# STRIVE: Structured Reasoning for Self-Improvement in Claim Verification

Anonymous ACL submission


## Abstract

Claim verification is the task of determining whether a claim is supported or refuted by evidence. Self-improvement methods, where reasoning chains are generated and those leading to correct results are selected for training, have succeeded in tasks like mathematical problem solving. However, in claim verification, this approach struggles. Low-quality reasoning chains may falsely match binary truth labels, introducing faulty reasoning into the self-improvement process and ultimately degrading performance. To address this, we propose STRIVE: Structured Reasoning for Self-Improved Verification. Our method introduces a structured reasoning design with Claim Decomposition, Entity Analysis, and Evidence Grounding Verification. These components improve reasoning quality, reduce errors, and provide additional supervision signals for self-improvement. STRIVE begins with a warm-up phase, where the base model is fine-tuned on a small number of annotated examples to learn the structured reasoning design. It is then applied to generate reasoning chains for all training examples, selecting only those that are correct and structurally sound for subsequent self-improvement training. We demonstrate that STRIVE achieves significant improvements over baseline models, with a 31.4% performance gain over the base model and 20.7% over Chain of Thought on the HOVER datasets, highlighting its effectiveness.


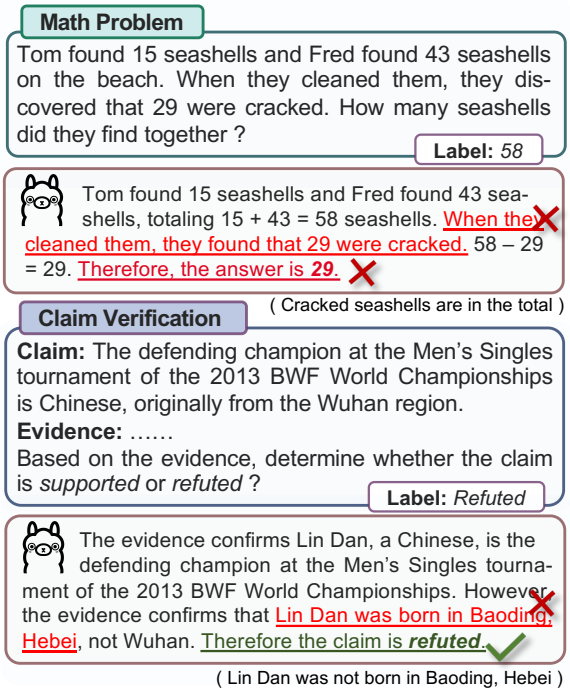


Figure 1: In math problem-solving, incorrect reasoning chains lead to mismatched final answers, while in claim verification, incorrect reasoning can still match the binary truth labels. Evidence omitted in figure.

## 1 Introduction

The proliferation of misinformation is a major challenge in today's society, eroding trust in digital information and affecting domains such as public health (Naeem and Bhatti, 2020) and politics (Mishra et al., 2022). As a result, claim verification—determining whether a claim is supported or refuted by evidence—has become crucial for ensuring reliable information.

Claim verification is a natural language inference task. While large language models (LLMs) demonstrate impressive reasoning capabilities, they struggle with claims requiring multi-hop evidence synthesis. Methods like few-shot learning and chain-of-thought (CoT) prompting have not yielded satisfactory results for complex claims.

Self-improvement methods have been successfully applied in domains like mathematical problem solving and commonsense reasoning (Zelikman et al., 2022; Hosseini et al., 2024). These methods involve generating reasoning chains and answers, selecting the ones that lead to correct results, and using them to train the model for improved performance. However, directly applying self-improvement to claim verification proves ineffective. In mathematical problem solving and commonsense reasoning, the answers are typically

numerical values or short phrases, with an almost infinite range of possibilities. This makes it difficult for incorrect reasoning chains to match the correct answer. In contrast, in claim verification, incorrect reasoning chains can easily match the binary truth labels (*Supported*/*Refuted*), leading to their selection for training, as illustrated in Figure 1. This introduces low-quality data into the training process, which undermines model performance. Our experiments validate this, showing that naive self-improvement degrades performance in claim verification task, with performance dropping by 4.5% and 1.1% on HOVER-2 and HOVER-4 (Table 1), respectively.

Upon closer examination, we found that reasoning chains in claim verification often suffer from issues such as evidence confusion, entity misidentification, and omission of key information—even when the final truthfulness decision is correct. To address this, we propose a structured reasoning design. Our design requires three key components: 1) Claim Decomposition: Breaking complex claims into smaller, manageable subclaims; 2) Entity Analysis: Linking ambiguous terms to grounded entities; and 3) Evidence Grounding Verification: Citing specific evidence snippets at each reasoning step. This structure improves the quality of reasoning chains and reduces the chance of error-prone chains being included in the training data. Moreover, the structural constraints provide additional supervision signals that, when combined with binary labels, enable more effective filtering for self-improvement.

In this work, we introduce STRIVE: Structured Reasoning for Self-Improved Verification. STRIVE begins with a warm-up phase, where a base model is fine-tuned to generate structured reasoning chains using a small set of annotated examples. This prepares the model to follow the desired reasoning format. Next, the warm-up model is used to generate reasoning chains and verification results for all examples in the training set. We select reasoning chains that both lead to correct verification results and follow the correct structural format. Finally, the correct, structured reasoning chains are used as training data to fine-tune the base model. Our experiments demonstrate that, compared to the base model, STRIVE achieves an average improvement of 31.4% and 20.7% over standard Chain of Thought (CoT) reasoning on HOVER datasets, underscoring the effectiveness of our approach.

Our contributions are summarized as follows:

- We identify the limitations of freely generated reasoning chains in claim verification for self-improvement and propose STRIVE, a solution that combines structured reasoning to address these challenges.
- We design a structured reasoning approach incorporating Claim Decomposition, Entity Analysis, and Evidence Grounding Verification, enhancing reasoning quality.
- Our experiments show that STRIVE achieves significant improvements over the baseline model across multiple datasets, demonstrating the effectiveness of our approach.

## 2 Related Works

### 2.1 Claim Verification

Early approaches to claim verification focused on fine-tuning pre-trained models, either by concatenating evidence and claims into a single input (Aly et al., 2021; Thorne et al., 2018; Hu et al., 2022) or processing evidence separately and aggregating the results (Soleimani et al., 2020; Jiang et al., 2021; Gi et al., 2021). Graph Neural Networks have also been applied to capture relationships between evidence pieces (Gong et al., 2024b; Zhao et al., 2020; Chen et al., 2022b). With the impressive generative capabilities demonstrated by large language models (LLMs), many studies have turned to LLMs for claim verification (Ma et al., 2024). FACT-GPT (Choi and Ferrara, 2024) and FactLlama (Cheung and Lam, 2023) fine-tune LLMs to directly predict the truthfulness of claims. Factscore (Min et al., 2023) employs systematic decomposition to assess the factuality of individual claim segments, while ProgramFC (Pan et al., 2023) frames claim verification as step-wise program execution. Other works, such as Li et al. (2024), Chen et al. (2022a), and Rani et al. (2023), transform the verification task into a series of sub-questions to be answered.

### 2.2 Chain of Thought Reasoning (CoT)

Chain of Thought (CoT) reasoning (Wei et al., 2022) was proposed to help LLMs solve complex problems by breaking them down into intermediate step-by-step reasoning. Kojima et al. (2022) demonstrated that adding a prompt such as "Let's think step by step" significantly boosts LLM performance. CoT reasoning has been applied to a variety of tasks, including claim verification. Studies like Hu et al. (2024) and Dougrez-Lewis et al.

(2024) evaluate CoT methods in different contexts. FOLK (Wang and Shu, 2023) leverages LLMs to transform claims into first-order logic as intermediate steps for explainable verification.

### 2.3 Self-Improvement Methods

Self-improvement methods for LLMs have garnered attention in recent years, where models are fine-tuned on their self-generated solutions, optionally iterating this process (Hosseini et al., 2024). ReST$^{EM}$ (Singh et al., 2023) generates reasoning chains for solving math problems and selects those leading to correct answers for retraining. RFT (Yuan et al., 2023) enhances reasoning chain diversity by sampling multiple chains before selection. STaR (Zelikman et al., 2022) introduces hints during reasoning generation for chains that lead to incorrect results. V-STaR (Hosseini et al., 2024) incorporates Direct Preference Optimization (Rafailov et al., 2023) into the self-improvement process. Our method shares similarities with STaR. We are the first to apply self-improvement to claim verification. We also highlight the unique challenges of claim verification, distinguishing it from tasks like math problem-solving, and address these challenges through the integration of structured reasoning design.

## 3 Method

In this section, we introduce our approach to claim verification. We first define the task itself, followed by our Structured Reasoning Design, which is specifically tailored for claim verification. Finally, we describe how this structure is applied within the self-improvement process to enhance the verification model's performance.

### 3.1 Task Formulation

The task of claim verification is to determine the truthfulness of a given claim based on a set of evidence. Our approach aims to generate reasoning chains connecting the claim and evidence to a final prediction, rather than outputting the answer directly. However, it's important to note that reasoning chains are not provided in the verification datasets, and only the final truth label is available.

Formally, given a claim $c$ and an evidence set $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$, where each $e_i$ is a descriptive piece of evidence (such as a sentence or paragraph), the goal is to obtain a model that can generate a reasoning chain $r$ that forms the intermediate reasoning steps. The final prediction

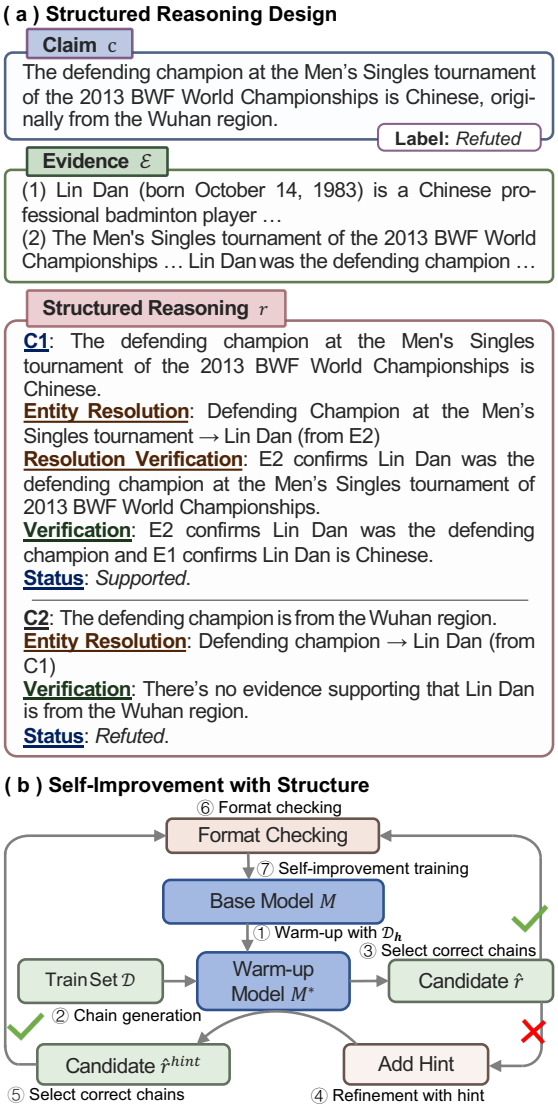


Figure 2: (a) An example demonstrating our Structured Reasoning Design. (b) Overview of STRIVE.

$\hat{p}$ is made based on the reasoning chain, where $\hat{p} \in \{Supported, Refuted\}$.

### 3.2 Structured Reasoning Design

Freely generated reasoning chains often struggle with verifying complex claims due to issues like evidence confusion, entity misidentification, and omission of key information. These problems can lead to low-quality chains being selected during self-improvement, affecting performance.

To address this, we introduce a Structured Reasoning Design that guides the verification process systematically. Our Structured Reasoning Design consists of three key components: Claim Decomposition, Entity Analysis, and Evidence Grounding Verification. An example is illustrated in Figure 2.

#### 3.2.1 Claim Decomposition

Complex claims typically describe multiple facets of a situation, such as in our example (Figure 2), where the claim covers the champion's nationality, victory, and origin. When a claim involves multiple elements, decomposing it into subclaims allows for independent verification of each component, a method known to improve performance in claim verification (Gong et al., 2024a; Min et al., 2023). In our design, we structure the reasoning chain into distinct blocks, each dedicated to a specific subclaim, labeled as `C1:`, `C2:`, and so on. This block-based structure helps mitigate the risk of overlooking key aspects during verification, which is common in free reasoning. Each subclaim block concludes with a verification result, labeled as `Status:`, to make the reasoning easy to follow and parse.

#### 3.2.2 Entity Analysis

Complex claims often involve ambiguous or unspecified entities or pronouns that may appear after Claim Decomposition. To resolve these ambiguities, we designed a two-step entity analysis process: Entity Resolution and Resolution Verification.

- **Entity Resolution** aims to identify the specific entity corresponding to an ambiguous term by leveraging evidence, labeled as `Entity Resolution:` in the reasoning chain. For instance, the term "defending champion" in the example claim (Figure 2) is resolved to the specific person "Lin Dan" using information from the evidence.

- **Resolution Verification** ensures the correctness of the Entity Resolution, labeled as `Resolution Verification:` in the reasoning chain. While Entity Resolution may correctly match an ambiguous term to a known entity in many cases, errors can occur. For example, if the claim stated a "Doubles tournament" instead of a "Singles tournament" in the example, the Entity Resolution step might still match the term "defending champion at the Men's Doubles tournament" to "Lin Dan", which is incorrect for the revised claim. Resolution Verification cross-checks the entity to ensure that the resolution is accurate, preventing errors in the reasoning process.

This two-step design ensures more precise entity analysis, minimizing the risk of errors that could impact the overall verification. We do not always require both steps to appear simultaneously. For instance, in example C2, the Entity Resolution is drawn from C1, and therefore the Resolution Verification step is omitted.

#### 3.2.3 Evidence Grounding Verification

Our structured reasoning framework incorporates explicit verification steps at multiple stages. In addition to the Resolution Verification process mentioned earlier, each subclaim undergoes a final verification step, labeled as `Verification:`. This step evaluates the subclaim as a whole, considering the resolved entities and assessing its validity against the provided evidence. It also serves as an explanation for the truthfulness decision of the subclaim.

Our structured reasoning format enforces grounding at every stage of the reasoning chain. As illustrated in Figure 2, both the resolution and verification processes explicitly cite the corresponding evidence or subclaim identifiers (e.g., C1, E2). This ensures that conclusions are drawn from verifiable sources, making the reasoning more transparent and easier for humans to interpret and verify.

### 3.3 Self-Improvement with Structure

In this section, we describe how STRIVE leverages the Structured Reasoning Design to improve model's performance in claim verification, with the general flow of this process shown in the bottom part of Figure 2. Given a training set

$$\mathcal{D} = \{(c_1, \mathcal{E}_1, p_1), \cdots, (c_N, \mathcal{E}_N, p_N)\},$$

where each $c_i$ is a claim, $\mathcal{E}_i$ is the corresponding evidence set, and $p_i$ is the final label, STRIVE follows three main steps to complete the self-improvement process: Structured Warm-up, Reasoning Chain Generation and Selection, and Self-Improvement Training.

#### 3.3.1 Structured Warm-up

Given a base model $M$, the purpose of the warm-up phase is to fine-tune $M$ into $M^*$, enabling it to generate reasoning chains conforming to our predefined structure. We use a preset prompt template $T(c, \mathcal{E})$ as follows:

> *Based on the evidence, determine if the claim is supported by the evidence or refuted by it. Output the reasoning chain.*
> *Claim:* `[claim text` $c$`]`
> *Evidence:* `(1)[evidence text` $e_1$`](2)...`

**Algorithm 1** STRIVE

**Input:** Base model $M$, training set $\mathcal{D}$, annotated set $\mathcal{D}_h$

**Output:** Improved model $M_{st}$

1: $M^* \leftarrow \text{train}(M, \mathcal{D}_h)$ // Structured Warm-up
2: $\hat{r}_i \leftarrow M^*(T(c_i, \mathcal{E}_i))$, $\hat{p}_i \leftarrow J(\hat{r}_i)\ \forall i \in [1, N]$ //Get reasoning chains and predicted labels
3: $\mathcal{D}_1 \leftarrow \{(c_i, \mathcal{E}_i, \hat{r}_i) \mid \hat{p}_i = p_i\}$ //Select correct reasoning chains based on ground truth label
4: $\hat{r}_i^{hint} \leftarrow M^*(T^{hint}(c_i, \mathcal{E}_i))$, $\hat{p}_i^{hint} \leftarrow J(\hat{r}_i^{hint})$, $\forall i \in [1, N]$, where $\hat{p}_i \neq p_i$ //Regenerate reasoning chains for incorrect predictions
5: $\mathcal{D}_2 \leftarrow \{(c_i, \mathcal{E}_i, \hat{r}_i^{hint}) \mid \hat{p}_i \neq p_i, \hat{p}_i^{hint} = p_i\}$//Select corrected reasoning chains
6: $\mathcal{D}_{st} \leftarrow \{(c_i, \mathcal{E}_i, r_i) \mid r_i \in (\mathcal{D}_1 \cup \mathcal{D}_2), f(r_i) = \text{True}\}$ //Select structurally valid chains
7: $M_{st} \leftarrow \text{train}(M, \mathcal{D}_{st} \cup \mathcal{D}_h)$ //Final Self-improvement training

Since reasoning chains are not available in the dataset, we manually annotate a very small set

$$\mathcal{D}_h = \{(c_1^h, \mathcal{E}_1^h, r_1^h, p_1^h), \cdots, (c_H^h, \mathcal{E}_H^h, r_H^h, p_H^h)\}$$

with $H$ examples, where each pair $(c_i^h, \mathcal{E}_i^h)$ is annotated with a reasoning chain $r_i^h$ that following our prescribed structure. We then fine-tune the base model $M$ on $\mathcal{D}_h$ using $T(c_i^h, \mathcal{E}_i^h)$ as input and $r_i^h$ as output, resulting in a new model $M^*$. Notably, only a very small number of examples is needed for this fine-tuning, since our structure is guided by keywords (e.g., "Verification:") that activate the model's inherent reasoning capabilities.

### 3.3.2 Reasoning Chain Generation and Selection

With the fine-tuned model $M^*$, our next objective is to leverage $M^*$ and training set $\mathcal{D}$ to generate and select high-quality structured reasoning chains for further self-improvement. Inspired by the chain generation strategy in STaR (Zelikman et al., 2022), our method involves three stages: (i) Initial Generation and Selection and (ii) Refinement with Hint and (iii) Format Checking:

**Initial Generation and Selection**: For each sample $(c_i, \mathcal{E}_i, p_i) \in \mathcal{D}$, we first generate a reasoning chain $\hat{r}_i$ using the prompt template $T(c_i, \mathcal{E}_i)$ with the model $M^*$:

$$\hat{r}_i = M^*(T(c_i, \mathcal{E}_i)) \quad \forall i \in [1, N].$$

A predicted label $\hat{p}_i$ is then derived from $\hat{r}_i$ by a rule-based function $J$:

$$\hat{p}_i = J(\hat{r}_i),$$

where if any subclaim in $\hat{r}_i$ is judged as *Refuted*, then $\hat{p}_i$ = *Refuted*, and only if all subclaims are *Supported*, then $\hat{p}_i$ = *Supported*. We select samples for which $\hat{p}_i$ matches the ground truth $p_i$ to form the set:

$$\mathcal{D}_1 = \{(c_i, \mathcal{E}_i, \hat{r}_i) \mid \hat{p}_i = p_i, \forall i \in [1, N]\}.$$

**Refinement with Hint**: For samples where $\hat{p}_i \neq p_i$, we incorporate additional guidance by modifying the prompt. If $p_i$ = *Supported*, we add the hint: "every detail in this claim is supported"; if $p_i$ = *Refuted*, we add "the claim should be refuted, locate the error in the reasoning chain". The modified prompt, inclusive of the hint, can be found in Appendix A.2. Using this modified prompt, we regenerate the reasoning chain:

$$\hat{r}_i^{hint} = M^*(T^{hint}(c_i, \mathcal{E}_i)),$$
$$\hat{p}_i^{hint} = J(\hat{r}_i^{hint}) \quad \forall i \in [1, N], \text{ where } \hat{p}_i \neq p_i.$$

Similarly, we select samples for which $\hat{p}_i^{hint}$ matches the ground truth $p_i$ to form the set:

$$\mathcal{D}_2 = \{(c_i, \mathcal{E}_i, \hat{r}_i^{hint}) \mid \hat{p}_i \neq p_i, \hat{p}_i^{hint} = p_i, \forall i \in [1, N]\}$$

**Format Checking**: At this stage, both $\mathcal{D}_1$ and $\mathcal{D}_2$ contain reasoning chains that yield correct final predictions. To further ensure the quality of these reasoning chains, we apply a rule-based structural verification function $f$, where $f(\hat{r}_i)$ = *True* or *False*, indicating whether $\hat{r}_i$ follow our predefined structure. Specifically, we enforce three criteria:

- Proper segmentation of subclaims, ensuring each reasoning step is explicitly delineated with a corresponding verification result.
- Correct evidence grounding, preventing incorrect references (e.g., citing non-existent evidence, citing C2 while verifying C1).
- Adherence to the structured format, ensuring reasoning steps are guided by predefined keywords (e.g. Entity Resolution:)

By applying this structural verification, we obtain a final set of reasoning chains $\mathcal{D}_{st}$, which not only yield correct conclusions but also adhere to the required structural format. Formally, it can be described as:

$$\mathcal{D}_{st} = \{(c_i, \mathcal{E}_i, r_i) \mid r_i \in (\mathcal{D}_1 \cup \mathcal{D}_2), f(r_i) = \textit{True}\}.$$

#### 3.3.3 Self-Improvement Training

Finally, we fine-tune the base model $M$ using both the previously selected dataset $\mathcal{D}_{st}$ and the human-annotated dataset $\mathcal{D}_h$. Note that $M^*$ is discarded after generating the reasoning chains. This allows us to obtain the final model $M_{st}$ with the enhanced reasoning ability defined by our structured approach. The entire process is summarized in Algorithm 1.

## 4 Experiment

### 4.1 Datasets

We evaluate STRIVE using two publicly available datasets, following prior work in claim verification (Gong et al., 2024a). All evaluations are performed on the validation sets, since the test sets are not publicly released. Detailed information can be found in Appendix A.3. Results are reported using the Macro-F1 score.

- **HOVER** (Jiang et al., 2020) This dataset comprises claims necessitating multi-step reasoning across multiple pieces of evidence and is categorized into three subsets: HOVER-2 for two-hop reasoning, HOVER-3 for three-hop reasoning, and HOVER-4 for four-hop reasoning.
- **FEVEROUS-S** (Aly et al., 2021) Derived from the FEVEROUS dataset, this subset contains claims that rely solely on unstructured textual evidence. Compared to HOVER, claims in FEVEROUS-S generally exhibit lower complexity in reasoning.

### 4.2 Baselines

To assess the effectiveness of STRIVE, we compare it against a variety of baselines, including pretrained/fine-tuned models, prompt-based approaches using the base model, and base model fine-tuned on the training data.

For pretrained/fine-tuned models, we include the following: (i) BERT-FC (Soleimani et al., 2020): Pretrained BERT model (Devlin et al., 2019) tailored for fact-checking tasks. (ii) LisT5 (Jiang et al., 2021): Pretrained T5 model (Raffel et al., 2020) specialized for fact-checking tasks. (iii) RoBERTa-NLI (Nie et al., 2020): Pretrained RoBERTa-large model (Liu et al., 2019) fine-tuned on four natural language inference datasets. (iv) MULTIVERS (Wadden et al., 2022): A Long-Former model (Beltagy et al., 2020) fine-tuned on the FEVER (Thorne et al., 2018) dataset.

For prompt-based approaches, we evaluate the following: (i) Zero-shot: The model predicts the final label directly, without requiring reasoning steps. (ii) Zero-shot + CoT: The model outputs reasoning chains before predicting the final label. (iii) Few-shot: Similar to zero-shot, but with the inclusion of labeled examples. (iv) Few-shot + Structured CoT: Similar to zero-shot + CoT, but with examples of structured reasoning chains from $\mathcal{D}_h$. All baseline prompts are similar to $T(c, \mathcal{E})$ for fairness, detailed in Appendix A.1.

Finally, for base models that undergo fine-tuning, we compare with: (i) Lora Fine-tuning: Fine-tuning the base model on the training set $\mathcal{D}$ using only binary labels. (ii) STaR* (Zelikman et al., 2022): We re-implemented this self-improvement method for claim verification. While similar to our approach (Algorithm 1), it differs in that reasoning chains are generated freely by the model, without structural constraints or a warm-up phase.

### 4.3 Implementation Details

We use Llama-3-8B-Instruct as the base model, as it is a widely used open-source language model. For all fine-tuning tasks, including those in baseline models and STRIVE, we employ the GPU memory-efficient LoRA fine-tuning method (Hu et al., 2021), allowing our experiments to fit on a single consumer-grade GPU (e.g., NVIDIA 4090).

In the Structured Warm-up process, we use a small human-annotated dataset $\mathcal{D}_h$ containing only $H = 10$ examples, with 8 labeled as *Refuted* and 2 as *Supported*. We prioritize teaching the model to identify errors rather than admitting correct claims. The intermediate model $M^*$ is obtained by fine-tuning on $\mathcal{D}_h$ for 10 epochs using LoRA. Despite the large number of epochs, only 0.1% of the model's parameters are updated, ensuring the model's overall performance is maintained while enforcing the prescribed structure. The training set $\mathcal{D}$ consists of $N = 600$ examples, with reasoning chains generated at a temperature setting of 0.01. The final model $M_{st}$ is obtained by fine-tuning for 2 epochs on the union of the collected set and the human-annotated set, $\mathcal{D}_{st} \cup \mathcal{D}_h$.

## 5 Results and Discussion

### 5.1 Overall Performance

We present the overall results of STRIVE and the baseline models in Table 1. Among all methods, STRIVE achieves the best performance, underscor-

| Model Family | Approach | HOVER-2 | HOVER-3 | HOVER-4 | FEVEROUS-S |
|---|---|---|---|---|---|
| PT/FT Models | BERT-FC | 53.40 | 50.90 | 50.86 | 74.71 |
| | LisT5 | 56.15 | 53.76 | 51.67 | 77.88 |
| | RoBERTa-NLI | 74.62 | 62.23 | 57.98 | 88.28 |
| | MULTIVERS | 68.86 | 59.87 | 55.67 | 86.03 |
| Llama-3-8B | Zero-shot | 55.10 | 55.09 | 53.51 | 78.21 |
| | Zero-shot + CoT | 63.76 | 57.13 | 57.47 | 84.94 |
| | Few-shot | 55.33 | 55.63 | 52.86 | 79.17 |
| | Few-shot + Structured CoT | 69.71 | 66.71 | 59.63 | 85.67 |
| LoRA-Llama-3 | LoRA Fine-tuning | 64.21 | 60.35 | 60.34 | 91.52 |
| | STaR* (Self-Improvement) | 60.90 | 58.61 | 56.86 | 87.45 |
| | **STRIVE (Ours)** | **76.13** ±0.84 | **70.50** ±0.55 | **68.50** ±1.27 | **91.91** ±0.44 |

Table 1: Macro-F1 scores for claim verification models on HOVER and FEVEROUS-S datasets. "PT/FT" refers to pretrained/fine-tuned models.

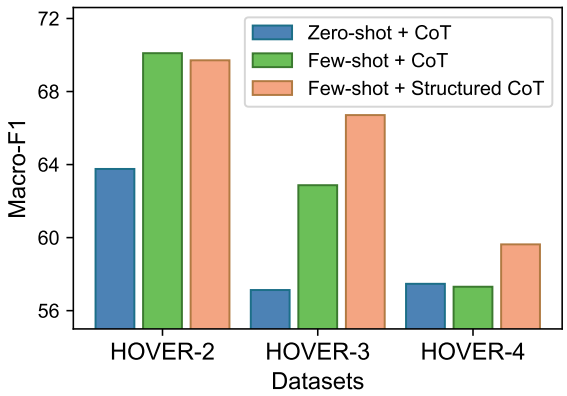


Figure 3: Comparison of Chain-of-Thought approaches with and without structured reasoning over HOVER datasets.

ing the effectiveness of our approach. The results also highlight the importance of explicitly reasoning steps in claim verification. Pretrained/fine-tuned models, zero-shot and few-shot methods that do not incorporate reasoning steps, generally perform worse than methods that explicitly utilize reasoning steps (e.g., Zero-shot + CoT, Few-shot + Structured CoT).

We present reasoning chains generated by STRIVE and **error analysis** in Appendix A.4.

### 5.2 Effectiveness of Structured Reasoning

In this section, we explain how Structured Reasoning contributes to performance improvement in claim verification.

**Structured Design Improves Reasoning Quality.** We compare three approaches: Zero-shot + CoT, Few-shot + CoT, and Few-shot + Structured CoT. Zero-shot + CoT and Few-shot + CoT use chain-of-thought prompting without structural constraints. In Few-shot + CoT, reasoning examples are rewritten by removing structural elements from $\mathcal{D}_h$'s chains. In contrast, Few-shot + Structured CoT directly uses reasoning examples from $\mathcal{D}_h$ with the desired structure. As shown in Figure 3, while Few-shot + CoT performs slightly better on simpler tasks like HOVER-2, structured reasoning shows significant advantages in more complex scenarios like HOVER-3 and HOVER-4. This confirms that structured reasoning improves reasoning quality, particularly for complex tasks.

**Structured Design Helps in Self-Improvement Training.** Figure 4 compares self-improvement results with (top) and without (bottom) structured reasoning. In the top part, performance before self-improvement corresponds to the model $M^*$ after the Structured Warm-up phase. In the bottom part, "before" and "after" refer to Zero-shot + CoT and STaR*, respectively. The figure shows that structured reasoning leads to steady performance improvement after the warm-up model, with gains increasing as task complexity rises (1.6% for 2-hop, 4.4% for 4-hop). In contrast, without the structured design, improvements are inconsistent. As shown in the bottom of Figure 4, performance even drops on HOVER-2 and HOVER-4 due to the inclusion of erroneous reasoning chains that incorrectly match the final label. These results highlight the importance of structured reasoning in ensuring successful self-improvement for claim verification.

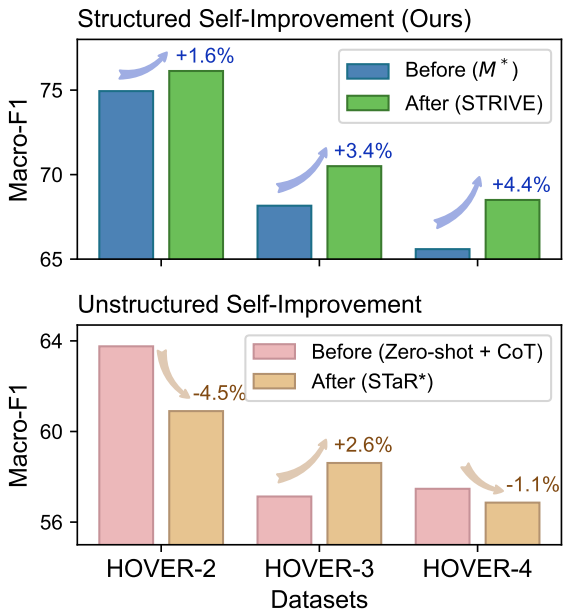


Figure 4: Comparison of self-improvement training results with (top) and without (bottom) structured reasoning over HOVER datasets.

| Models | HOVER-2 | HOVER-3 | HOVER-4 |
|---|---|---|---|
| STRIVE | **76.13** | 70.50 | **68.50** |
| STRIVE$_{\text{w/o CD}}$ | 73.69 | 67.36 | 64.03 |
| STRIVE$_{\text{w/o EA}}$ | 75.96 | 67.36 | 64.77 |
| STRIVE$_{\text{w/o EG}}$ | 75.08 | 69.42 | 68.22 |
| STRIVE$_{\text{w/o FC}}$ | 76.11 | 69.69 | 66.89 |
| STRIVE$_{\text{w/o hint}}$ | 75.90 | 70.34 | 67.20 |
| STRIVE$_{\text{2 rounds}}$ | 75.98 | **70.55** | 68.43 |

Table 2: Ablation study results. Macro-F1 scores for different model variants across the HOVER datasets. "CD" stands for Claim Decomposition, "EA" stands for Entity Analysis, "EG" stands for Evidence Grounding Verification, and "FC" stands for Format Checking.

### 5.3 Analysis of Structure Design

In this section, we analyze the structural reasoning design of STRIVE. As outlined in Section 3.2, the core components—Claim Decomposition, Entity Analysis, and Evidence Grounding Verification—are introduced to address the issue of low-quality reasoning chains generated by the model. To assess the contribution of each component, we conduct an ablation study by removing each design element individually, resulting in three variants: STRIVE$_{\text{w/o CD}}$, STRIVE$_{\text{w/o EA}}$, and STRIVE$_{\text{w/o EG}}$. The results are presented in Table 2.

From the table, it is evident that Claim Decomposition is the most impactful design; its removal causes performance to drop across all HOVER datasets. Entity Analysis also plays a significant role, particularly in more complex scenarios (HOVER-4), as complex claims often involve multiple ambiguous entities. In contrast, the Evidence Grounding Verification design has a smaller impact on performance but still contributes valuable benefits, primarily by improving human readability of the reasoning chains.

### 5.4 Analysis of Self-Improvement Training

To analyze the self-improvement training process, we propose three variants: STRIVE$_{\text{w/o FC}}$, STRIVE$_{\text{w/o hint}}$ and STRIVE$_{\text{2 rounds}}$, with performance results shown in Table 2.

(i) STRIVE$_{\text{w/o FC}}$ removes the format checking step during the Reasoning Chain Selection process, causing a performance drop, particularly on the HOVER-4 dataset. (ii) STRIVE$_{\text{w/o hint}}$ eliminates the hint-based refinement in the Reasoning Chain Generation process (steps 4 and 5 in Algorithm 1), leading to a small performance decrease across the HOVER datasets. (iii) STRIVE$_{\text{2 rounds}}$ adds an extra round of self-improvement training (repeating steps 2-7 in Algorithm 1 with $M^*$ replaced by $M_{st}$). However, this results in performance similar to the original model, indicating no significant benefit from the additional round. Therefore, we opt to use a single round of training.

Interestingly, unlike math problem solving (Zelikman et al., 2022; Hosseini et al., 2024), where extra rounds are critical, STRIVE performs well with a single round. This is likely due to its use of structured guidance, which activates the model's inherent reasoning abilities, and no new capabilities are gained from further training rounds.

## 6 Conclusion

We presented STRIVE: Structured Reasoning for Self-Improved Verification, a method that integrates structured reasoning and self-improvement training for claim verification. By incorporating Claim Decomposition, Entity Analysis, and Evidence Grounding Verification, STRIVE improves the quality of reasoning chains and enables more effective self-improvement. Our experiments demonstrate that STRIVE significantly outperforms baseline approaches, highlighting the effectiveness of structured reasoning for self-improvement in claim verification.

## 7 Limitations

Our approach relies on a structured warm-up phase that requires a small amount of annotated data. In our experiments, we selected 10 moderately difficult claims for reasoning chain labeling without further extensive sample filtering. While this approach has yielded positive results, we recognize that the choice of these samples may influence the subsequent model training. We believe that more carefully selected or diverse samples could further enhance the model's performance and provide additional insights into how sample selection impacts self-improvement. Additionally, our approach improves the claim verification capabilities of LLM in a resource-efficient manner. Both the quantity of annotations and the training strategies were designed for resource efficiency. This low-cost approach has proven effective for performance enhancement, but it also presents opportunities for future research, particularly in terms of scalability. Expanding to larger datasets and more complex models could offer valuable insights, though it remains to be explored in future works.

## References


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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv preprint*, abs/2004.05150.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022a. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhendong Chen, Siu Cheung Hui, Fuzhen Zhuang, Lejian Liao, Fei Li, Meihuizi Jia, and Jiaqi Li. 2022b. Evidencenet: Evidence fusion network for fact verification. In *WWW*, pages 2636–2645. ACM.

Tsun-Hin Cheung and Kin-Man Lam. 2023. Factllama: Optimizing instruction-following language models with external knowledge for automated fact-checking. In *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 846–853. IEEE.

Eun Cheol Choi and Emilio Ferrara. 2024. Fact-gpt: Fact-checking augmentation via claim matching with llms. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 883–886, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

John Dougrez-Lewis, Mahmud Elahi Akhter, Yulan He, and Maria Liakata. 2024. Assessing the reasoning abilities of chatgpt in the context of claim verification. *arXiv preprint arXiv:2402.10735*.

In-Zu Gi, Ting-Yu Fang, and Richard Tzong-Han Tsai. 2021. Verdict inference with claim and retrieved elements using RoBERTa. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 60–65, Dominican Republic. Association for Computational Linguistics.

Haisong Gong, Huanhuan Ma, Qiang Liu, Shu Wu, and Liang Wang. 2024a. Navigating the noisy crowd: Finding key information for claim verification. *arXiv preprint arXiv:2407.12425*.

Haisong Gong, Weizhi Xu, Shu Wu, Qiang Liu, and Liang Wang. 2024b. Heterogeneous graph reasoning for fact checking over texts and tables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 100–108.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Nan Hu, Zirui Wu, Yuxuan Lai, Xiao Liu, and Yansong Feng. 2022. Dual-channel evidence fusion for fact verification over texts and tables. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5232–5242.

Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S. Yu, and Zhijiang Guo. 2024. Towards understanding factual knowledge of large language models. In *The Twelfth International Conference on Learning Representations*.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with t5 for fact

verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410, Online. Association for Computational Linguistics.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Miaoran Li, Baolin Peng, Michel Galley, Jianfeng Gao, and Zhu Zhang. 2024. Self-checker: Plug-and-play modules for fact-checking with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 163–181, Mexico City, Mexico. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Huanhuan Ma, Weizhi Xu, Yifan Wei, Liuji Chen, Liang Wang, Qiang Liu, Shu Wu, and Liang Wang. 2024. EX-FEVER: A dataset for multi-hop explainable fact verification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9340–9353, Bangkok, Thailand. Association for Computational Linguistics.

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.

Shreyash Mishra, S Suryavardan, Amrit Bhaskar, Parul Chopra, Aishwarya N Reganti, Parth Patwa, Amitava Das, Tanmoy Chakraborty, Amit P Sheth, Asif Ekbal, et al. 2022. Factify: A multi-modal fact verification dataset. In *DE-FACTIFY@ AAAI*.

Salman Bin Naeem and Rubina Bhatti. 2020. The covid-19 'infodemic': a new front for information professionals. *Health Information and Libraries Journal*, 37:233 – 239.

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.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Anku Rani, S.M Towhidul Islam Tonmoy, Dwip Dalal, Shreya Gautam, Megha Chakraborty, Aman Chadha, Amit Sheth, and Amitava Das. 2023. FACTIFY-5WQA: 5W aspect-based fact verification through question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10421–10440, Toronto, Canada. Association for Computational Linguistics.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. BERT for evidence retrieval and claim verification. In *ECIR (2)*, volume 12036 of *Lecture Notes in Computer Science*, pages 359–366. Springer.

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, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022. Multivers: Improving scientific claim verification with

weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76.

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. *arXiv preprint arXiv:2310.05253*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul N. Bennett, and Saurabh Tiwary. 2020. Transformer-xh: Multi-evidence reasoning with extra hop attention. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

# A Appendix

## A.1 Prompts for Baseline Models

We incorporate various prompt-based methods in our experiments to ensure a fair comparison. To maintain consistency, we keep most of the prompt content similar to that used for STRIVE in Section 3.3.1. Below, we list the prompts used for the baseline models.

For the zero-shot and LoRA fine-tuning experiments, we use the following prompt:

> *Based on the evidence, determine if the claim is supported by the evidence or refuted by it.*
> *Claim:* `[claim text` $c$`]`
> *Evidence:* `(1)[evidence text` $e_1$`](2)...`
> *Please respond with only whether the claim is "Supported" or "Refuted."*

For the zero-shot + CoT and STaR* experiments, the prompt is as follows:

> *Based on the evidence, determine if the claim is supported by the evidence or refuted by it.*
> *Claim:* `[claim text` $c$`]`
> *Evidence:* `(1)[evidence text` $e_1$`](2)...`
> *Think step by step, output your response in the following format:*
> *Chain:* `[your reasoning chain]`
> *Answer:*`[the claim is supported or the claim is refuted]`

For the Few-shot experiment, the prompt is similar to the zero-shot prompt but includes examples:

> *Based on the evidence, determine if the claim is supported by the evidence or refuted by it. Please respond with only whether the claim is "Supported" or "Refuted." Here are some examples:*
> *Claim: Simon Grundel-Helmfelt is most ...*
> *Evidence: (1) Baron Simon Grundel ... (2)...*
> *Output: Refuted*
>
> *(...more examples...)*
>
> *Follow the above examples:*
> *Claim:* `[claim text` $c$`]`
> *Evidence:* `(1)[evidence text` $e_1$`](2)...`
> *Output:*

For the few-shot + structured CoT, we use the same prompt as STRIVE, but with examples of structured reasoning:

> *Based on the evidence, determine if the claim is supported by the evidence or refuted by it. Output the reasoning chain. Here are some examples:*
> *Claim: Simon Grundel-Helmfelt is most ...*
> *Evidence: (1) Baron Simon Grundel ... (2)...*
> *Chain: C1: Simon Grundel ...*
>
> *(...more examples...)*
>
> *Follow the above examples:*
> *Claim:* `[claim text` $c$`]`
> *Evidence:* `(1)[evidence text` $e_1$`](2)...`
> *Chain:*

## A.2 Prompts with Hint

In STRIVE, we add hint and regenerate reasoning chains for the ones that falsely predict the label of the claim. If the truth label is $p =$ *Supported*, we

use prompt:

> *Based on the evidence, determine if the claim is supported by the evidence or refuted by it. Output the reasoning chain.*
> *Claim:* `[claim text` $c$`]`
> *Evidence:* `(1)[evidence text` $e_1$`](2)...`
> *Hint: Every detail in this claim is supported.*

If the truth label is $p =$ *Refuted*, we use prompt:

> *Based on the evidence, determine if the claim is supported by the evidence or refuted by it. Output the reasoning chain.*
> *Claim:* `[claim text` $c$`]`
> *Evidence:* `(1)[evidence text` $e_1$`](2)...`
> *Hint: The claim should be refuted, locate the error in the reasoning chain.*

### A.3 Dataset Statistics

The following table presents the information of the dataset (validation set) that we have tested on. "HV" represents HOVER, while "FS" stands for FEVEROUS.

| Dataset | HV-2 | HV-3 | HV-4 | FS-S |
|---|---|---|---|---|
| Total | 1,126 | 1,835 | 1,039 | 2,962 |
| Supported Claims | 521 | 968 | 511 | 1411 |
| Refuted Claims | 605 | 867 | 528 | 1551 |
| Avg. Words in Claim | 19.6 | 24.1 | 32.2 | 27.6 |
| Avg. Evidence Pieces | 2.0 | 3.0 | 4.0 | 2.1 |
| Avg. Words in Evidence | 137.3 | 211.1 | 278.2 | 50.3 |

### A.4 STRIVE-Generated Reasoning Chains

In this section, we present several reasoning chains generated by STRIVE. The evidence is omitted due to its length, which makes it unsuitable for display. Red coloring is used to highlight where the reasoning chains become incorrect.

**Claim**: The artist whose work was displayed in 1974 at Debel Gallery, was closely associated with the Viennese Actionism group, while Howard Zieff was an American television commercial director.
**Label**: Supported

**Model Output**:
C1: The artist whose work was displayed in 1974 at Debel Gallery was closely associated with the Viennese Actionism group.
Entity Resolution: Artist -> Rudolf Schwarzkogler (from E2)
Resolution Verification: E1 confirms Rudolf Schwarzkogler's work was displayed in 1974 at Debel Gallery.
Verification: E2 supports that Rudolf Schwarzkogler was closely associated with the Viennese Actionism group.
Status: Supported.

C2: Howard Zieff was an American television commercial director.
Entity Resolution: Howard Zieff -> Howard B. Zieff (from E3)
Resolution Verification: E3 confirms Howard B. Zieff was an American director, television commercial director, and advertising photographer.
Verification: E3 supports that Howard Zieff was an American television commercial director.
Status: Supported.

**Claim**: Ben Karlin wrote the 2013 episode of the TV show, Netflix, that was directed by the actor who played Kevin Arnold in "The Wonder Years"
**Label**: Refuted

**Model Output**:
C1: Ben Karlin wrote the 2013 episode of the TV show, Netflix.
Entity Resolution: TV show -> Modern Family (from E1)
Resolution Verification: E1 confirms 'ClosetCon '13' aired in 2013, and E1 also confirms Ben Karlin wrote it.
Verification: E1 supports that Ben Karlin wrote the 2013 episode, but it's clear that the TV show is Modern Family, not Netflix.
Status: Refuted.

C2: The actor who played Kevin Arnold in T̈he Wonder Yearsd̈irected the 2013 episode.
Entity Resolution: Actor -> Fred Savage (from E2)
Resolution Verification: E2 confirms Fred Savage played Kevin Arnold in "The Wonder Years".
Verification: E2 supports that Fred Savage played Kevin Arnold, but it's clear that he directed the episode in 2013, not that he played Kevin Arnold in 2013.
Status: Refuted.

C3: The TV show was Netflix.
Entity Resolution: TV show -> Modern Family (from E1)
Resolution Verification: E1 confirms 'ClosetCon '13' aired on Modern Family, not Netflix.
Verification: E1 refutes that the TV show was Netflix.
Status: Refuted.

**Claim**: The builder of Shadow Creek Golf Course is an Jewish real estate developer. The builder also owns the Encore hotel and casino in Las Vegas.
**Label**: Refuted

**Model Output**:
C1: The builder of Shadow Creek Golf Course is an Jewish real estate developer.
Entity Resolution: Builder of Shadow Creek Golf Course -> Steve Wynn (from E3)
Resolution Verification: E3 confirms Steve Wynn built Shadow Creek Golf Course.
Verification: E3 shows Steve Wynn built Shadow Creek Golf Course, but it doesn't provide any information about his religion. There's no evidence to support the claim that Steve Wynn is Jewish.
Status: Refuted.

C2: The builder also owns the Encore hotel and casino in Las Vegas.
Entity Resolution: Builder -> Steve Wynn (from E3)
Resolution Verification: E3 confirms Steve Wynn built Shadow Creek Golf Course, and E2 confirms Wynn Resorts owns Encore.
Verification: E3 shows Steve Wynn built Shadow Creek Golf Course, and E2 confirms Wynn Resorts owns Encore.
Status: Supported.

**Claim**: Carnegie Mellon University, not the institution which is home to the Lyme Academy of Fine Arts, is a university in Pennsylvania.
**Label**: Supported

**Model Output**:
C1: Carnegie Mellon University, not the institution which is home to the Lyme Academy of Fine Arts, is a university in Pennsylvania.
Entity Resolution: institution which is home to the Lyme Academy of Fine Arts -> University of New Haven (from E1)
Resolution Verification: E1 confirms Lyme Academy College of Fine Arts is part of University of New Haven, not CMU. E3 confirms CMU is in Pennsylvania.
Verification: E1 refutes the institution which is home to the Lyme Academy of Fine Arts being CMU. E3 supports CMU being in Pennsylvania.
Status: Refuted.

(The model misunderstood the meaning of the claim.)

**Claim**: Orange is the main flavor of both zero-calorie soft drinks, Solo from Norway, and the drink advertised with Krupa (song).
**Label**: Refuted

**Model Output**:
C1: Orange is the main flavor of Solo from Norway.
Verification: E1 confirms Solo is an orange-flavored soft drink from Norway.
Status: Supported.

C2: Orange is the main flavor of the drink advertised with Krupa (song).
Entity Resolution: Drink advertised with Krupa -> Sunkist
Resolution Verification: E3 confirms Krupa is from an Sunkist advert.
Verification: E3 confirms Krupa is from an Sunkist advert and E2 confirms Sunkist is primarily orange flavored.
Status: Supported.

(The model lost the information of "zero-calorie" in the claim during the Claim Decomposition process, leading to an error.)

**Claim**: The star of Forever Strong played a character that is based on the Hanna-Barbera show "Birdman and the Galaxy Trio".
**Label**: Supported

**Model Output**:
C1: The star of Forever Strong played a character that is based on the Hanna-Barbara show "Birdman and the Galaxy Trio".
Entity Resolution: Star of Forever Strong -> Gary Cole (from E1)
Resolution Verification: E1 confirms Gary Cole was a star of Forever Strong.
Entity Resolution: Character played by Gary Cole -> ? (from E2)
Resolution Verification: E2 confirms Gary Cole voiced Harvey Birdman, but it doesn't mention him playing a character in Forever Strong.
Verification: E1 shows Gary Cole was a star of Forever Strong, but E2 shows he voiced Harvey Birdman, not a character in Forever Strong. Therefore, the claim is refuted.
Status: Refuted.

(The model incorrectly associated the character in the claim with "Forever Strong," leading to an error in Entity Resolution.)