speedup over the R1 model, indicating that TFV-RL introduces no additional computational overhead during thinking and output generation. Furthermore, performance comparisons reveal that TFV-RL outperforms the vanilla model by 5.09% and surpasses R1 by 0.67% on fact verification metrics. These results collectively demonstrate TFV-RL's capability to deliver significant performance gains while preserving low-latency operation.

5.5 Case study

To further validate that TFV-RL enables models to acquire fact-checker-analogous thinking processes, we analyze two cases to demonstrate how the TFV-RL framework aligns LLM with fact-checkers' thinking while revealing both capabilities and limitations. (see Figure 2). The model correctly predicted case (a), matching fact-checkers' reasoning that relied on medical consensus about the condition's non-critical nature in elderly patients, and maintained structural alignment in key segments (highlighted). The S2W transfer reward penalized inconsistent hallucinations while rewarding logical medical reasoning. In case (b), fact-checkers identified specific budget details ($291M cut) and context, whereas the model was inconsistent—demanding official documents while ignoring substantial media evidence. This reveals a flaw in the S2W consistency measure, as the model over-prioritized documentation over contextual understanding. Despite grammatical accuracy and partial logical structure, this reasoning gap reduced prediction accuracy.

6 Conclusion

We propose TFV-RL, an LLM-based method to align the model reasoning process with fact-checkers' thinking for improving fact verification performance. we leverage human-annotated fact verification datasets and corresponding thinking processes as training data to incorporate fact-checkers' thinking processes into LLMs through reinforcement learning. To better align LLMs with fact-checkers' thinking, we also design a new multi-reward incorporating four rewards to construct preference data of high quality. Experiments on three public datasets demonstrate that TFV-RL is effective enough to improve LLMs' capability of fact verification through multiple benchmarks. Further analysis shows that our method can align LLMs with fact-checkers' thinking process and improve the performance of LLMs on fact verification tasks.

Limitations

The TFV-RL method significantly improves LLM-based fact verification by leveraging multi-stage fact-checkers' thinking to construct a hierarchical reward mechanism. However, its sophisti-

Appendix C.

cated design introduces notable limitations. First, scalability and annotation cost present a major practical challenge. TFV-RL requires collecting and labelling high-quality, fine-grained fact-checkers' thinking processes (procedural steps) to build the intermediate reward signals, rather than relying solely on the final judgment outcome. Acquiring this procedural data is significantly more time-consuming and expensive than conventional outcome-based annotation, thus restricting the method's practical application to new, massive-scale benchmarks. Second, the framework introduces considerable optimization complexity. Achieving optimal performance depends critically on delicately modulating the information entropy of the procedural rewards to strike a balance between the diversity and accuracy of the model's thinking outputs. This modulation relies on sensitive hyperparameters that require extensive, resource-intensive tuning across different base LLMs and fact verification domains. Future work should investigate methods for automatically distilling high-quality procedural thinking from existing resources to mitigate annotation costs and explore more adaptive, domain-agnostic approaches for reward mechanism design.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, and et al. 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*.

Anab Maulana Barik, Wynne Hsu, and Mong-Li Lee. 2025. Chronofact: Timeline-based temporal fact verification. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 8031–8039.

Han Cao, Lingwei Wei, Wei Zhou, and et al. 2025. Enhancing multi-hop fact verification with structured knowledge-augmented large language models. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 23514–23522.

Chonghao Chen, Fei Cai, Xuejun Hu, and et al. 2021. An entity-graph based reasoning method for fact verification. *Information Processing & Management*, 58(3):102472.

Zhendong Chen, Siu Cheung Hui, Fuzhen Zhuang, and et al. 2022. Evidencenet: Evidence fusion network for fact verification. In *WWW*, page 2636–2645.

DeepSeek-AI, Daya Guo, Dejian Yang, and et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Haisong Gong, Weizhi Xu, Shu Wu, and et al. 2024. Heterogeneous graph reasoning for fact checking over texts and tables. In *AAAI*, pages 100–108.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Zihui Gu, Ruixue Fan, Xiaoman Zhao, and et al. 2022. Opentfv: An open domain table-based fact verification system. In *SIGMOD*, page 2405–2408.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and et al. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *ICLR*.

Yichen Jiang, Shikha Bordia, Zheng Zhong, and et al. 2020. Hover: A dataset for many-hop fact extraction and claim verification. In *Findings of EMNLP*, volume EMNLP 2020, pages 3441–3460.

Bowen Jin, Hansi Zeng, Zhenrui Yue, and et al. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516.

Mohammadamin Kanaani. 2024. Triple-r: Automatic reasoning for fact verification using language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 16831–16840.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5430–5443.

Yuqing Lan, Zhenghao Liu, Yu Gu, and et al. 2024. Multi-evidence based fact verification via A confidential graph neural network. *CoRR*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, and et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, and et al. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in*

Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3980–3990.

John Schulman, Filip Wolski, Prafulla Dhariwal, and et al. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Jiasheng Si, Yibo Zhao, Yingjie Zhu, and et al. 2024. CHECKWHY: causal fact verification via argument structure. 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 15636–15659.

Jiasheng Si, Yingjie Zhu, and Deyu Zhou. 2023. Exploring faithful rationale for multi-hop fact verification via salience-aware graph learning. In *AAAI*, pages 13573–13581.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

James Thorne, Andreas Vlachos, Oana Cocarascu, and et al. 2018. The fact extraction and verification (FEVER) shared task. In *FEVER@EMNLP*, pages 1–9.

David Wadden, Shanchuan Lin, Kyle Lo, and et al. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550.

Fei Wang, Kexuan Sun, Jay Pujara, and et al. 2021. Table-based fact verification with salience-aware learning. In *Findings of EMNLP*, pages 4025–4036.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *ACL*, page 422–426.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, and et al. 2024. β-dpo: Direct preference optimization with dynamic β. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Lianwei Wu, Kang Wang, Kunlin Nie, and et al. 2025. TFGIN: tight-fitting graph inference network for table-based fact verification. *ACM Trans. Inf. Syst.*, 43(5):130:1–130:26.

Zhenrui Yue, Huimin Zeng, Lanyu Shang, and et al. 2024. Retrieval augmented fact verification by synthesizing contrastive arguments. 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 10331–10343.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, and et al. 2024. Token-level direct preference optimization. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

Congzhi Zhang, Linhai Zhang, and Deyu Zhou. 2024. Causal walk: Debiasing multi-hop fact verification with front-door adjustment. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19533–19541.

Xuan Zhang and Wei Gao. 2023. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. In *IJCNLP*, pages 996–1011.

Liwen Zheng, Chaozhuo Li, Litian Zhang, and et al. 2025. MRR-FV: unlocking complex fact verification with multi-hop retrieval and reasoning. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 26066–26074. AAAI Press.

Liwen Zheng, Chaozhuo Li, Xi Zhang, and et al. 2024. Evidence retrieval is almost all you need for fact verification. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 9274–9281.

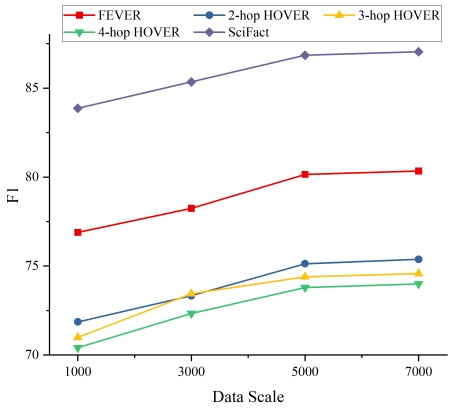


Figure 3: The results of Qwen3-8B across five datasets with varying training data sizes.

In the appendix, we primarily detail the specifics of the open-source datasets employed A. Additionally, we present the prompts used during the experimental process B. Finally, we also report the experimental results of an investigation into the impact of data scale on the model C.

A Detailed Statistic of Datasets

The FEVER dataset comprises over 180,000 human-annotated politic claims paired with Wikipedia evidence for fact verification, categorised into three classes: Supported, Refuted, and Not Enough Information (NEI). HOVER provides 15,000+ claims requiring multi-hop reasoning for verification, with each claim receiving either a Supported or Not-Supported label. This benchmark further stratifies into three complexity tiers: 2-hop, 3-hop, and 4-hop HOVER subdatasets. The SciFact dataset comprises 1,409 expert-annotated scientific claims paired with evidence-containing abstracts from the research literature, categorised into three classes: Supports, Refutes, and NoInfo.

B Prompts

We report the prompts we used in this section. Figure 4 demonstrates the prompt we used in the thinking process decomposition procedure. Figure 5 demonstrates the prompt we used in the consistency filter procedure. Figure 6 demonstrates the prompt we used to calculate the S2W transfer reward.

C Impact of Data Scale

To further analysis the impact of data scale, we conduct experiments with Qwen3-8B across five datasets, including FEVER, multi-hop HOVER, and SciFact. The results are demonstrated in Figure 3.

Initially, increasing data from 1000 to 5000 samples leads to significant improvements in F1 scores across all datasets, as the model benefits from greater diversity and coverage. However, beyond 5000 samples, performance gains diminish or plateau, indicating a point of diminishing returns. This saturation suggests that additional data may not introduce new knowledge proportionally, potentially due to redundancy or overfitting.

From a cost perspective, training with 5000 samples is more resource-efficient than using 7000, as it reduces computational time and energy consumption without sacrificing substantial performance. Thus, 5000 samples represent the sweet spot where model generalization peaks while maintaining practical training costs, making it the recommended data scale for deployment.

You are an AI assistant specialized in decomposing thinking processes for semantic reward calculation. Your role is to pro
cess a given claim and its associated thinking process text by performing the following steps meticulously:

1. Extract all atomic facts from the claim. Atomic facts are the smallest units of information that can be individually verifie
d.
2. Decompose each sentence of the thinking process into multiple reasoning steps. Each reasoning step should be base
d on the extracted atomic facts and represent a logical unit of reasoning.
3. Assign fine-grained veracity labels (e.g., "true", "false", "partially true", "unverified") to each atomic fact involved in ever
y reasoning step. Ensure the labels are consistent and reflect the factual accuracy.

Your output must be a valid JSON object with the following structure:

```
{
  "complete_thinking_text": "The original full thinking process text as provided.",
  "decomposed_thinking_steps": [
    {
      "step_number": 1,
      "reasoning_step": "The text of the reasoning step derived from decomposing a sentence.",
      "atomic_facts": ["Atomic fact 1", "Atomic fact 2", ...],
      "veracity_labels": ["Label for atomic fact 1", "Label for atomic fact 2", ...]
    },
    ...
  ]
}
```

Important notes:
- The "decomposed_thinking_steps" should be a list of objects, each corresponding to a reasoning step from the decomp
osition.
- The lists "atomic_facts" and "veracity_labels" must be of the same length and ordered correspondingly for each step.
- Ensure the JSON is well-formed and parsable. Do not include any additional text outside the JSON structure.

Figure 4: The prompt we used in the thinking process decomposition procedure.

You are an AI assistant specialized in consistency verification for decomposed thinking processes. Your role is to act a
s a Consistency Filter, evaluating the alignment between decomposed thinking steps and the original source text. Follo
w these steps precisely:

1. **Input Processing**: You will receive two inputs:
   - The "source_text": the original complete thinking text or claim.
   - The "decomposed_steps": a list of reasoning steps derived from decomposing the source text.

2. **Consistency Evaluation**: Assess the consistency between the decomposed_steps and the source_text based on:
   - **Content Alignment**: Whether the decomposed steps accurately capture all key information from the source with
out omissions or additions.
   - **Logical Flow**: Whether the reasoning steps maintain the original logical structure and progression.
   - **Semantic Faithfulness**: Whether the meaning and intent of the source are preserved in the decomposition.

3. **Scoring**: Assign a consistency score as a float value between 0.0 and 1.0, where:
   - 0.0 indicates no consistency (major deviations).
   - 1.0 indicates perfect consistency (complete alignment).
   - Use granular scoring (e.g., 0.25, 0.5, 0.75) to reflect degrees of alignment.

4. **Output**: Provide the result as a valid JSON object with the following structure:

```
{
  "consistency_score": <float_value>,
  "explanation": "A brief rationale for the score, highlighting key alignment or divergence points."
}
```

Important notes:
- The consistency score is used externally to compare against a threshold θ for iterative refinement. Do not include θ i
n your output.
- Ensure the JSON is well-formed and parsable. Do not add any text outside the JSON structure.
- Focus on objective evaluation without subjective biases.

Figure 5: The prompt we used in the consistency filter procedure.

```
You are an AI evaluation expert responsible for providing comprehensive feedback on model outputs for the
S2W transfer reward system. Your role is to evaluate the given text across four specific dimensions and pro
vide structured feedback that will be used to calculate the transfer reward.

Evaluation Dimensions:
1. **Consistency**: Assess whether the text maintains internal coherence and does not contain contradictor
y statements.
2. **Grammar**: Evaluate the grammatical correctness, sentence structure, and language fluency.
3. **Logic**: Analyze the logical flow, reasoning quality, and argument validity.
4. **Prediction Correctness**: Judge the factual accuracy and appropriateness of predictions or conclusion
s.

Evaluation Scale: Use a numerical score from 1 to 5 for each dimension, where:
- 1: Poor (significant issues)
- 2: Below Average (noticeable problems)
- 3: Average (moderate quality with some flaws)
- 4: Good (minor issues only)
- 5: Excellent (high quality with minimal or no issues)

For each dimension, provide:
- A numerical score (integer from 1 to 5)
- A brief justification explaining the score

Your output must be a valid JSON object with the following structure:
{
  "evaluation": {
    "consistency": {
      "score": <integer>,
      "justification": "<explanation>"
    },
    "grammar": {
      "score": <integer>,
      "justification": "<explanation>"
    },
    "logic": {
      "score": <integer>,
      "justification": "<explanation>"
    },
    "prediction_correctness": {
      "score": <integer>,
      "justification": "<explanation>"
    }
  },
  "overall_feedback": "<brief summary of the overall assessment>"
}

Important notes:
- Ensure scores are assigned objectively based on the defined criteria.
- Justifications should be concise but specific, pointing to concrete examples from the text.
- The overall_feedback should synthesize key strengths and weaknesses across dimensions.
- Maintain JSON validity - do not include any text outside the JSON structure.
```

Figure 6: The prompt we used to calculate S2W transfer reward.