UNIVERSITY OF CALIFORNIA
SANTA CRUZ

EFFICIENT LARGE LANGUAGE MODEL REASONING FOR FACT VERIFICATION AND MATHEMATICAL REASONING

A dissertation submitted in partial satisfaction
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Rongwen Zhao

December 2025

The Dissertation of Rongwen Zhao
is approved:

_____
Professor Jeffrey Flanigan, Chair

_____
Professor Yi Zhang

_____
Professor Pranav Anand

_____
Peter Biehl
Vice Provost and Dean of Graduate Studies

Contents

Figures and Tables

List of Figures

List of Tables

Abstract

Efficient Large Language Model Reasoning for Fact Verification and Mathematical Reasoning

by

Rongwen Zhao

Large language models (LLMs) demonstrate strong performance on complex reasoning, yet their reliability and efficiency in real-world deployments remain limited, particularly in specialized domains with sparse coverage, in scalable post-training for reasoning, and in the underexplored dynamics of distilling long chain-of-thought (CoT) behaviors. This dissertation investigates these challenges through the lens of reasoning-intensive tasks, focusing on **claim verification** and **mathematical reasoning**.

First, we introduce SynthVerify, a step-by-step synthetic data generation framework that integrates structured domain specification, evidence synthesis, and aspect-conditioned claim creation. Without relying on external corpora or costly human annotation, SynthVerify yields diverse training data and substantially improves zero-shot transfer claim verification across multiple specialized domains. Second, we propose a three-stage training recipe that develops smaller reasoning models capable of accurate, context-grounded claim verification trained via reinforcement learning with verifiable rewards. Empirically, the trained model R1-CV consistently outperforms the out-of-box reasoning LLMs baselines of comparable scale while preserving efficiency.

Third, we present a systematic study of long CoT reasoning distillation for large reasoning models in mathematical reasoning. Our cross-model, cross-architecture analysis

reveals: student scale strongly governs successful reasoning transfer with architecture-specific effects; mid-sized teachers often generalize more effectively across domains; and reasoning quality in training traces outweighs quantity. Together, these contributions provide practical methods and empirical principles for building domain-robust, efficient reasoning LLMs and offer actionable guidance for data creation, model selection, and training strategies beyond the two focal tasks.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Jeffrey Flanigan, for his unwavering guidance, encouragement, and support throughout my PhD journey. His mentorship has shaped not only the trajectory of my research but also my development as an independent researcher. I have learned immensely from his insight, clarity of thought, and high standards of scholarship.

I am grateful to the members of my dissertation committee, Professor Yi Zhang and Professor Pranav Anand, for their valuable feedback, thought-provoking questions, and generous time, especially during the final stages of this dissertation. Their input helped sharpen the scope and clarity of my work.

I would also like to thank my labmates in the JLab group. Your discussions, feedback, and camaraderie enriched my experience and broadened my perspective. I am especially thankful to those who read early drafts of my work, debugged experiments with me, or offered moral support through paper deadlines and conference seasons.

I would like to thank my friends for being there through the highs and lows of graduate school, celebrating every small win, and keeping me grounded during tough times. Special thanks to those who reminded me to take breaks, share meals, and stay human.

Finally, I thank my family for their unconditional love, patience, and support. Your belief in me, even when I doubted myself, has carried me through this long and meaningful journey. This dissertation is dedicated to you.

Chapter 1

Introduction

Large language models (LLMs) (Jaech et al. 2024; OpenAI, 2025; Guo et al. 2025a; Team et al. 2025; Team, 2025) have transformed natural language processing (NLP) by enabling impressive performance across a wide range of text understanding and complex reasoning tasks (Hendrycks et al., 2021a; Jimenez et al., 2023; Rein et al., 2024; Jain et al., 2024; Chen et al. 2025; Yao et al., 2024). LLM emergent capabilities, such as multi-step reasoning (Paranjape et al., 2023; Aksitov et al. 2023; Patel et al., 2024), factual reasoning (Yu et al., 2024; Es et al., 2024; Asai et al., 2024), and agentic tool use (Lu et al., 2025; Singh et al., 2025; Wu et al., 2025), hold promise for applications in scientific discovery (Shojaee et al., 2024; Shojaee et al., 2025; Ma et al., 2024), decision-making in complex environment (Hao et al., 2023; Wang et al., 2024; Cheng et al. 2024), and trustworthy AI (Hua et al., 2024; 2024; Huang et al. 2024).

However, despite rapid progress, reasoning with LLMs is far from solved. LLMs face fundamental challenges that hinder their reliability and efficiency in real-world deployments. First, LLMs often struggle to reason accurately in specialized or knowledge-intensive domains due to gaps in their training data (Reizinger et al., 2024; Zhang et al.,

2024; Wang et al., 2024). Second, current alignment* and fine-tuning approaches are computationally expensive and difficult to scale across diverse and complex reasoning tasks (Sui et al. 2025; Kumar et al., 2025; Cuadron et al. 2025). Lastly, there is a lack of systematic evaluation of long chain-of-thought distillation for reasoning LLMs across model scales, architectures, and downstream tasks (Feng et al., 2024; Chenglin et al., 2024a; Tian et al., 2025).

This thesis investigates these challenges in the context of *reasoning-intensive tasks*, with a particular focus on the data, reasoning strategies, and model transferability. Although the proposed methods are broadly applicable to a variety of reasoning settings, we focus on two tasks, **claim verification**, and **mathematical reasoning**. Claim verification provides a representative testbed for reasoning research: It requires interpreting diverse evidence, applying domain knowledge, and making logically sound veracity judgments, making it an ideal benchmark for evaluating general reasoning improvements. Our second task, mathematical reasoning, requires understanding complex relationships and uncertainties in language data. General mathematical reasoning ability allows LLMs to generalize and perform a wide range of tasks. Both tasks play an important role in real-world scenarios.

We address the aforementioned challenges through three complementary research directions. First, we explore how a proposed step-by-step **synthetic data generation** framework can improve model generalization for the zero-shot claim verification task without relying on expensive human annotation or domain-specific corpora. Second, we develop an **efficient reasoning training framework** to develop smaller reasoning models efficiently while maintaining high accuracy in complex, context-grounded claim

*Instruction-tuning

verification tasks. Third, we conduct a **systematic study of long chain-of-thought (CoT) reasoning distillation** for large reasoning model to uncover how teacher–student model relationships affect the transfer of mathematical reasoning.

The rest of this chapter is an overview of the thesis document, followed by the thesis statement and contributions.

§ 1.1 Synthetic Data Generation for Zero-Shot Claim Verification

Claim verification, the task of determining whether a claim is supported by given evidence, plays a critical role in combating misinformation (Litou et al., 2017; Hassan et al., 2017; Shu et al., 2017). However, despite the success of LLMs across many NLP tasks, they face key challenges in this setting (Zhang and Gao, 2023a; Guan et al., 2024; Augenstein et al. 2024; Quelle and Bovet, 2024). A major issue is the lack of domain-specific knowledge in pretraining corpora, such as financial or medical domains, which are often underrepresented compared to general sources like Wikipedia or newswire (Vladika and Matthes, 2024). This leads to performance degradation on specialized claims, affecting both evidence identification and reasoning. Although fine-tuning on existing datasets (e.g., FEVER (Thorne et al., 2018), VitaminC (Schuster et al., 2021)) can partially address this, these datasets are often domain-limited or costly to scale across diverse domains (Zhu et al., 2022; Nan et al., 2022; Gu et al., 2023). Synthetic data generation has emerged as a potential solution (Pan et al., 2021; Wright et al., 2022; Bussotti et al., 2024), but existing methods either lack scalability (Pan et al., 2021; Wright et al., 2022) or depend heavily on external resources (Bussotti et al.,

2024), limiting generalization and practicality.

To overcome these limitations, in Chapter 3, we propose SYNTHVERIFY, a step-by-step prompting-based synthetic data generation framework. Our approach integrates structured domain specification, evidence synthesis, and aspect-based claim generation to ensure both verifiability and diversity without relying on external corpora. Through extensive experiments, we show that models trained on our synthetic data achieve stronger zero-shot transfer, better reasoning on complex claims, and more robust performance across multiple specialized domains.

§ 1.2 Efficient Reasoning for Context-Grounded Claim Verification

Current LLM-based approaches (Min et al., 2023; Chern et al. 2023; Zhao et al. 2023; Kamoi et al., 2023a; Wei et al. 2024; Song et al., 2024; Wang et al., 2024) for claim verification often follow the Decompose-Then-Verify paradigm, where a complex claim is broken into sub-claims using the Chain-of-Thought (CoT) (Wei et al., 2022), and each sub-claim is then verified using provided or retrieved evidence. While this reduces task complexity, decomposition errors (Min et al., 2023; Chern et al. 2023; Zhao et al. 2023) and the reliance on expert-designed prompts (Min et al., 2023; Wang et al., 2024) limit performance. Recent reasoning LLMs like OpenAI o1 (Jaech et al. 2024) and DeepSeek R1 (Guo et al. 2025a) demonstrate that reinforcement learning (RL) can enable self-reflective reasoning and iterative refinement (OpenAI, 2025; Team et al. 2025; Team, 2025), with models such as DeepSeek-R1 (Guo et al. 2025a) achieving state-of-the-art results in reasoning-heavy tasks through purely reinforcement learning-driven training.

Motivated by these advances, in Chapter 4, we propose **R1-ClaimVerifier (R1-CV)**, a three-stage framework for efficient, reasoning-driven claim verification using RL. First, we construct a high-quality reasoning dataset via data distillation. Second, we fine-tune a base LLM through CoT-SFT training to establish a strong reasoning foundation. Finally, we apply reinforcement learning-based refinement to enhance reasoning quality. To the best of our knowledge, we are the first to develop efficient small reasoning LLMs for the context-grounded claim verification tasks using RL with verifiable rewards (Guo et al. 2025a). Extensive experiments across multiple context-grounded claim verification tasks show that R1-CV consistently outperforms distilled R1 models of similar scale while maintaining substantially lower inference costs.

§ 1.3 Rethinking Reasoning Distillation for Reasoning LLMs

Recent large reasoning models (LRMs) (Jaech et al. 2024; OpenAI, 2025; Guo et al. 2025a; Team, 2025), have achieved remarkable success in complex reasoning tasks such as mathematical reasoning (Hendrycks et al., 2021b), code generation (Jain et al., 2025). Models like DeepSeek-R1 (Guo et al. 2025a) employ long CoT reasoning with self-reflection, verification, and correction mechanisms, but their deployment is limited by heavy computational costs (Sui et al. 2025). Knowledge distillation (Hinton et al., 2015) offers a promising approach to develop smaller, efficient reasoning models, yet the process of transferring long CoT reasoning skills remains poorly understood (Li et al., 2023; Wang et al., 2023; Chen et al. 2025). Open questions persist regarding the role of student model size, teacher model scale, domain generalization, training CoT

quality, and architecture compatibility in reasoning distillation (Shridhar et al., 2023; Chen et al., 2023; Chenglin et al., 2024b).

In Chapter 5, we present the first systematic investigation of knowledge distillation for long CoT reasoning, focusing on complex mathematical reasoning as a representative domain. Our experiments across diverse teacher models and multiple student architectures reveal that (i) student model scale strongly affects reasoning transfer, with architecture-specific variations; (ii) larger teacher models do not always yield better students, exhibiting a non-monotonic teacher–student relationship; (iii) mid-sized teachers often enable better cross-domain generalization than larger ones; (iv) reasoning quality in training data outweighs quantity (Ye et al., 2025) in distillation effectiveness; and (v) architectural alignment between teacher and student models is critical for optimal performance.

§ 1.4 Thesis Statement

This thesis proposes and evaluates a set of methods aimed at enhancing the *generalization*, *efficiency*, and *transferability* of reasoning in large language models. Using claim verification and mathematical reasoning as a representative reasoning-intensive tasks, we demonstrate how structured synthetic data generation, computationally efficient reasoning strategies, and empirically informed distillation practices can improve LLM performance across diverse domains and complex reasoning scenarios.

§ 1.5 Thesis Contributions

The main contributions of this thesis are as follows. Our contributions for synthetic data generation for zero-shot claim verification are:

- We propose a step-by-step prompting-based framework for generating synthetic claim verification training samples using a secondary model without relying on external corpora or costly human annotation.
- We show that models trained with our step-by-step synthetic data generation method can outperform prompting-based baselines and prior prompt-based synthetic data generation methods for fact verification. Additionally, our method improves upon a prior baseline method that uses external human-selected corpora.

Our contributions for efficient reasoning for context-grounded claim verification are:

- To the best of our knowledge, we are the first to develop efficient small reasoning LLMs using reinforcement learning with verifiable rewards for the context-grounded claim verification tasks.
- We show that using reinforcement learning with verifiable rewards can achieve better performance on multiple context-grounded claim verification tasks compared to the out-of-box reasoning LLMs in the same size.

Our contributions for rethinking reasoning distillation for reasoning LLMs are:

- We present, to the best of our knowledge, the first systematic investigation of knowledge distillation for long CoT reasoning capabilities, with 56 teacher-student model combinations in total.

- We demonstrate that mid-sized teacher models may outperform larger models in both in-domain and out-of-domain evaluations.
- We also provide actionable insights for developing more efficient reasoning models.

§ 1.6 Thesis Outline

The rest of this thesis is structured as follows. Chapter 3 presents SynthVerify, detailing its structured domain specification, plan-guided evidence synthesis, and aspect-conditioned claim generation modules. Chapter 4 describes the efficient reasoning training framework for context-grounded tasks. Chapter 5 analyzes reasoning distillation efficiency, evaluating teacher–student relationships and their impact on reasoning transfer. Chapter 6 concludes with a summary of findings, discussion of limitations, and directions for future research. We add the used prompts and the details of evaluation benchmarks of each chapter in the appendix.

Chapter 2

Related Work

§ 2.1 Synthetic Data Generation for Zero-Shot Claim Verification

In this section, we first introduce several LLM-based claim verification systems and point out their limitation. And then we present the existing synthetic data generation methods for the fact verification tasks.

2.1.1 LLM-based Claim Verification

Numerous datasets have been compiled for fact verification across various domains, such as politics (Vlachos and Riedel, 2014), encyclopedias (Thorne et al., 2018; Thorne et al., 2021; Eisenschlos et al., 2021), news (Pérez-Rosas et al., 2018), climate (Diggelmann et al., 2020), science (Wadden et al., 2020a), and healthcare (Kotonya and Toni, 2020a). (Honovich et al., 2022) consolidated several datasets to evaluate the ability to measure input-output consistency. The statements in these datasets are typically single sentences,

generated either by crawling specific websites (Vlachos and Riedel, 2014), manually altering factual sentences (Thorne et al., 2018), or reformulating QA pairs (Thorne et al., 2021). In this work, we study the problem of LLM-based zero-shot claim verification across multiple domains.

2.1.2 Synthetic Data Generation for Fact Verification

Recent advancements in synthetic data generation have significantly enhanced fact verification systems by reducing reliance on costly human annotations. Pan et al. (2021) introduced QACG, a framework that generates claims from Wikipedia by transforming question–answer pairs into supported, refuted, or unverifiable statements. Building upon this, Wright et al. (2022) proposed CLAIMGEN-BART and KBIN to automatically produce atomic scientific claims and their negations in the biomedical domain, enabling zero-shot fact checking with performance reaching up to 90% of fully supervised models. Extending to multimodal data, Bussotti et al. (2024) developed UNOWN, a system that synthesizes training examples from both textual and tabular sources. Collectively, these works underscore the effectiveness of synthetic data in scaling fact verification across diverse domains and data modalities. In contrast to these works, we propose a step-by-step prompting-based generation framework that does not rely on external input documents and remains scalable across diverse domains.

§ 2.2 Efficient Reasoning for Context-Grounded Claim Verification

In this section, we first present recent state-of-the-art large reasoning models and then we introduce the prompting-based Decompose-then-Verify paradigm for the context-grounded claim verification task.

2.2.1 Large Reasoning Models

Recent advancements in large reasoning models (LRMs), such as OpenAI o1 (Jaech et al. 2024), o3-mini (OpenAI, 2025), DeepSeek R1 (Guo et al. 2025a), Kimi k1.5 (Team et al. 2025), and QwQ-32B (Team, 2025), have demonstrated significant improvements across various NLP tasks. Recent research has explored their adaptation to broader domains. For instance, He et al. (2025) introduced a framework for inference-time reasoning in general machine translation, leveraging human-aligned chain-of-thought (CoT) fine-tuning followed by reinforcement learning. In document reranking, Zhuang et al. (2025) developed Rank-R1, which integrates reinforcement learning to enhance LLM-based ranking decisions. Distinct from these approaches, our work focuses on developing an efficient reasoning LLM specifically for context-grounded claim verification. Our objective is to design a model capable of systematically reasoning over claims while maintaining efficiency and adaptability across different verification scenarios.

2.2.2 LLM-Based Context-Grounded Claim Verification

LLM prompting has been widely explored as an effective approach for claim verification (Zhang and Gao, 2023b; Wang and Shu, 2023; Dammu et al., 2024; Zhao et al., 2024a; Wang et al., 2025). To encourage explicit reasoning before producing a final verdict, various prompting techniques have been proposed. For example, Zhang and Gao (2023b) introduced a hierarchical step-by-step prompting strategy tailored for news claim verification, while Wang and Shu (2023) developed a first-order logic-guided knowledge-grounded framework to enhance explainability in claim verification. Additionally, Dammu et al. (2024) proposed a human-centric framework designed to generate detailed annotations aligned with users' informational and verification needs. Despite their effectiveness, these prompting-based methods often rely on extensive human expertise, making them less adaptable when transitioning LLMs to new domains. In contrast, our approach aims to develop a generalizable reasoning model using reinforcement learning, enabling systematic reasoning for context-grounded claim verification without the need for extensive manual intervention.

§ 2.3 Rethinking Reasoning Distillation

In this section, we first introduce how recent large reasoning models leverage long CoTs in various reasoning tasks. Then, we present several prior works using distillation techniques to develop small and efficient models.

2.3.1 Chain-of-Thought Reasoning

Recent advancements in large reasoning models (LRMs), such as OpenAI o1 (Jaech et al. 2024), o3-mini (OpenAI, 2025), DeepSeek R1 (Guo et al. 2025a), Kimi k1.5 (Team et al. 2025), and QwQ-32B (Team, 2025), have demonstrated significant improvements across various NLP tasks. Recent research has explored their adaptation to broader domains. For instance, He et al. (2025) introduced a framework for inference-time reasoning in general machine translation, leveraging human-aligned chain-of-thought (CoT) fine-tuning followed by reinforcement learning. In document reranking, Zhuang et al. (2025) developed Rank-R1, which integrates reinforcement learning to enhance LLM-based ranking decisions. Distinct from these approaches, our work focuses on developing an efficient reasoning LLM specifically for context-grounded claim verification. Our objective is to study the distillation patterns when fine-tuning non-reasoning models on the reasoning data generated by these large reasoning models.

2.3.2 Efficient Reasoning Models

Small efficient reasoning models (SRMs) have emerged as a promising approach to improve the efficiency of reasoning models. Unlike LRMs, SRMs are compact and can exhibit distinct capabilities and cognitive trajectories compared to LRMs (Wang et al., 2025). Researchers have explored various techniques to develop SRMs, including knowledge (Cho and Hariharan, 2019; Zhang et al., 2023), other model compression techniques, and reinforcement learning (Feng et al., 2025). Efficient reasoning models have also been explored through prompt-guided techniques, which explicitly instruct LLMs to generate fewer reasoning steps (Sui et al., 2025). This approach can be

highly effective in improving the efficiency of reasoning models, with various methods proposing different prompts to ensure concise reasoning outputs. For example, Token-Budget proposes setting a token budget in prompts to reduce unnecessary reasoning tokens, while TALE-EP estimates a reasonable token budget using a training-free, zero-shot method (Sui et al., 2025). Overall, small efficient reasoning models have the potential to provide a cost-effective solution for real-world inference, and researchers are actively exploring new techniques to improve their efficiency and effectiveness. In our work, we focus on the knowledge distillation setting.

Chapter 3

Enhancing Zero-Shot Claim Verification through Step-by-Step Synthetic Data Generation

Claim verification, the task of determining whether a claim is supported by given evidence, has emerged as a crucial component in combating misinformation and ensuring information integrity (Litou et al., 2017; Hassan et al., 2017; Shu et al., 2017). While LLMs have shown impressive capabilities in various NLP tasks, LLMs face several critical challenges in claim verification (Zhang and Gao, 2023a; Guan et al., 2024; Augenstein et al. 2024; Quelle and Bovet, 2024).

A primary concern is the lack of domain-specific knowledge during verification. For example, in Figure 3.1, we show that in a popular pre-training dataset, CommonCrawl, there exists some underrepresented domains, like financial or medical-related. As we can see, most pretraining data are focusing on some popular domains, like Wikipedia and web, social media or newswire. For some domains like financial or medical-related, they are not popular in the pretraining data.

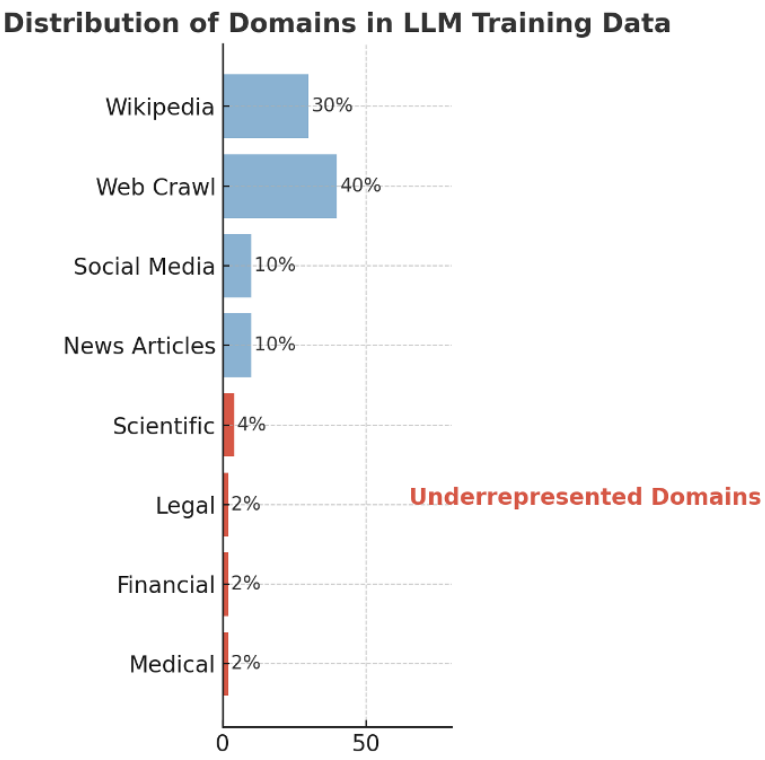

Figure 3.1: We randomly selected 100 preprocessed chunks from CommonCrawl training dataset (Patel, 2020). Then we loosely classify them into different domains based on their contents. Because we care about the distribution of entities and facts, we label any text that contains entities that are in Wikipedia as "Wikipedia."

When evaluating claims in specialized fields such as medicine, law, or scientific research, LLMs often fail to understand the relationships between these domain-specific terminologies (Vladika and Matthes, 2024), which leads to performance drop. For example, in a preliminary experiment, we trained models with the in-distribution data and evaluated them on three domain-related tasks. As shown in Figure 3.2, on each task, we can see a drop compared to the base models. This limitation affects both the evidence analysis step, where models must identify relevant domain-specific information, and the logical reasoning step, where they need to apply domain-specific constraints.

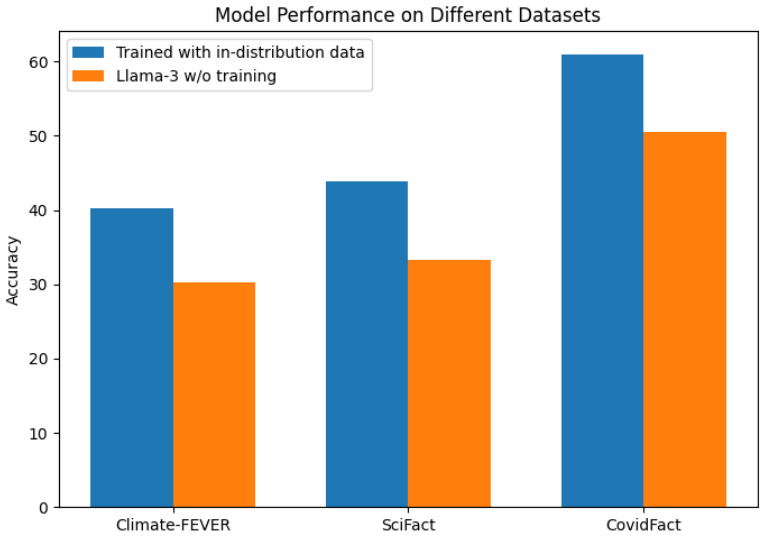

Figure 3.2: We trained models with the in-distribution data and evaluated them on three domain-related tasks, Climate-FEVER (Diggelmann et al., 2020), SciFact (Wadden et al., 2020a) and CovidFact (Saakyan et al., 2021), respectively.

Fine-tuning on domain-specific datasets can help LLMs mitigate these issues. While existing datasets like FEVER (Thorne et al., 2018), and VitaminC (Schuster et al., 2021) have driven progress in this field, they often focus on narrow domains (e.g., Wikipedia articles) or specific types of claims. This specialization leads to poor generalization when systems encounter claims from previously unseen domains (Zhu et al., 2022; Nan et al., 2022; Gu et al., 2023). Furthermore, the manual creation of claim verification datasets is both time-consuming and expensive, particularly when aiming to cover multiple knowledge domains with sufficient depth and diversity.

Synthetic data generation has emerged as a promising solution to address this challenge (Pan et al., 2021; Wright et al., 2022; Bussotti et al., 2024). Despite its potential, existing approaches face notable limitations. Some methods lack scalability across diverse domains, making them impractical for broad deployment (Pan et al., 2021; Wright

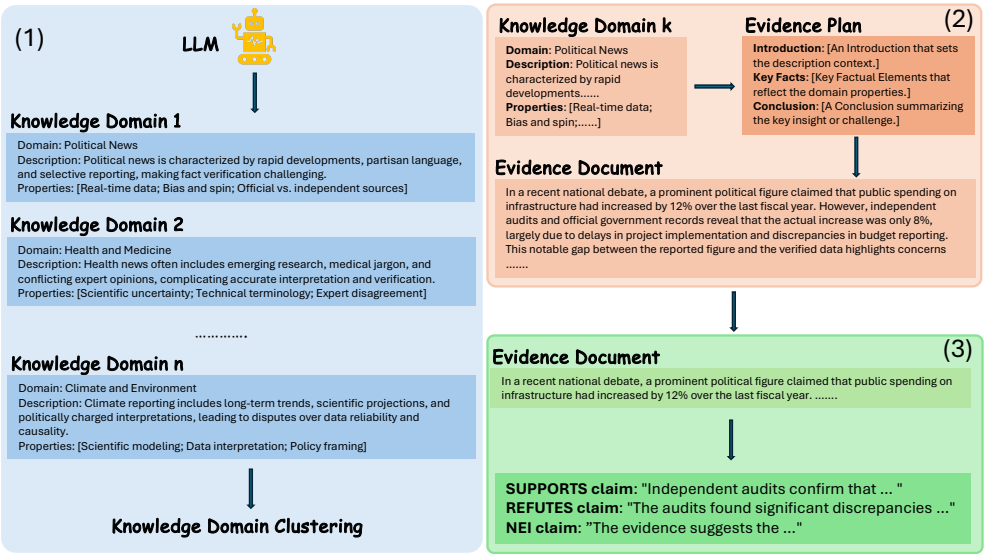

Figure 3.3: Overview of our synthetic data generation framework for claim verification. The framework consists of three main stages: (1) Knowledge Domain Generation (§3.2.2), (2) Plan-guided Evidence Synthesis (§3.2.3), and (3) Aspect-Conditioned Claim Generation (§3.2.4).

et al., 2022), while others rely heavily on external documents to guide data synthesis, introducing dependencies that may not always be available or reliable (Pan et al., 2021; Bussotti et al., 2024). As a result, there is a growing need for more efficient and self-contained strategies that can generalize across multiple domains without compromising data quality or diversity.

To address these challenges, we propose SYNTHVERIFY, a novel step-by-step prompting-based synthetic data generation framework. Our framework carefully controls the generation process to ensure data quality and verifiability. The core insight is that guiding generation with domain-specific claim patterns and structured evidence plans, without requiring access to external corpora or sacrificing generalizability.

First, we develop a structured domain specification stage that captures the nature of do-

main knowledge and verification requirements, providing a foundation for domain-aware claim verification. We then employ a simple but effective evidence synthesis method that provides clear verification signals, addressing the challenge of generating realistic and verifiable evidence. Building upon this stage, we implement an aspect-based claim generation approach that ensures both diversity in claim types and verifiability of generated claims.

Through extensive evaluation, we demonstrate that our synthetic data generation approach leads to prominent improvements on multiple claim verification tasks. Models trained on our synthetic dataset show improved zero-shot transfer capabilities, better handling of complex claims, and more robust verification across diverse knowledge domains. Our further analysis reveals that the structured nature of our generation process helps models learn generalizable verification strategies rather than superficial patterns specific to particular domains. Overall, our main contributions are:

- We propose a step-by-step prompting-based framework for generating synthetic claim verification training samples using a secondary model without relying on external corpora or costly human annotation.
- We show that models trained with our step-by-step synthetic data generation method can outperform prompting-based baselines and prior prompt-based synthetic data generation methods for fact verification. Additionally, our method improves upon a prior baseline method that uses external human-selected corpora.

§ 3.1 Background and Problem Setup

Problem Setup Following the prior work (Thorne et al., 2018; Eisenschlos et al., 2021; Schuster et al., 2021; Wadden et al., 2020a), given a sentence-level claim c and a set of sentence-level evidences * $E = \{e_1, ..., e_{|E|}\}$, the task of claim verification is to determine whether E supports or refutes c, or if there is insufficient information to make a determination. Here, we assume that each given claim c is *check-worthy* and can be fully verified based on the evidence set E, without relying on the external context. Formally, given an LLM M, we define a verification function $f_M : C \times \mathcal{E} \rightarrow \mathcal{Y}$, where C represents the space of possible claims, $\mathcal{E}$ denotes the space of evidence sets and $\mathcal{Y} = \{\texttt{SUPPORTS}, \texttt{REFUTES}, \texttt{NOT\_ENOUGH\_INFO}\}$ is the set of verification labels.

For a given claim-evidence pair (c, E), the LLM M performs the verification through three main steps. First, M performs evidence analysis by processing each evidence $e_i \in E$ to extract and understand the information relevant to the claim. Second, M performs logical reasoning to determine the relationship between the extracted evidence and the claim, applying its understanding of the domain knowledge. Finally, M assigns a verification label $y \in \mathcal{Y}$ based on the aggregated evidence and reasoning outcome.

Challenges of LLM-based Claim Verification LLM-based claim verification systems face two significant challenges in real-world applications. First, although LLMs acquire extensive domain knowledge during pre-training, they often demonstrate limited generalization capabilities in unseen domains (Pan et al., 2023). Human-annotated, domain-specific datasets can substantially enhance performance in such contexts. How-

*We acknowledge that retrieval is a critical component of real-world fact-checking pipelines. However, our evaluation setup assumes that claims are accompanied by evidences, focusing specifically on the claim verification stage rather than evidence retrieval.

ever, the high costs associated with annotation, both in terms of time and resources, constrain the coverage, diversity, and scalability of these datasets. Second, LLMs are susceptible to reasoning errors and over-reliance on spurious correlations (Tang et al., 2023), which can lead to incorrect veracity assessments. This is particularly problematic when claims require complex reasoning or contextual understanding that spans multiple knowledge types. In this work, we introduce an innovative synthetic data generation approach designed to bolster the capabilities of LLMs in addressing these challenges effectively.

§ 3.2 Methodology: Zero-Shot Claim Verification Data Generation

In this section, we describe the details on how we design a structured approach to generate synthetic claim verification data that enables zero-shot transfer across diverse domains. Our goal is to construct a dataset $\mathcal{D} = \{(D_i, C_i, E_i, y_i)\}_{i=1}^{N}$ of N instances, where D_i represents a specification of knowledge domain, C_i is a claim within that domain, E_i is the corresponding evidence, and $y_i \in \{\texttt{SUPPORTS}, \texttt{REFUTES}, \texttt{NOT\_ENOUGH\_INFO}\}$ is the verification label.

3.2.1 Overview

As shown in Figure 3.3 , our synthetic data generation framework consists of three stages. First, the **Knowledge Domain Generation** stage defines the scope and constraints of the target domain, organizing knowledge hierarchically to ensure generated content remains

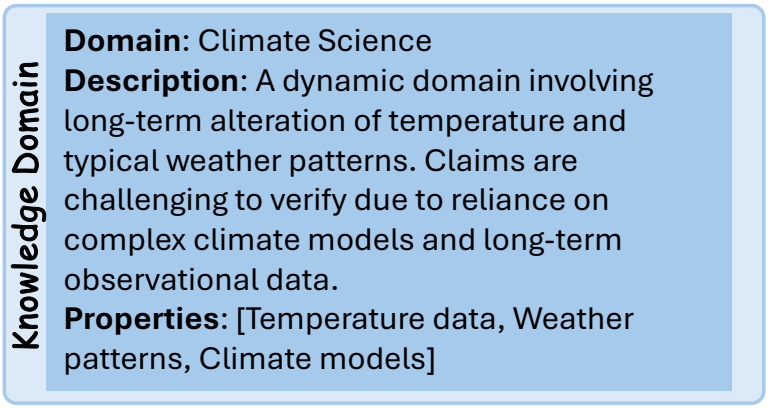

Figure 3.4: An example of the generated knowledge domains. Our **Knowledge Domain Generation** module (§3.2.2) generates a diverse set of knowledge domains, coupled with a clustering-based diversity enforcement.

focused and relevant. Second, the **Plan-guided Evidence Synthesis** stage synthesizes informative evidence documents by leveraging the knowledge domains from the previous step. Third, the **Aspect-Conditioned Claim Generation** stage generate three different types of claims (Supported/Refuted/Not Enough Info) conditioned on each generated evidence document.

3.2.2 Knowledge Domain Generation

First, to collect a diverse set of domains where evidence and claims are sampled, a set of knowledge domains are randomly sampled from an LLM as structured triplets. Each knowledge domain O_i is formally characterized by: $O_i = (D_i, S_i, P_i)$, where D_i is the domain name, S_i is a basic description of the domain explaining why claims in this domain are challenging to verify. $P_i = \{p_1, ..., p_k\}$ is a set of domain properties. The knowledge domain O_i will be used to guide the generation of the evidence documents. The additional information S_i and P_i can lead to more non-trivial and domain-diverse

generations compared to using D_i alone.

We prompt an LLM to generate a list of knowledge domains that require claim verification in the real-world. The detailed prompt and a sample of generated knowledge domain is shown in Appendix. However, naive LLM prompting often results in redundant or overlapped domains when generating full set of knowledge domains O_i, leading to low coverage (Ding et al., 2023).

To ensure broad coverage across knowledge spaces, we employ an iterative clustering-based generation and filtering process. Let $f(\cdot)$ be an embedding function that maps each domain D_i to a d-dimensional semantic space. For a batch of k generated knowledge domains, we compute their embeddings: $\mathbf{E} = \{\mathbf{e}_i = f(D_i)\}_{i=1}^{k}$. To identify and remove near-duplicate domains, we apply density-based clustering DBSCAN (Ester et al. 1996):

$$\text{clusters} = \text{DBSCAN}(\mathbf{E}, \epsilon, \text{min\_samples}) \tag{3.2.1}$$

where ϵ and min_samples are empirically determined parameters. This process continues iteratively until we obtain n diverse domains.

3.2.3 Plan-guided Evidence Synthesis

In this stage, we generate an informative evidence document by leveraging the knowledge domain from the previous step. However, in a preliminary experiment, prompting LLMs to get the evidence document based on the knowledge domain resulted in more generic content that lack sufficient details to generate meaningful and check-worthy claims in the future stage. To address this issue, inspired by some prior studies in the story generation (Yang et al., 2022; Yang et al., 2023), we generate the evidence document from the knowledge domain in two steps.

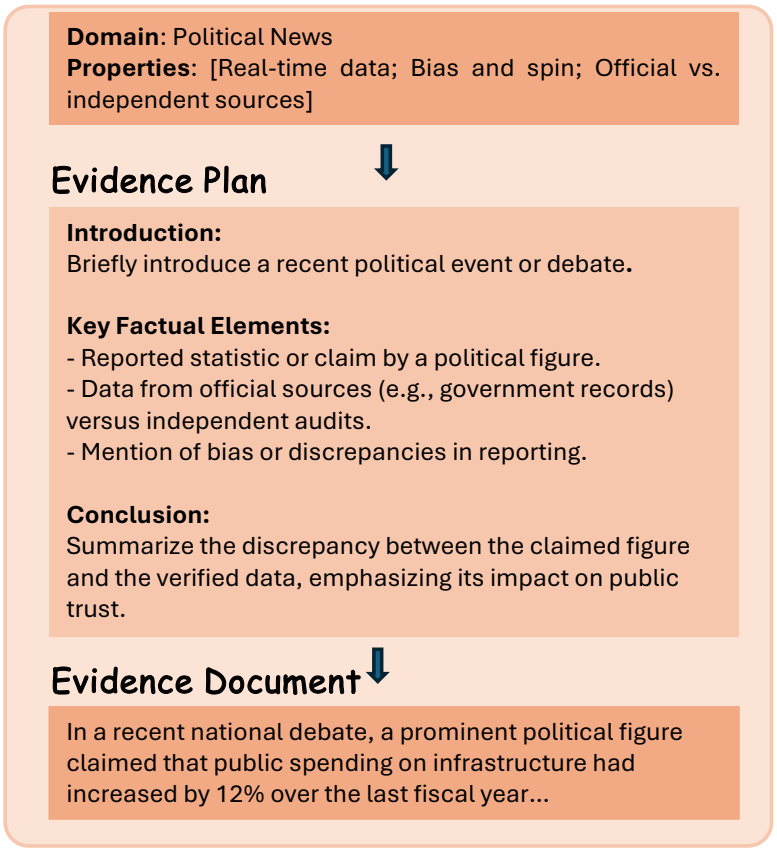

Figure 3.5: Given a knowledge domain, our **Plan-guided Evidence Synthesis** module (§3.2.3) first generates an evidence plan (upper), then it outputs a concise evidence document.

The key idea is to leverage a knowledge domain to generate a structured plan, which is subsequently expanded into a coherent evidence document. This two-step process ensures that the final output adheres not only to the factual but also to the stylistic nuances of the target domain but also maintains a consistent academic structure.

Step 1: Evidence Plan Generation. We first prompt the LLM to generate a concise plan that serves as a blueprint for the evidence document. The plan is designed to capture a brief domain-dependent statement and several specific details drawn from the

domain properties.

Step 2: Evidence Document Synthesis. Once the plan is generated, the next step is to expand it into a full evidence document. The model is instructed to integrate all plan components into a polished paragraph written in an information-preserving style. This expansion ensures that the evidence document is comprehensive, coherent, and reflective of the domain-specific characteristics.

Figure 3.5 provides an overview of the plan-guided evidence synthesis process. By grounding the generation process in a structured plan derived from the knowledge domain, our method achieves enhanced control over content structure, ensuring both consistency and domain-specific nuance. This approach has demonstrated improved quality in synthetic evidence, thereby contributing to more robust downstream claim verification across diverse domains.

3.2.4 Aspect-Conditioned Claim Generation

As shown in Figure 3.6, we generate three different types of claims (Supported/Refuted/Not Enough Info) conditioned on the generated evidence document. Our approach increases the diversity and complexity of the generated claims through a aspect-based generation method that ensures both structural diversity and verifiability while maintaining semantic coherence within each source document. We first generate the supported claim based on the evidence document, then generate the other two claims conditioned on the supported claims.

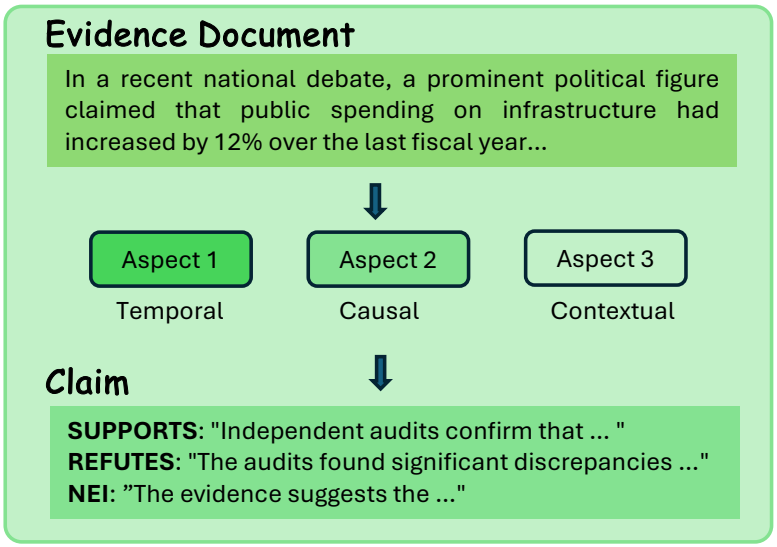

Figure 3.6: Our **Aspect-Conditioned Claim Generation** module (§3.2.4) takes in the generated evidence. It first generates three aspect features (e.g., temporal, causal and contextual) (middle), then it outputs the three claim (lower) based on the generated key aspects.

Supported Claim Generation. We first extract several *key aspects* from the evidence document. These key aspects are concise descriptors that capture different dimensions or viewpoints present in the evidence (e.g., contextual background, numerical details, source credibility, or timing). By subsequently conditioning the claim generation on each identified aspect, the model produces multiple supported claims that, while all factually correct with respect to the evidence, emphasize different details or angles.

Given an evidence document and the associated knowledge domain, the model is prompted to generate a list of key aspects. Each key aspect is a short description that highlights a particular facet of the evidence. For example, one aspect might focus on the statistical discrepancy mentioned in the text, while another might highlight the reliability of the sources. This step ensures that the extracted aspects are aligned with the unique properties defined in the knowledge domain.

For each key aspect, a secondary prompt is created that conditions the model on both the original evidence and the specified aspect. The prompt instructs the LLM to generate a supported claim that is not only consistent with the evidence but also emphasizes the conditioned aspect. This encourages the model to explore multiple views of the evidence, thereby producing a diverse set of claim formulations.

Refuted Claim Generation. Starting from a supported claim, we generate refuted claims by applying a series of controlled perturbations designed to introduce discrepancies between the claim and the underlying evidence. These perturbations include: (1) **Entity Substitution:** Replacing critical entities with alternative, contextually plausible candidates. (2) **Temporal Modification:** Adjusting time-related references while preserving overall historical coherence. (3) **Relationship Reversal:** Inverting the relational dynamics between entities to alter the claim's meaning. (4) **Attribute Modification:** Modifying specific properties or characteristics to create a divergence from the original details.

Not-Enough-Info (NEI) Claim Generation. Given a supported claim, the model is instructed to modify it by removing or replacing the identified key elements with more vague, qualitative, or uncertain language. For example, a claim stating "The study shows a 70% reduction in hospitalization rates" may be transformed into "The study suggests a significant reduction in hospitalization rates," where the exact figure is omitted.

Rate sample quality: Evaluate the quality of the claim–evidence pair using the following criteria (1–5 scale):

Fluency: Are both the claim and the evidence well-formed and natural within the domain?
Plausibility: Does the claim sound like a realistic statement?
Evidence informativeness: Does the evidence contain sufficient detail to verify the claim?
Domain relevance: If a domain label is given, is the content appropriate to that domain?

Figure 3.7: The prompt for filtering out low-quality samples.

3.2.5 Quality Control

Before fine-tuning LLMs using the generated synthetic data, we adopt the LLM-as-a-Judge paradigm (Zheng et al., 2023) to filter out low-quality samples. Following the prior work (Bai et al. 2022), we prompt LLM with a rubric to rate fluency, plausibility, informativeness, and domain relevance on a 1 to 5 scale for each synthetic sample. The prompt is shown in Figure 3.7.

3.2.6 Human Evaluation

In order to investigate the accuracy and quality of the labels on our generated (evidence, claim) pairs. We randomly sample 50 examples from the generated dataset. Each example was annotated with a label by three NLP researchers. The examples are unedited outputs from the models, with no revisions made during annotation. The average Cohen's κ score between annotators is 65.83%, indicating substantial agreement.

Instruction: You are given a claim and evidence. Your task is to verify the claim based solely on the evidence provided. Analyze the claim and evaluate the evidence. Based on your evaluation, return one of the following labels: "supports", "refutes" or "not enough info".

Evidence: {EVIDENCE}
Claim: {CLAIM}
Label: {LABEL}

Figure 3.8: The instruction template that we use for compiling the final instruction tuning dataset to instruct tuning LLMs for the claim verification.

Furthermore, a majority label (agreed upon by 2 out of 3 annotators) was obtained for 94% of examples, while all three annotators reached unanimous agreement on 72% of examples. The model achieved a high accuracy of 79.6% against the majority label and 88.5% against the unanimous label. The Cohen's κ scores between the model's labels and both the majority and unanimous labels are 69.73% and 82.46%, respectively.

3.2.7 The Final Instruction Tuning Dataset

Using our proposed framework SYNTHVERIFY, we generate a domain-diverse synthetic claim verification dataset for training zero-shot verification models. We use the generated dataset to fine-tune the various LLMs (e.g., Llama-3.1-8B). To achieve this, we combine the predefined instruction template in Figure 3.8 and instance input into a single prompt and train the LLMs in a standard supervised learning setup to generate the corresponding instance output.

Model	FEVER	FM2	VitaminC	C-FEVER	PubHealth	SciFact	CovidFact	FAVIQ	Avg
Qwen2.5-0.5B-Instruct	35.7	50.0	37.0	32.5	31.3	38.1	52.0	50.0	40.8
Qwen2.5-1.5B-Instruct	34.0	50.0	33.3	33.3	33.3	34.8	50.0	49.0	39.7
gemma-2-2b-it	55.7	78.5	47.7	49.6	38.7	52.9	66.5	57.5	55.9
Qwen2.5-3B-Instruct	49.7	61.0	38.0	39.0	32.7	51.9	60.5	53.0	48.2
gemma-2-9b-it	37.7	51.0	33.0	38.2	34.7	34.8	63.0	55.5	43.5
Llama-2-13b-hf	33.3	50.0	33.3	33.3	33.7	33.3	50.0	49.5	39.6
InternLM2-Chat-20B	59.3	69.5	47.3	43.1	29.3	54.3	64.0	54.0	52.6
InternLM2.5-20B-Chat	46.0	56.0	47.3	39.0	36.3	36.7	57.0	52.0	46.3
Llama-3.2-1B-Instruct	43.7	53.0	31.3	35.8	35.7	37.1	52.5	56.0	43.1
Llama-3.2-1B-Instruct (Ours)	56.7	64.0	58.0	53.7	44.0	48.1	61.5	60.0	**55.8**
Llama-3.2-3B-Instruct	43.7	63.0	36.7	48.8	38.7	54.8	64.5	57.0	50.9
Llama-3.2-3B-Instruct (Ours)	61.0	75.5	56.0	49.6	51.2	53.3	64.5	60.5	**59.0**
Mistral-7B-Instruct-v0.3	46.0	54.0	36.0	34.1	36.3	36.2	50.5	48.0	42.6
Mistral-7B-Instruct-v0.3 (Ours)	59.0	74.5	54.0	53.7	52.3	52.9	67.0	55.5	**58.6**
Llama-3.1-8B-Instruct	37.0	60.5	32.7	38.2	36.3	38.1	54.5	56.0	44.2
Llama-3.1-8B-Instruct (Ours)	60.3	76.0	56.7	50.4	45.3	59.5	68.0	59.5	**59.5**
Llama-3-8B-Instruct	33.3	50.0	33.3	33.3	33.7	33.3	50.5	51.0	39.8
+ Zero-Shot CoT	43.7	53.0	47.3	39.0	36.3	36.7	52.5	53.0	42.7
+ QACG (Pan et al., 2021)	37.7	53.0	38.0	38.2	35.7	34.8	52.5	52.0	42.7
Llama-3-8B-Instruct (Ours)	**59.3**	**80.5**	**57.0**	**49.6**	**49.3**	**55.7**	**65.5**	**62.5**	**59.9**
InternLM2.5-7B-Chat	44.3	76.5	33.3	35.0	34.0	46.7	67.0	60.0	49.6
+ Zero-Shot CoT	46.0	76.5	36.7	35.8	36.3	48.1	67.0	60.5	50.9
+ QACG (Pan et al., 2021)	46.0	78.5	37.7	42.3	32.7	51.0	67.0	60.0	51.9
InternLM2.5-7B-Chat (Ours)	**64.0**	**83.5**	**58.7**	**53.7**	**49.3**	**57.1**	**71.0**	**65.0**	**62.8**

Table 3.1: Zero-shot performance across different datasets, using Balanced Accuracy (BAcc). For Llama-3 and InternLM2.5 models, we bold the best model for each dataset. Otherwise, we bold the best average score.

§ 3.3 Experimental Setup

Evaluation Benchmarks. In this work, following the prior work (Pan et al., 2023), we use a wide range of existing claim verification benchmarks from various domains: (1) Wikipedia domain: We use FEVER (Thorne et al., 2018), VitaminC (Schuster et al., 2021) and FoolMeTwice (Eisenschlos et al., 2021). These datasets were created based on Wikipedia documents; (2) Climate domain: We use Climate-FEVER (Diggelmann et al., 2020), which consists of real-world claims related to climate change; (3) Science domain: We use SciFact (Wadden et al., 2020a); (4) Health domain: We

use PubHealth (Kotonya and Toni, 2020a); (5) Other domains: we also use Covid-Fact (Saakyan et al., 2021) from the forum domain and FAVIQ (Park et al., 2022) from the question domain in our evaluations. The statistics of these benchmarks are shown in Table A.1. Please refer to Appendix A for more details.

Evaluation Metric. Following the prior work (Laban et al., 2022), we use the Balanced Accuracy (BAcc), the average of recall obtained on each label, as our evaluation metric. Here, BAcc = $\frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}}\frac{\text{TP}_y}{\text{TP}_y+\text{FN}_y}$, where $\mathcal{Y}$ is the set of all class labels (e.g., supports, refutes, not_enough_info), TP_y is the number of true positives for class y, and FN_y is the number of false negatives for class y.

Baselines. (1) Zero-Shot: By default, we compare the model fine-tuned by our synthetic dataset to the zero-shot setting. We use the same prompt as shown in Figure 3.8. (2) Zero-Shot Chain-of-Thought (Kojima et al., 2022): We also compare to the proposed zero-shot CoT prompting method. We append the prompt "Let's think step by step" to the instruction prompt in Figure 3.8. (3) We also compare our method to QACG (Pan et al., 2021), which generates synthetic claims conditioned on the Wikipidia data. We take their released dataset and fine-tune Llama-3 and InternLM2.5 models. (4) Recently, there is a concurrent work (Ashok and May, 2024), which generates synthetic true claims through naive prompting given the evidence. This approach is identical to one of our ablation studies. Thus, we compare our method to it in Section 4.4.

There are also some other works, that we do not compare to, that generate synthetic data for fact verification or hallucination detection tasks. We do not compare to these works, either because they are tailored to a specific domain, or take different types of inputs than us. Wright et al. (2022) generated synthetic claims only for the science

domain, while our method can be more applicable to diverse domains. Tang et al. (2024) generated document-claim pairs from human-selected documents for LLM hallucination detection, which is different from our evaluation goal. Bussotti et al. (2024) generated synthetic claims from both table and text inputs, while we only target text-based input.

Implementation Details. We compare against various state-of-the-art instruction-tuned LLMs, including: Qwen 2.5 (Yang et al. 2024), Gemma 2 (Team et al. 2024), InternLM (Cai et al. 2024), Mistral (Jiang et al. 2023), and Llama-3 family (Dubey et al. 2024).

For the synthetic data generation, we use GPT-4o-mini with temperature 0.9 for all prompting generation. For domain similarity computation, we employ the all-MiniLM-L6-v2 sentence transformer, which provides 384-dimensional embeddings. The clustering parameters ($\epsilon = 0.3$, min_samples = 2). The all prompts, dataset statistics and fine-tuning hyperparameters are available in Appendix A.

§ 3.4 Main Results

Table 3.1 presents a comprehensive evaluation of various language models on eight claim verification benchmarks. Several notable findings emerge from our experimental results.

First, our synthetic data framework demonstrates consistent and substantial improvements across all variants of the model. Most notably, InternLM2.5-7B-Chat with our method achieves the best overall performance with an average balanced accuracy of 62.8, surpassing its base version by a large margin of 13.2 absolute points (from 49.6

to 62.8). This improvement is particularly pronounced in other datasets like FEVER (+19.7 points) and VitaminC (+25.4 points).

Second, we observe that larger model size does not necessarily translate to better claim verification performance. For example, Gemma-2-2B-it, despite its relatively small size, achieves a competitive average accuracy of 55.9, outperforming several larger models such as InternLM2-Chat-20B (52.6) and InternLM2.5-20B-Chat (46.3). This suggests that verification performance is more closely tied to a model's capabilities than its size.

Third, our method shows remarkable consistency in improving performance across different model architectures and scales. The enhancement is particularly evident in the Llama family: Llama-3.2-1B-Instruct improves from 43.1 to 55.8 (+12.7 points), Llama-3.2-3B-Instruct from 50.9 to 59.0 (+8.1 points), and Llama-3.1-8B-Instruct from 44.2 to 59.5 (+15.3 points). This consistent improvement pattern suggests the robustness and generalizability of our approach.

We also observe that our method is better than zero-shot chain-of-thought (CoT) prompting. Zero-shot CoT can usually get some improvements, but fine-tuning models on domain-relevant data would be more beneficial. Furthermore, our method is also better than QACG (Pan et al., 2021). The major reason is that our method benefits from LLMs. LLMs store a lot of domain knowledge during the pre-training stage.

§ 3.5 Further Analysis

In this section, we conduct more analysis studies, comparing our framework to a direct prompting method, analyzing the effect of training data size and the impact of domain

Synthesis Framework	VitaminC	SciFact	FAVIQ
No Training	33.3	46.7	60.0
Direct Prompting	35.0	34.8	58.5
R1	57.0	**56.2**	62.5
R2	54.0	51.0	60.5
R3	56.0	54.3	62.5
Full Framework	**57.3**	**56.2**	**64.0**

Table 3.2: The results of our ablation studies. **R1** means removing knowledge domain specification. **R2** represents that removing the evidence plan generation. **R3** is removing the key aspect generation.

knowledge.

3.5.1 Ablation Study

To understand the contributions of each component of our pipeline, we systematically evaluate our framework's three main components. We use the same domains as our framework to generate 20k samples for each setting. We use InternLM2.5-7B-Chat as our base model and evaluate on Vitaminc (Schuster et al., 2021), SciFact (Wadden et al., 2020a), and FAVIQ (Park et al., 2022) datasets. The full prompt is shown in Appendix.

Direct Prompting The most straightforward way is the direct prompting without any intermediate step. This method directly prompts the LLM to generate the claim-evidence pairs when given a domain label. Table A.20 shows the prompt we use to perform direct prompting.

Removing knowledge domain Specification (R1) When removed, we replace the structured knowledge domain with simple domain labels (e.g., climate science), eliminating the domain-specific constraints. This tests the impact of structured knowledge

domain representation in our framework. The prompt is shown in Table A.17.

Removing Evidence Plan Generation (R2) We replace our evidence synthesis module with basic prompting generation, removing the evidence plan generation. This tests the value of the evidence plan in guiding the evidence document generation. The prompt is shown in Table A.18.

Removing Key aspect Generation (R3) Without this component, we use basic prompt-based generation without key aspects to generate the supported claim. This method is the same as the concurrent work Ashok and May (2024), which generates true claims given the evidence. This evaluates the benefit of our diversity-oriented claim generation approach over simple prompting. The prompt is shown in Table A.19.

Table 3.2 shows the results of our ablation studies. The results demonstrate the effectiveness of our proposed framework. We can observe that performance decreases slightly when removing a specific component. The results also show consistent improvements over both the no-training baseline and direct prompting approaches. Specifically, our full framework achieves the highest performance across all datasets. The improvement is particularly pronounced for VitaminC, where our framework outperforms the no-training baseline by 24 points. Interestingly, direct prompting shows mixed results compared to no training, performing slightly better on VitaminC (+1.7 points) but worse on SciFact (-11.9 points). These findings strongly suggest that our designed framework contributes significantly to the framework's overall effectiveness.

Synthesis Framework	VitaminC	SciFact	FAVIQ
No Training	33.3	46.7	60.0
Direct Prompting (30k)	38.0	48.1	60.0
Full Framework (5k)	**56.0**	**55.7**	**62.5**

Table 3.3: Ablation study by considering the budget constraints.

3.5.2 Impact of the Budget Control

Since our data synthesis framework uses more API calls compared to the direct prompting method. In order to investigate the benefit of the structure-aware data synthesis, we also compare our framework to the direct prompting method incorporating budget constraints. Since it is hard to control the synthesis budget directly, we conduct an additional experiment under the rough estimation. The table Table 3.3 compares a model fine-tuned with 30k samples from direct prompting against a model fine-tuned with only 5k samples generated using our framework. As we can see, our framework is still significantly better than the direct prompting even under the similar budget.

3.5.3 Effect of the Training Dataset Size

Here we study the effect of the size of the training dataset. In particular, we study how the performance of the Llama-3.2-1B-Instruct and Llama-3-8B-Instruct models varies when trained on different sizes of synthetic instruction tuning data. Figure 3.9 shows that training on more steps typically improves performance. We find that fine-tuned LLMs on all datasets reach the peak performance generally after 30,000 steps. It means training on more of the synthetic data may not be necessary.

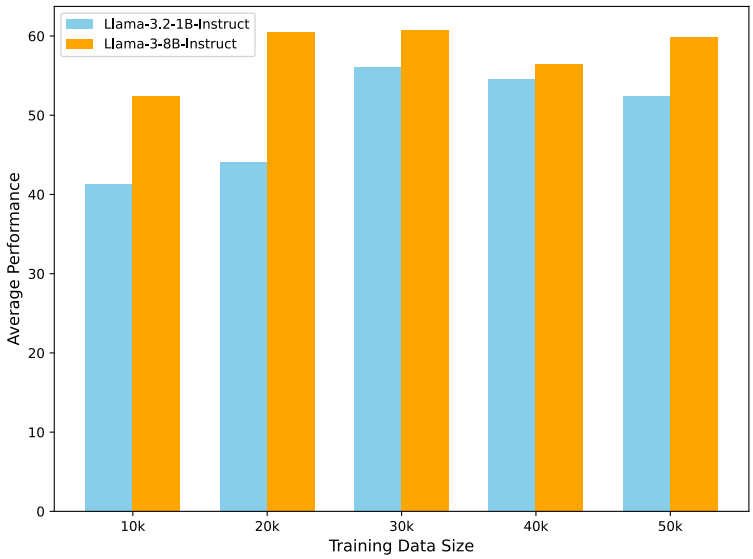

Figure 3.9: The results of how the performance of Llama-3.2-1B-Instruct and Llama-3-8B-Instruct models varies when trained on different sizes of synthetic instruction tuning data.

3.5.4 Impact of Domain Knowledge

Fine-tuning models on unseen domains can improve the verification performance (Pan et al., 2023). We hypothesize that the synthetic data generated by SYNTHVERIFY can benefit models further from the general verification training. To investigate the effectiveness of SYNTHVERIFY in domain adaptation setting, we conduct experiments comparing the performance of two Llama models (3.2-1B and 3-8B) across three distinct verification datasets. We examine three training configurations for each model: the base instruction-tuned model, fine-tuning with FEVER data only, and fine-tuning with both FEVER and our synthetic data generated by SYNTHVERIFY.

The results in Table 3.4 demonstrate that SYNTHVERIFY consistently enhances verification performance across all domains. The improvement is particularly pronounced in

Model	C-FEVER	SciFact	CovidFact
Llama-3.2-1B-Instruct	35.8	37.1	52.5
w/ FEVER	38.2	43.8	54.5
w/ (SynthVerify & FEVER)	42.3	47.7	57.0
Llama-3-8B-Instruct	33.3	33.3	50.5
w/ FEVER	40.3	43.8	61.0
w/ (SynthVerify & FEVER)	50.4	58.6	67.0

Table 3.4: Performance comparison across C-FEVER, SciFact, and CovidFact datasets.

the Llama-3-8B model, where the combination of SynthVerify and FEVER training leads to substantial gains, improving from 33.3 to 58.6 on SciFact. Notably, even the smaller 3.2-1B model shows consistent improvements with SynthVerify, suggesting that our approach effectively improves domain adaptation across model sizes.

§ 3.6 Conclusion

We present SynthVerify, a novel synthetic data generation framework for synthesizing domain-diverse claim verification datasets. Our approach addresses the critical challenge of limited domain coverage in existing datasets through automated generation of claim-evidence pairs across novel domains. Experimental results demonstrate that models trained on SynthVerify-generated data significantly outperform baselines on multiple benchmarks. This success highlights that the step-by-step nature of our generation process helps models learn generalizable verification strategies.

x

LLM Prompting Since our method is prompting an LLM to generate synthetic data. Some less powerfull LLMs may not be able to do this task. We will explore how to

self-improve using small LLMs.

Semantic Redundancy in Generated Data While SYNTHVERIFY successfully generates valuable training datasets for zero-shot claim verification, it does not ensure that each synthesized claim is semantically unique. For zero-shot applications, this limitation is less problematic, as such models are designed to adapt to any provided claims and evidence to determine veracity. Addressing this redundancy would improve the applicability of SYNTHVERIFY for broader training scenarios.

Label Quality Given the fully automated nature of our data synthesis process, some noise in the veracity labels is unavoidable. This noise likely underestimates the true impact of training data domain diversity on zero-shot claim verification, as models trained on SYNTHVERIFY are inadvertently exposed to, and learn to predict, this noise. Ideally, a dataset with the same level of domain diversity as SYNTHVERIFY, but with gold-standard veracity labels, would serve as the basis for our experiments. The lack of such a dataset underscores the motivation for our work.

Chapter 4

Efficient Reasoning for Claim Verification via Structured Chain-of-Thought Distillation and Reinforcement Learning

Building on the foundation laid in Chapter 3, which focuses on improving domain generalization through structured synthetic data generation, we next turn our attention to the reasoning process itself. While high-quality, domain-diverse training data enhances model performance, many real-world verification tasks involve complex, context-rich evidence that demands efficient and structured reasoning (Wang and Shu, 2023; Strong et al., 2024; Zhao and Flanigan, 2025). For example, as shown in Figure 4.1, the claim need to be varied against to both the provided documents and tables.

In Chapter 4, we shift from data-centric improvements to model-centric strategies by proposing an efficient reasoning framework. This framework aims to reduce the computational cost of claim verification without sacrificing accuracy, particularly in settings where input evidence is lengthy or heterogeneous.

Current LLM-based approaches (Min et al., 2023; Chern et al. 2023; Zhao et al. 2023;

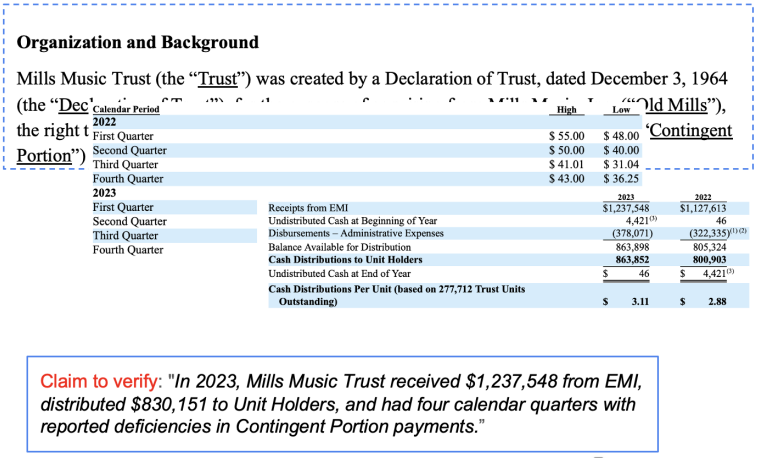

Figure 4.1: In the real-world scenarios, the context of a claim verification task could be more complex, given different documents and the associated tables. Models need to verify the claim based on the grounded context.

Kamoi et al., 2023a; Wei et al. 2024; Song et al., 2024; Wang et al., 2024) often follow the Decompose-Then-Verify paradigm to reduce the complexity of claim verification. These methods first utilize the Chain-of-Thought (Wei et al., 2022) capabilities of LLMs to break down a complex claim into several independent sub-claims. Next, LLMs assess the veracity of each sub-claim using either provided or retrieved evidence. A claim is ultimately considered true only if all of its sub-claims are verified as correct. While this strategy simplifies the verification process for complex claims, the decomposition step is prone to errors (Min et al., 2023; Chern et al. 2023; Zhao et al. 2023), potentially producing ambiguous or incorrect sub-claims, which can significantly degrade overall verification accuracy. In addition, this strategy usually requires a complex and context-specific prompt designed by the human expert (Min et al., 2023; Wang et al., 2024).

Recent advancements in reasoning LLMs, such as OpenAI o1 (Jaech et al. 2024) and DeepSeek R1 (Guo et al. 2025a), have demonstrated the power of automatically generating a long reasoning chain-of-thought (CoT) (Wei et al., 2022) without human

efforts across specialized domains like mathematics and code generation (Guo et al. 2025a). Reinforcement learning (RL) techniques have further enhanced these models, enabling self-reflective reasoning behaviors without relying solely on supervised fine-tuning (SFT) (Jaech et al. 2024; OpenAI, 2025; Guo et al. 2025a; Team et al. 2025; Team, 2025). Notably, DeepSeek-R1-Zero (Guo et al. 2025a) has demonstrated that purely RL-driven training can unlock self-verification and iterative refinement capabilities, setting new benchmarks for reasoning-intensive applications.

Despite these advancements, the application of reinforcement learning for reasoning in context-grounded claim verification remains underexplored. To address this gap, we introduce R1-ClaimVerifier (R1-CV), a three-stage framework designed to optimize claim verification through reinforcement learning, following the reasoning-driven paradigm of DeepSeek-R1.

First, in the training data construction phase, we employ data distillation based on Qwen2.5-32B-Instruct, integrating human-reasoning-inspired CoT templates and a sampling-based data filtering mechanism to curate a high-quality reasoning dataset for claim verification. Second, In the CoT-SFT model training stage, we fine-tune a base LLM, Qwen2.5-7B-Instruct, using the distilled reasoning dataset to establish a strong reasoning foundation. Finally, in the reinforcement learning (RL) stage, we leverage filtered data samples to further refine and enhance the model's reasoning capabilities.

Extensive evaluations across diverse claim verification tasks demonstrate that R1-CV outperforms distilled R1 models of the same scale while maintaining significantly lower inference costs.

Our contributions are as follows:

- To the best of our knowledge, we are the first to develop efficient small reasoning

LLMs using reinforcement learning with verifiable rewards (Guo et al. 2025a) for the context-grounded claim verification tasks.

- We show that using reinforcement learning with verifiable rewards can achieve better performance on multiple context-grounded claim verification tasks compared to the out-of-box reasoning LLMs in the same size.

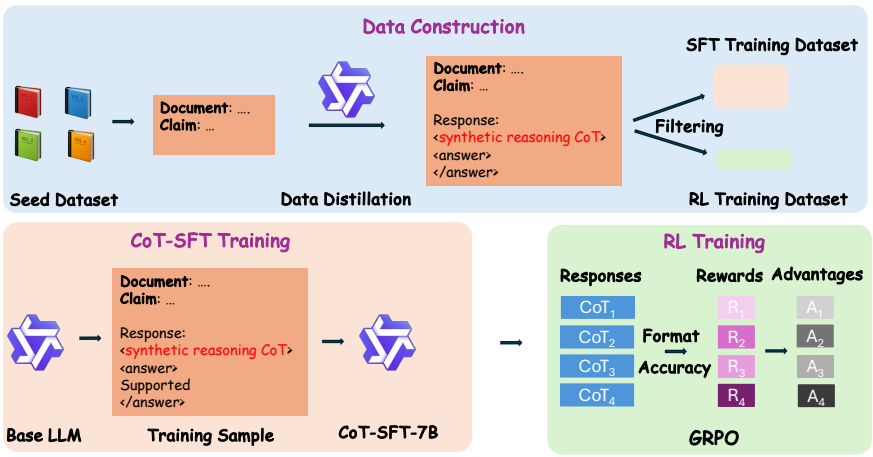

Figure 4.2: The pipeline of our R1-CV framework. In the training data construction stage, we first collect a set of diverse claim verification seed dataset. Then we leverage a SOTA LLM to do data distillation to obtain synthetic reasoning CoTs. After that, we apply a data filtering mechanism to get CoT-SFT training set and RL training set. In the CoT-SFT model training stage, we train a Qwen2.5-7B-Instruct model using the distilled dataset. And finally, we further perform RL training with the fine-tuned CoT-SFT-7B model.

§ 4.1 Task Formulation

In this work, we focus on the task of reasoning-oriented *context-grounded claim verification*, where the goal is to determine the veracity of a given claim based on an associated context document. The model is required to output a valid reasoning chain

before the veracity label. Formally, let c denote an input claim and let d represent a context document (or set of documents), consisting of unstructured text and structured data (e.g., tables). The task is to predict a veracity label y from the label set $\mathcal{Y} = \{\text{SUPPORTEDT}, \text{UNSUPPORTED}\}$, and to generate a natural language reasoning chain e that justifies the decision.

Our approach defines the verification process as finding:

$$(e^*, y^*) = \arg\max_{y \in \mathcal{Y}} P(e, y \mid c, d), \tag{4.1.1}$$

where $P(e, y \mid c, d)$ is modeled using a generative LLM. To encourage explicit and interpretable reasoning, the LLM is prompted to decompose the claim into intermediate sub-claims or reasoning steps (e.g., via chain-of-thought prompting) before arriving at the final veracity decision.

The overall process can be broken down into two key stages: **Context Processing and Evidence Extraction.** Given the input claim c, the system first explicitly identifies and extracts claim-relevant passages from the context d, which may consist of unstructured text and structured data (e.g., tables). This step is crucial to mitigate the noise inherent in lengthy documents. **Verification and Reasoning Generation.** Using the extracted evidence, the LLM may conducts a series of complex reasoning process, such as performing numerical calculations, applying external professional knowledge, and conducting comparative analyses to verify the claim. In this stage, the model outputs both a valid reasoning chain e^* mimicing its cognitive reasoning process and the final label y^*, thereby improving transparency and facilitating downstream validation.

Our Goal. Our objective is to develop a reasoning model that is both **effective** and **efficient** for the task of context-grounded claim verification. However, achieving this objective is hindered by two key **challenges**. First, annotating claim verification datasets with explicit reasoning steps is inherently complex and labor-intensive for human annotators, especially in cases requiring multi-step reasoning and diverse inferential operations. Second, scaling the annotation of authentic reasoning processes across diverse domains poses significant difficulties, leaving open the question of how models can adapt and self-evolve when encountering novel scenarios.

§ 4.2 Methodology

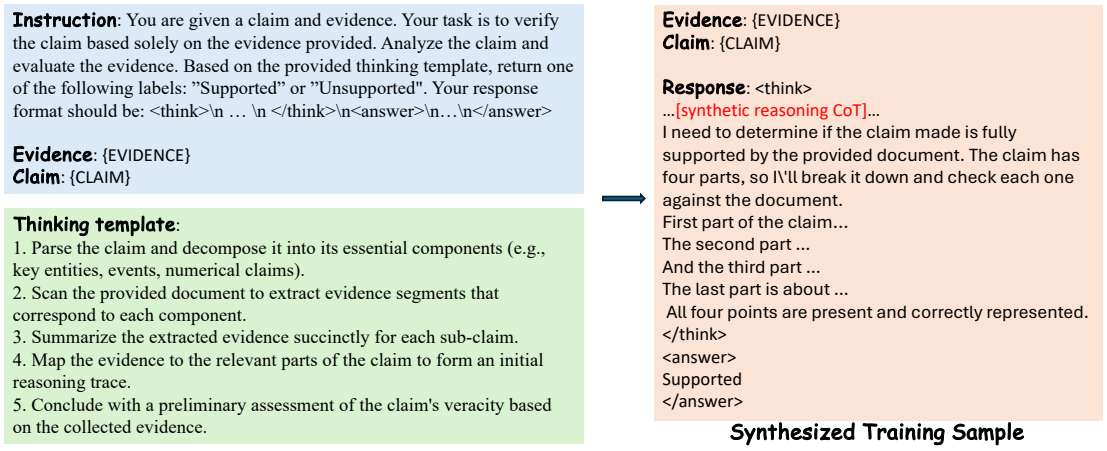

Figure 4.3: The pipeline for CoT data synthesis. For each (document, claim) pair, we prompt the model with documents, claims, and structured CoT reasoning instruction template to synthesize a reasoning trajectory.

To address these challenges, drawing inspiration from DeepSeek R1 (Guo et al. 2025a), we propose a two-stage training framework for claim verification designed to improve reasoning capabilities while enhancing training stability during reinforcement

learning.

4.2.1 Overview

As illustrated in Figure C.7, our approach consists of three key stages. In the training data construction phase, we employ data distillation based on Qwen2.5-32B-Instruct, integrating human-reasoning-inspired CoT templates and a sampling-based data filtering mechanism to curate a high-quality reasoning dataset for claim verification. In the subsequent CoT-SFT model training stage, we fine-tune the base language model, Qwen2.5-7B-Instruct, using the distilled dataset to establish a strong reasoning foundation. Finally, in the reinforcement learning (RL) stage, we leverage filtered data samples to further refine and enhance the model's reasoning capabilities.

4.2.2 Training Data Construction

In this stage, our goal is to construct a high-quality supervised fine-tuning dataset that fosters explicit reasoning for context-grounded claim verification. To this end, we design a systematic and rigorous data construction pipeline that integrates data distillation and filtering techniques to ensure both quality and diversity. The full pipeline is illustrated in Figure 4.3.

Data Source

We begin by constructing a source seed dataset to establish a foundational basis for general claim verification across diverse domains, reasoning tasks, and input modalities. Our seed dataset comprises about 30k (document, claim, label) pairs, sampled

from a broad spectrum of real-world open-source claim verification datasets. This selection ensures comprehensive coverage in terms of input document length distribution, domain knowledge, and reasoning types. Specifically, we randomly sample 10,000 training instances from DocNLI (Yin et al., 2021), a document-level natural language inference dataset spanning Wikipedia and news domains. To incorporate non-textual reasoning, we include 10,000 samples from TabFact (Chen et al., 2020), a table-based fact verification dataset within the Wikipedia domain. Additionally, we enhance domain diversity by incorporating training samples from PubHealth (Kotonya and Toni, 2020b) in the healthcare domain and SciFact (Wadden et al., 2020b) in the scientific domain.

Reasoning Chain-of-Thought Data Distillation

After constructing the preprocessed seed dataset, the complexity of real-world claim verification tasks makes it impractical for human annotators to manually generate explicit reasoning steps at scale, as this process is both labor-intensive and time-consuming. To address this challenge, we leverage the Qwen2.5-32B-Instruct model to generate intermediate reasoning chains in the form of Chain-of-Thought (CoT) rationales. As illustrated in Figure 4.3, we prompt the model with documents, claims, and structured CoT reasoning instruction template to synthesize reasoning trajectories that integrate domain knowledge with human-like inferential processes. Specifically, the model decomposes each claim into its fundamental components, extracts key supporting evidence from the provided document, and systematically performs stepwise verification. For each instance, the model generates a structured CoT rationale, followed by a veracity label used for downstream data filtering. This approach yields a refined dataset in which each claim-document pair is enriched with an explicit reasoning path, mak-

ing the decision-making process more interpretable and aligned with human cognitive reasoning.

Data Filtering

Given that the distilled labels may not always align perfectly with the ground truth, we begin by filtering out samples where the distilled label is incorrect. This results in the CoT-SFT subset with the size of 20.3k, which is subsequently used in the supervised fine-tuning stage. We classify the discarded samples as "hard" instances, which are then utilized in the reinforcement learning (RL) stage to facilitate the model's self-evolution across various reasoning scenarios. To further enhance training efficiency, we perform an additional round of strategic data filtering. Specifically, we conduct three sampling passes on the hard subset with a higher sampling temperature, filtering out samples in which all elements are entirely incorrect. Such samples are considered too challenging to contribute meaningfully to performance improvement in the RL stage.This strategy ensures that the model prioritizes learnable instances that provide meaningful optimization signals, thereby enhancing training effectiveness and reducing exposure to uninformative extremes. The filtered subset, with the size of 5k, is then employed in the RL training stage to refine the model's reasoning capabilities.

4.2.3 Chain-of-Thought Supervised Fine-Tuning

The first training stage is designed to equip models with human-reasoning-inspired, advanced claim verification skills using synthesized chain-of-thought (CoT) data. We fine-tune a base LLM on our enriched dataset, teaching it to replicate the reasoning demonstrated in the provided CoT examples. This supervised phase trains the model

on both the verification task and the explicit reasoning steps, helping it adopt strategies that align with human thought processes.

4.2.4 Reinforcement Learning with Verifiable Rewards

After the initial CoT-SFT training, we applied reinforcement learning on a separate training subset to further improve the model's complex verification capabilities, building on its document-grounded claim verification foundation. As shown in Figure C.7, we use the Group Relative Policy Optimization algorithm (GRPO) (Shao et al. 2024), a method that optimizes the policy model by sampling multiple outputs for each question and computing a group-relative advantage rather than using a critic model. To tailor GRPO for document-grounded claim verification tasks, we developed a rule-based reward system with the trained CoT-SFT model.

Training Objective. Let q represent the concatenation of the system prompt, input documents, and claim, and let $\{o_1, \ldots, o_G\}$ denote the outputs sampled from the old policy model $\pi_{\theta_{old}}$. Then, the optimization objective for the policy model π_θ is formally defined as:

$$\begin{aligned} \mathcal{J}_{GRPO}(\theta) &= \mathbb{E}\left[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O \mid q)\right] \\ &\frac{1}{G}\sum_{i=1}^{G}\left(\frac{\pi_\theta\left(o_i \mid q\right)}{\pi_{\theta_{old}}\left(o_i \mid q\right)} A_i - \beta \mathbb{D}_{KL}\left(\pi_\theta || \pi_{ref}\right)\right) \end{aligned} \tag{4.2.1}$$

$$\mathbb{D}_{KL}\left(\pi_\theta || \pi_{ref}\right) = \frac{\pi_{ref}\left(o_i \mid q\right)}{\pi_\theta\left(o_i \mid q\right)} - \log \frac{\pi_{ref}\left(o_i \mid q\right)}{\pi_\theta\left(o_i \mid q\right)} - 1 \tag{4.2.2}$$

where the KL divergence term serves as a regularization mechanism for the policy update, constraining π_θ to remain sufficiently aligned with the reference model π_{ref}. This prevents excessive deviation, thereby maintaining stability during training. $\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)}$ is the policy ratio and A_i is the advantage computed based on a group of rewards $\{r_1, \ldots, r_G\}$ corresponding to the outputs of each group:

$$A_i = \frac{r_i - mean(\{r_1, \ldots, r_G\})}{std(\{r_1, \ldots, r_G\})} \tag{4.2.3}$$

Reward Modeling. We adopted a rule-based reward system specifically tailored for the document-grounded claim verification tasks, comprising two types of rewards: format reward and accuracy reward . Our RL framework uses two reward components:

Format Reward. Ensures that the model's output adheres to a predefined structure (e.g., the reasoning trace is enclosed within designated markers).

$$Reward_{format} = \begin{cases} 1, & \text{if format is correct} \\ -1, & \text{if format is incorrect} \end{cases} \tag{4.2.4}$$

Accuracy Reward. Evaluates the accuracy of the final veracity prediction by comparing it against reference labels.

$$Reward_{accuracy} = \begin{cases} 1, & \text{if the answer fully matches the ground truth} \\ -0.5, & \text{if the answer semantically matches the ground truth} \\ -1, & \text{if the answer cannot be parsed or is wrong} \end{cases} \tag{4.2.5}$$

§ 4.3 Experiment

4.3.1 Implementation Details

Evaluation settings. We conduct experiments on various context-grounded claim verifications, including text and table-based documents. These tasks include some seen tasks: DocNLI (Yin et al., 2021), PubHealth (Kotonya and Toni, 2020b), and SciFact (Wadden et al., 2020b). Some unseen tasks are also included. FV-IE, FV-Math and FV-Know are obtained from (Zhao et al., 2024b). We also use SciTab (Lu et al., 2023) and WiCE (Kamoi et al., 2023b) tasks. We use the accuracy as the evaluation metric. Please refer to the Appendix B.2 for more details of these evaluation benchmarks.

Baseline Models. To comprehensively evaluate the reasoning capabilities of R1-CV in claim verification tasks, we conduct a thorough comparative assessment against multiple

state-of-the-art reasoning LLM and general LLMs. We use a series of distilled models from DeepSeek R1 (Guo et al. 2025a), including DeepSeek-R1-Distill-Qwen-1.5B (R1-1.5B), DeepSeek-R1-Distill-Qwen-7B (R1-7B), DeepSeek-R1-Distill-Llama-8B (R1-8B), DeepSeek-R1-Distill-Qwen-14B (R1-14B), DeepSeek-R1-Distill-Qwen-32B (R1-32B), DeepSeek-R1-Distill-Llama-70B (R1-70B) and QwQ-32B (Team, 2025). We also compare to a series of Qwen2.5 instructed models, including Qwen2.5-1.5B-Instruct (Qwen2.5-1.5B),Qwen2.5-3B-Instruct (Qwen2.5-3B), Qwen2.5-7B-Instruct (Qwen2.5-7B), Qwen2.5-14B-Instruct (Qwen2.5-14B) and Qwen2.5-32B-Instruct (Qwen2.5-32B). We also compare our method to the existing state-of-the-art prompting-based methods, which rely on the Decompose-then-Verify paradigm (Wei et al. 2024; Wang et al., 2024). Thus, we adapted this method to our evaluation setting and compared it to our model.

Training Details. We use Adam optimizer (Kingma and Ba, 2015) with learning rate as $1e-5$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1e-8$ in the CoT-SFT stage and AdamW optimizer (Loshchilov and Hutter, 2019a) with learning rate as $1e-6$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1e-6$ in the RL stage. The accumulated batch size is set to 16 in the CoT-SFT stage. For the RL stage, the accumulated batch size is set to 12. The RL stage employs a maximum completion length of 1024 tokens with 3 samples per input.

4.3.2 Main Results

Accuracy. Table 4.1 presents the performance of various language models across multiple fact verification datasets. The models are evaluated on eight distinct benchmarks, including DocNLI, FindVER-IE, FindVER-Math, FindVER-Know, PubHealth,

Model	DocNLI	PubHealth	SciFact	FV-IE	FV-Math	FV-Know	SciTab	WiCE	AVG
QwQ-32B	0.77	0.66	0.87	0.90	0.81	0.80	0.77	0.77	0.79
R1-70B	0.78	0.69	0.84	0.92	0.77	0.82	0.75	0.77	0.79
R1-32B	0.80	0.65	0.85	0.91	0.80	0.80	0.75	0.80	0.80
R1-14B	0.75	0.65	0.85	0.88	0.74	0.74	0.74	0.80	0.77
R1-8B	0.71	0.66	0.81	0.87	0.68	0.73	0.68	0.67	0.73
R1-7B	0.63	0.61	0.75	0.76	0.63	0.71	0.69	0.61	0.68
R1-1.5B	0.56	0.59	0.66	0.65	0.61	0.59	0.59	0.53	0.60
Qwen2.5-32B	0.76	0.67	0.86	0.94	0.71	0.82	0.77	0.81	0.79
Qwen2.5-14B	0.74	0.66	0.85	0.92	0.71	0.84	0.74	0.77	0.78
Qwen2.5-7B	0.70	0.69	0.86	0.80	0.63	0.82	0.69	0.73	0.74
Qwen2.5-3B	0.68	0.78	0.84	0.72	0.66	0.74	0.61	0.67	0.71
Qwen2.5-1.5B	0.60	0.64	0.80	0.73	0.56	0.62	0.67	0.62	0.66
Decompose-then-Verify	0.7	0.63	0.89	0.88	0.7	0.82	0.69	0.7	0.75
Qwen2.5-7B (SFT)	0.84	0.84	0.88	0.82	0.62	0.8	0.68	0.68	0.77
R1-CV (7B)	0.87	0.87	0.9	0.85	0.72	0.82	0.69	0.73	0.80

Table 4.1: Performance of various models on different fact verification datasets.

SciFact, SciTab, and WiCE. The final column reports the average performance across all datasets. Among the models evaluated, R1-CV achieves the highest average score of 0.80, demonstrating superior generalization across diverse fact verification tasks. Notably, Qwen2.5-7B(SFT) also performs competitively, particularly excelling in DocNLI and PubHealth, with scores of 0.84 and 0.84, respectively. The QwQ-32B model achieves an average score of 0.79, closely competing with the DeepSeek-R1-Distill models, such as R1-70B and R1-32B, which obtain average scores of 0.79 and 0.80, respectively. These results highlight that while larger models tend to perform better overall, model architecture and training methodologies play a crucial role in determining performance. Our model can also outperform the existing state-of-the-art prompting-based method Decompose-then-Verify, which shows that our training pipeline improves model's reasoning capability.

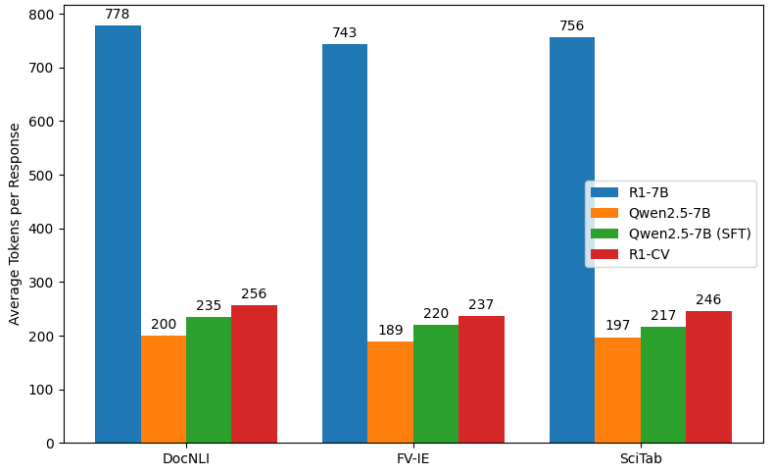

Figure 4.4: For each task, we compute the the average response tokens per response of each model.

Reasoning Efficiency. In addition to improvements in accuracy, we also assessed the average response tokens generated by each 7B model across various tasks. As shown in Table 4.4, our R1-CV model, which is trained for reasoning tasks, achieves a comparable token consumption to the CoT prompting-based model, Qwen2.5-7B, across several tasks. However, it significantly outperforms the state-of-the-art reasoning model, R1-7B, in terms of token efficiency. Specifically, the R1-7B model generates an average of 778 tokens per sample on the DocNLI task, which is more than three times the token consumption of our R1-CV model, which generates only 256 tokens. This demonstrates that R1-CV generates more concise reasoning CoTs, a crucial advantage for tasks involving long inputs, as it reduces user inference costs. Overall, we found that our method not only improves performance but also uses fewer tokens. In contrast, the R1-7B model exhibits a tendency to overthink (Chen et al. 2024) during claim verification tasks, sometimes continuing generation until it hits the maximum token limit. This over-generation is one of the key reasons for its significantly higher average

Model Variant	AVG Score	Relative Drop
R1-CV (Full)	0.80	—
w/o RL Fine-tuning	0.78	-2.2%
w/o CoT Templates (SFT)	0.76	-3.2%
w/o Both RL and CoT	0.74	-6.0%

Table 4.2: Ablation study results. Removing the RL module, the CoT templates, or the reward shaping mechanism degrades performance, demonstrating that each component plays a significant role in enhancing both accuracy and reasoning transparency.

token count compared to other models.

§ 4.4 Ablation Study

To better understand the contribution of each component in our R1-CV framework, we conduct a series of ablation experiments. In these experiments, we selectively remove or modify key modules, including the human-reasoning-inspired CoT templates, the self-evolving reinforcement learning (RL) stage, and the rule-based reward shaping, while keeping all other components unchanged. Table 4.2 summarizes the average performance (AVG score across all benchmarks) of the full model against various ablated variants.

Effect of RL Fine-tuning. Excluding the RL component leads to a notable drop (approximately 2.2%) in the average score. This indicates that the RL phase, which promotes the autonomous evolution of reasoning paths, significantly improves the model's robustness, particularly in ambiguous or low-resource verification scenarios.

Role of CoT Templates. When the system is trained solely with supervised fine-tuning (SFT) without the incorporation of the human-reasoning-inspired CoT template, the performance drops by about 3.2%. This suggests that the diversity provided by the elaborate CoT template is critical for capturing the nuanced reasoning patterns required for accurate claim verification.

Combined Ablations. The largest drop (6.0%) is observed when both RL fine-tuning and CoT guidance are removed, affirming that these components complement each other and are jointly crucial for the overall performance of R1-CV.

§ 4.5 Case Study

To better understand the differences in reasoning behavior between R1-7B and R1-CV, we present a case study from the FDV-Math task (Zhao et al., 2024b) in Figure 4.5. This analysis provides insights into how each model approaches complex claim verification in financial math scenario.

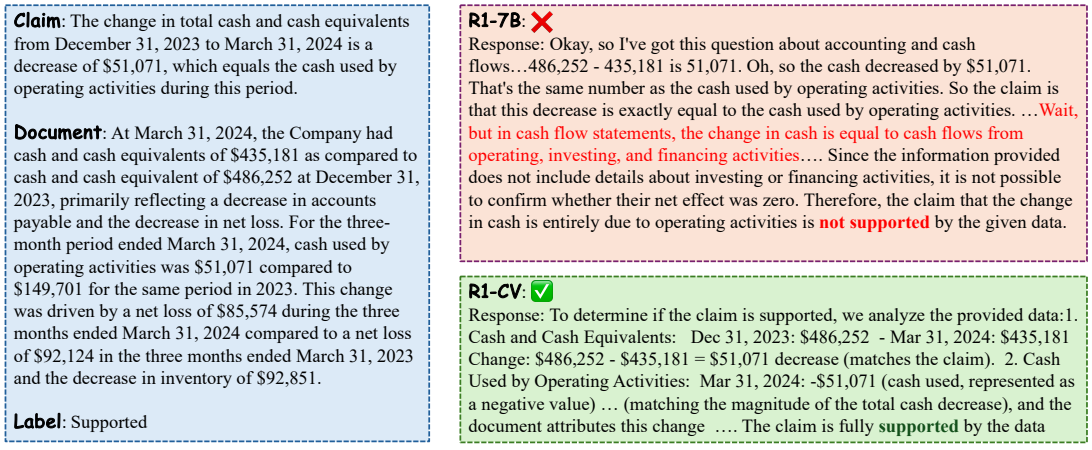

Figure 4.5: Case study.

Examining the response from R1-7B, we observe that the model is initially capable of extracting the relevant information from the provided document, such as identifying the necessary case equivalents to assess the claim. Additionally, its numerical calculations are correct. However, a critical issue arises when the model experiences an "Aha moment," indicated by the phrase "Wait." At this point, R1-7B asserts that the analysis must also account for investing and financing activities based on cash flow statements. Notably, neither the provided document nor the claim mentions cash flow statements, this information originates from the model's internal knowledge. Since the task requires verification strictly based on the given document, this behavior exemplifies the model's overthinking issue, where it introduces external knowledge instead of adhering to the evidence presented.

In contrast, R1-CV effectively extracts the necessary information and correctly verifies the two-step claim without relying on internal knowledge. This demonstrates its ability to maintain a more disciplined reasoning process, ensuring that verification remains grounded in the provided evidence rather than extraneous inference.

§ 4.6 Conclusion

In this work, we introduced R1-ClaimVerifier (R1-CV), a reasoning-enhanced reinforcement learning (RL) framework for context-grounded claim verification. Unlike conventional approaches that rely heavily on prompt engineering or human-annotated explanations, R1-CV enables scalable and adaptive reasoning by leveraging a three-stage training paradigm: (1) constructing a high-quality Chain-of-Thought (CoT) dataset through distillation and filtering, (2) supervised fine-tuning to establish a strong rea-

soning foundation, and (3) reinforcement learning with verifiable rewards to refine the model's verification capabilities with hard samples. Experimental results across multiple fact verification benchmarks demonstrate that R1-CV outperforms existing reasoning models of the same scale while maintaining significantly lower inference costs.

§ 4.7 Limitations

First, although our pipeline effectively synthesizes high-quality reasoning chains, the reliance on teacher models in the first step introduces potential bias. Any systematic errors or hallucinations from the teacher LLM may propagate to the distilled dataset, thereby influencing the downstream fine-tuning and reinforcement learning stages. Second, our evaluation primarily focuses on English-language datasets across a limited set of domains (e.g., scientific, health, financial). The generalizability of R1-CV in multilingual or low-resource settings remains unexplored and represents an important direction for future research.

Chapter 5

A Systematic Study of Long Chain-of-Thought Transfer in Mathematical Tasks

While Chapter 3 and Chapter 4 explore how synthetic data and efficient reasoning frameworks can enhance LLM performance on complex reasoning tasks, these improvements often rely on large-scale or resource-intensive models. In practical settings, however, deploying such models can be computationally prohibitive. This brings forth a new challenge: how can we effectively transfer complex reasoning abilities from larger, more powerful models to smaller ones without compromising performance? It motivates the need for scalable techniques that transfer reasoning capabilities to smaller, more efficient models without significant performance degradation. In Chapter 5, we investigate this challenge through the lens of *reasoning distillation*. Rather than treating distillation as a black-box compression technique, we explore how factors such as teacher–student architectural compatibility and reasoning style alignment influence the transferability of reasoning skills. This final line of inquiry complements our earlier work by extending reasoning improvements beyond standalone model training to cross-model knowledge transfer.

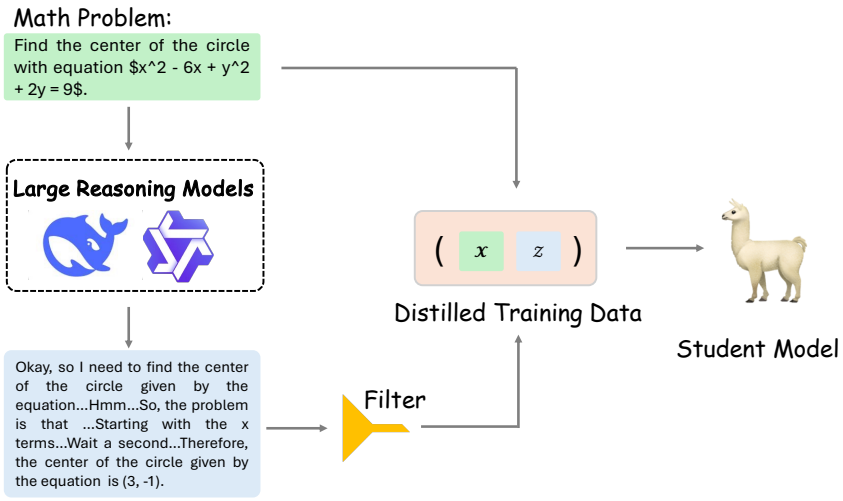

Figure 5.1: In this work, we uncover the distillation patterns when fine-tuning various sizes of non-reasoning student models on reasoning data distilled from different large reasoning teacher models (e.g., DeepSeek-R1 (Guo et al. 2025a) and QwQ-32B (Team, 2025)).

Recent advances in reasoning-oriented large language models (LLMs), or large reasoning models (LRMs) (Jaech et al. 2024; OpenAI, 2025; Guo et al. 2025a; Team, 2025) have demonstrated remarkable capabilities in complex reasoning tasks, including mathematical reasoning (Hendrycks et al., 2021b), code generation (Jain et al., 2025) and web agentic tasks (Yao et al., 2022; Zhou et al., 2024). LRMs like DeepSeek-R1 generate a long chain-of-thoughts (CoT) before outputting the final answer, implement elaborate self-reflection, verification and self-correction mechanisms that mimic human problem-solving processes (Guo et al. 2025a). However, deploying these LRMs with advanced reasoning capabilities in real-world settings often demands substantial computational resources, posing significant challenges for deployment in resource-constrained environments (Sui et al. 2025). This motivates the development of smaller, more efficient reasoning models.

Knowledge distillation (Hinton et al., 2015), the process of transferring capabilities from large "teacher" models to more compact "student" models, offers a promising approach to address this challenge. While distillation has proven effective for transferring general language capabilities (West et al., 2022; Gu et al., 2024; Muralidharan et al., 2024), the distillation of complex long CoT reasoning skills presents unique challenges (Li et al., 2023; Wang et al., 2023; Chen et al. 2025). Despite the growing importance of efficient reasoning models via knowledge distillation from general LLMs (Shridhar et al., 2023; Chen et al., 2023; Chenglin et al., 2024b), several critical questions remain unexplored when it comes to LRMs: How does the architecture and scale of student models affect their ability to acquire long CoT reasoning capabilities? Does teacher model scale consistently correlate with better distillation outcomes? Can domain-specific long CoT reasoning skills (e.g., mathematical reasoning) transfer to broader reasoning tasks? How sensitive is long CoT reasoning distillation to the quality of training long CoTs? Understanding these dynamics is essential for developing efficient and effective reasoning models that maintain the capabilities of their larger counterparts while requiring significantly fewer computational resources.

In this work, we present, to the best of our knowledge, the first systematic investigation of knowledge distillation for long CoT reasoning capabilities, focusing specifically on complex mathematical reasoning as an exemplar domain that demands structured, multi-step problem-solving. Through comprehensive experiments across diverse long CoT reasoning teacher models and two families of non-reasoning student models with various sizes and experimental conditions, we make the following contributions.

First, we demonstrate that student model scale significantly impacts distillation efficacy, with larger models showing enhanced capacity to absorb complex reasoning

patterns, though with notable architecture-specific variations (5.3). Then we reveal that teacher model selection critically influences distillation outcomes, with larger teachers not always producing better students, revealing a non-monotonic relationship between teacher scale and distillation effectiveness (§5.3). We also show that cross-domain generalization of distilled reasoning follows different patterns than in-domain performance, with mid-sized teacher models often facilitating better transfer across domains than their larger counterparts (§5.4). Furthermore, we establish that the quality of reasoning demonstrations in training data is more important than quantity (Ye et al., 2025), with clean, correct examples consistently outperforming larger but mixed-quality datasets (§5.5). Last, we identify significant architecture compatibility effects, where certain teacher-student pairings show unexpectedly strong or weak performance, suggesting the importance of architectural alignment in reasoning distillation (§5.6).

Our findings challenge the conventional assumption in reasoning LLM knowledge distillation that larger reasoning teacher models should always be used to fine-tune downstream reasoning models, regardless of the student model's size or architecture, even though these factors can significantly influence downstream performance, such as when training short CoT student models (Magister et al., 2023; Ho et al., 2023) or fine-tuning non-reasoning student models (Phogat et al., 2024). We provide actionable insights for developing more capable yet computationally efficient reasoning models. By understanding the factors that influence successful reasoning transfer, this work establishes a foundation for democratizing access to advanced reasoning capabilities through smaller, more deployable models.

§ 5.1 Preliminaries

Formal Definition. In this work, we uncover the patterns of fine-tuning non-reasoning LLMs on reasoning data distilled from a series of LRMs that are able to generate long chain-of-thoughts. Generally, LRMs can be formally characterized as parameterized functions that map input sequences to output sequences through a series of reasoning steps. Given an input query $x \in \mathcal{X}$, an LRM with parameters θ generates a long chain-of-thought, i.e., reasoning trace z before producing the final output $y \in \mathcal{Y}$. Formally, we define this process as: $p_\theta(y|x) = \sum_{z \in \mathcal{Z}} p_\theta(y|z,x) \cdot p_\theta(z|x)$, where $p_\theta(z|x)$ represents the probability of generating the long reasoning CoT z given input x. $p_\theta(y|z,x)$ denotes the probability of producing output y given both the input x and long reasoning CoT z. $\mathcal{Z}$ is the space of all possible reasoning CoTs. Long reasoning CoT z can be decomposed into a sequence of intermediate reasoning steps $z = (z_1, z_2, ..., z_T)$, where each step z_t often includes self-reflection, verification, and self-correction, mimicking humans when solving complex problems.

Reasoning Distillation via Supervised Fine-Tuning. Supervised Fine-Tuning (SFT) is a dominant paradigm for distilling reasoning capabilities from large teacher LLMs into smaller student models. In this approach, a teacher model, typically a computationally expensive LLM, generates intermediate reasoning steps and final answers for a set of problem instances. These demonstrations form a dataset of (input, reasoning, output) triples used to train a more compact student model. The student model is trained to mimic the teacher's reasoning through standard maximum likelihood estimation: $\theta_S = \arg\max_\theta \sum_{(\mathbf{x},\mathbf{z},\mathbf{y}) \in \mathcal{D}_{\text{SFT}}} \log P\big(\mathbf{z}, \mathbf{y} \mid \mathbf{x}; \theta\big)$, where x_i is the input prompt, z represents the reasoning steps generated by the teacher LRM, y is the final answer, and θ denotes the

parameters of the non-reasoning student model. This approach enables smaller models to acquire complex reasoning abilities while requiring significantly fewer parameters than their teachers.

§ 5.2 Experimental Settings

In this section, we introduce our major experimental settings, including the chosen training dataset, evaluation benchmarks and some training details.

Long CoT Reasoning Teacher Models. In this work, we use a series of DeepSeek-R1 (Guo et al. 2025a) distilled reasoning models: DeepSeek-R1-Distill-Qwen-1.5B (abbreviated as **R1-1.5B** for simplicity), DeepSeek-R1-Distill-Qwen-7B (**R1-7B**), DeepSeek-R1-Distill-Llama-8B (**R1-8B**), DeepSeek-R1-Distill-Qwen-14B (**R1-14B**), DeepSeek-R1-Distill-Qwen-32B (**R1-32B**), DeepSeek-R1-Distill-Llama-70B (**R1-70B**) and QwQ-32B (Team, 2025).

Non-Reasoning Student Models. We use two foundation model families: Llama (Grattafiori et al. 2024) and Qwen (Qwen et al., 2025a). Specifically, we use Llama-3.2-1b-Instruct, Llama-3.2-3B-Instruct, Llama-2-7B-Chat, Llama-3.1-8B-Instruct, and instruction-tuned Qwen2.5 models from 0.5B to 7B.

Training Dataset. Following the prior work (Liu et al., 2025; Zeng et al., 2025; Ye et al., 2025), we use `MATH` dataset (Hendrycks et al., 2021b) as our training dataset. We chose mathematical reasoning as the focus of our investigation because it serves as a prototypical domain for structured, multi-step problem-solving, making it an ideal

testbed for evaluating the effectiveness of long CoT knowledge distillation. Unlike tasks that may be solved heuristically or with shallow pattern matching, mathematical reasoning demands precise logical consistency and interpretability in intermediate reasoning steps. This allows us to easily assess how distilled models internalize complex reasoning trajectories from large teacher models.

We prompt each reasoning teacher model using the training dataset to generate corresponding CoT reasoning steps and final answers. The prompt is shown in the Appendix C.1. To ensure data quality, we retain only those samples where all teacher models converge on the correct answers, resulting in a refined training set comprising 5,009 samples. Please refer to the Appendix C.3 to see the distilled example outputs.

Evaluation Benchmarks. Following the prior work (Guo et al. 2025b; Ye et al., 2025; Muennighoff et al., 2025), we conduct a comprehensive evaluation on a wide-range of reasoning tasks with both in-distribution and out-of-distribution settings. **In-Domain Setting:** These include several mathematical reasoning benchmarks: the American Mathematics Competitions (AMC23), GSM8K (Cobbe et al. 2021) and MATH-500 (Hendrycks et al., 2021c) datasets. **Out-of-Domain Setting:** To evaluate broader generalization capabilities beyond the math domain, we also evaluate the trained models on GPQA (Rein et al., 2024), ARC (Clark et al., 2018), TruthfulQA (Lin et al., 2022) and HellaSwag (Zellers et al., 2019). These benchmarks assess diverse reasoning skills and cognitive capabilities, illuminating the model's ability to extend its mathematical reasoning to broader areas. Please refer to the Appendix C.4 for more details of these evaluation benchmarks.

Evaluation Configurations. We use one of the popular LLM evaluation frameworks, Lighteval * released from Hugging Face **, to conduct all evaluations. Without specifications, we adopt the default settings from Lighteval. We use the **Accuracy** as the evaluation metric for all benchmarks except for TruthfulQA, where we use **MC1 (Single-true)** as the evaluaiton metric.

Implementation Details For each non-reasoning student model, we fine-tune it on a training set distilled from a teacher model for 2 epochs on NVIDIA A100 GPUs. In particular, we use a learning rate of 1e-5 with the AdamW optimizer (Loshchilov and Hutter, 2019b). We train in bfloat16 precision with a batch size of 2, the gradient accumulation step of 4. The maximum sequence length is set to 8192 during the training. We also set the maximum generation length as 8192 during the evaluation. For Llama-2-7B-Chat, we set 4096 for both training and generation due to the context window limit.

§ 5.3 Cross-Model Long Reasoning CoT Distillation Patterns in Mathematical Reasoning

Figure 5.2 presents the in-distribution performance of various models fine-tuned on reasoning datasets distilled from a series of teacher reasoning models (R1-1.5B through R1-70B). Our analysis reveals several significant patterns in knowledge transfer across model architectures and scales.

*https://huggingface.co/docs/lighteval/en/index
**https://huggingface.co

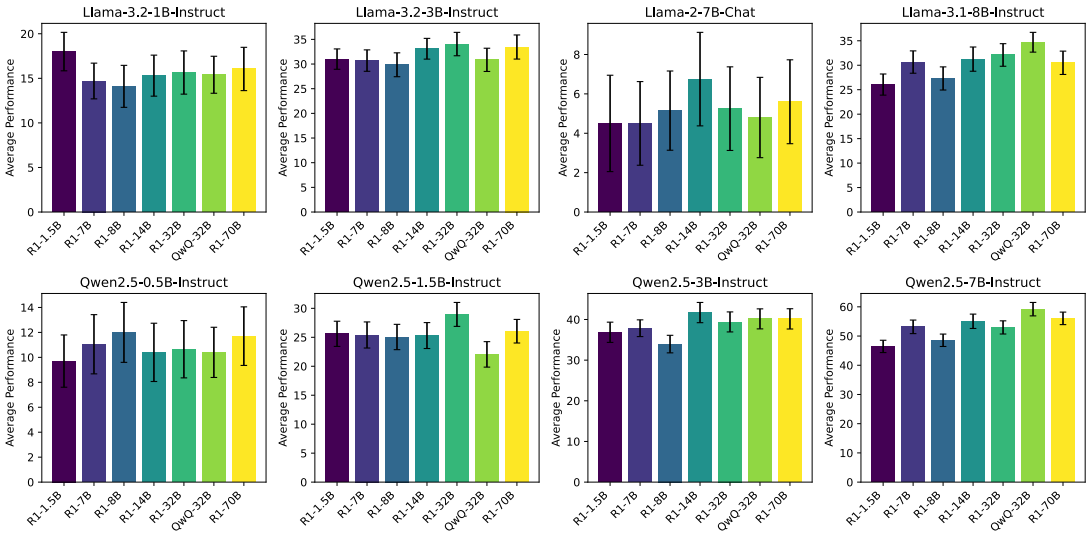

Figure 5.2: Average performance in the in-distribution setting. In each subplot, the x-axis represents the seven reasoning teacher models used in this study. For simplicity, we refer to DeepSeek-R1-Distill-Qwen-1.5B as R1-1.5B, and adopt similar abbreviations for other models as described in Section5.2. The top label (e.g., Llama-3.2-1B-Instruct) indicates the non-reasoning student model. We fine-tune each student model on reasoning training data distilled from a specific reasoning teacher model, and evaluate the fine-tuned model on in-distribution benchmarks to report the average performance.

5.3.1 Effect of Student Model Scale on Distillation Efficacy

We find a clear correlation between student model size and distillation performance. As student model capacity increases from Llama-3.2-1B-Instruct (18-26% average performance) to Qwen2.5-7B-Instruct (46-72% average performance), we observe a consistent improvement in the ability to acquire mathematical reasoning capabilities. This suggests that larger student models possess greater capacity to absorb complex reasoning patterns from teacher-generated data.

Notably, the Qwen family exhibits superior distillation receptivity compared to similarly-sized Llama counterparts. For instance, Qwen2.5-3B-Instruct consistently

outperforms Llama-3.2-3B-Instruct across all teacher models, indicating architecture-specific factors may influence distillation effectiveness beyond mere parameter count.

5.3.2 Teacher Model Impact on Knowledge Transfer

We note that the size of the teacher reasoning model does not always correlate with improved student performance. While there is a moderate upward trend when moving from R1-1.5B to larger teachers for most student models, this relationship is non-monotonic. For example, R1-14B and QwQ-32B teachers often produce better student performance than the larger R1-70B teacher.

This suggests that the quality and structure of the generated reasoning, rather than simply the teacher's scale, significantly impacts distillation outcomes. The superior performance of QwQ-32B as a teacher for several students indicates that specialized training or architectural differences in teacher models may be more important than raw parameter count.

5.3.3 Architecture Compatibility in Knowledge Transfer

The distillation patterns reveal potential compatibility effects between teacher and student architectures. The unexpectedly poor performance of Llama-2-7B-Chat (4-9% average) despite its size suggests that some models may be less receptive to reasoning knowledge transfer, possibly due to their pre-training objectives or architectural constraints.

These results highlight the complex interplay between model architectures, scales, and training objectives in mathematical reasoning distillation. The results demonstrate that

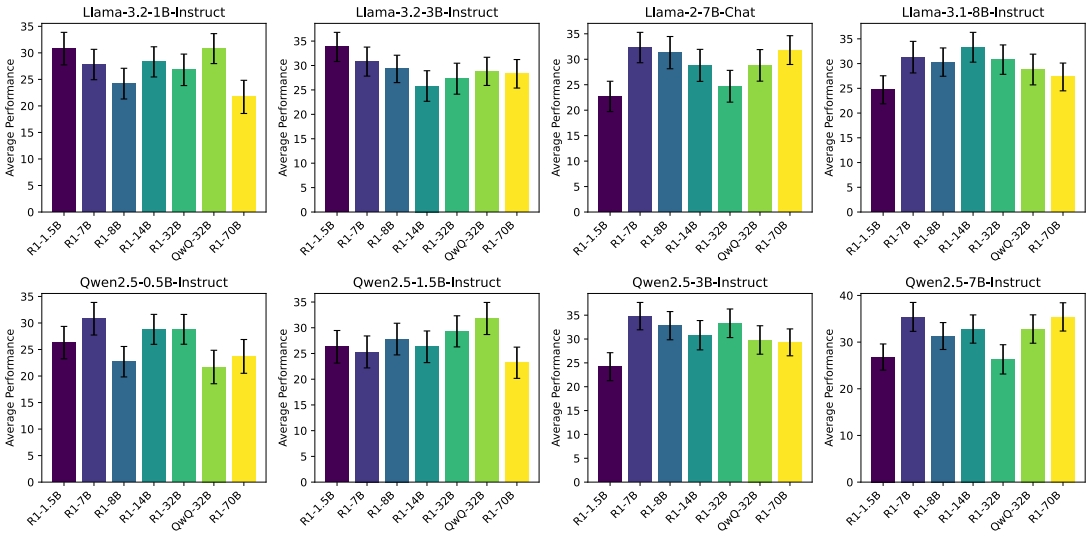

Figure 5.3: Average performance in the out-of-distribution setting. In each subplot, the x-axis represents the seven reasoning teacher models used in this study. For simplicity, we refer to DeepSeek-R1-Distill-Qwen-1.5B as R1-1.5B, and adopt similar abbreviations for other models as described in Section5.2. The top label (e.g., Llama-3.2-1B-Instruct) indicates the non-reasoning student model. We fine-tune each student model on reasoning training data distilled from a specific reasoning teacher model, and evaluate the fine-tuned model on out-of-distribution benchmarks to report the average performance.

effective knowledge transfer depends on both the capacity of student models to absorb complex reasoning patterns and the quality of reasoning demonstrations provided by teacher models.

§ 5.4 Cross-Domain Generalization of Long Reasoning CoT Distillation

Figure 5.3 presents the out-of-distribution performance of various student models fine-tuned on reasoning datasets distilled from teacher reasoning models (R1-1.5B through

R1-70B) and evaluated on factual knowledge and QA reasoning tasks. The results reveal several notable patterns that contrast with in-distribution performance.

5.4.1 Diminished Correlation Between Model Scale and Generalization

As we can see, out-of-distribution performance exhibits a remarkably flat trend across model scales. Performance ranges generally fall within 20-35% for most model-teacher combinations, suggesting that the ability to generalize mathematical reasoning to broader factual reasoning tasks is not strongly correlated with model size. For instance, Llama-3.2-1B-Instruct (21.7-30.8%) sometimes outperforms the much larger Llama-3.1-8B-Instruct (24.7-33.3%) depending on the teacher model.

5.4.2 Teacher Model Selection Critically Impacts Generalization

Our results indicate that teacher model selection has a more pronounced effect on cross-domain generalization than on in-domain performance. For example, when using Qwen2.5-3B-Instruct as the student, performance varies from 24.2% (with R1-1.5B teacher) to 34.8% (with R1-7B teacher), a substantial 10.6 percentage point difference. This suggests that certain teacher models may encode reasoning patterns that are more transferable across task domains.

Interestingly, mid-sized teacher models (R1-7B, R1-8B) frequently outperform their larger counterparts in facilitating cross-domain generalization. This contradicts the conventional wisdom that larger teachers invariably provide better knowledge transfer, and indicates that reasoning quality and transferability may be partially decoupled from

model scale.

5.4.3 Architecture-Specific Transfer Characteristics

The results reveal distinct transfer patterns across model families. The Qwen2.5 family exhibits greater performance variability across teacher models compared to the Llama family, suggesting architecture-specific sensitivities to teacher model selection. We hypothesize that this discrepancy is due to the architectural and post-training differences between Qwen and Llama models. Qwen-2.5's architectural feature set (Qwen et al., 2025b) (such as early-layer activity, MoE gating, complex post-training) amplifies any teacher model bias or style mismatch. LLaMA, by contrast, is architecturally designed to be more stable under teacher variability via uniform depth usage, dense routing and robust post-training (Grattafiori and Dubey, 2024).

Notably, Llama-2-7B-Chat, which performed poorly on in-distribution tasks, achieves competitive out-of-distribution performance (22.7-32.3%), indicating that some architectures may better preserve general reasoning capabilities even when fine-tuned on domain-specific tasks.

These findings uncover the complex relationship between mathematical reasoning acquisition and broader reasoning generalization. The non-monotonic relationship between model scale and cross-domain transfer efficacy suggests that effective knowledge distillation for generalization requires careful consideration of both student and teacher architectures beyond simply selecting the largest available models.

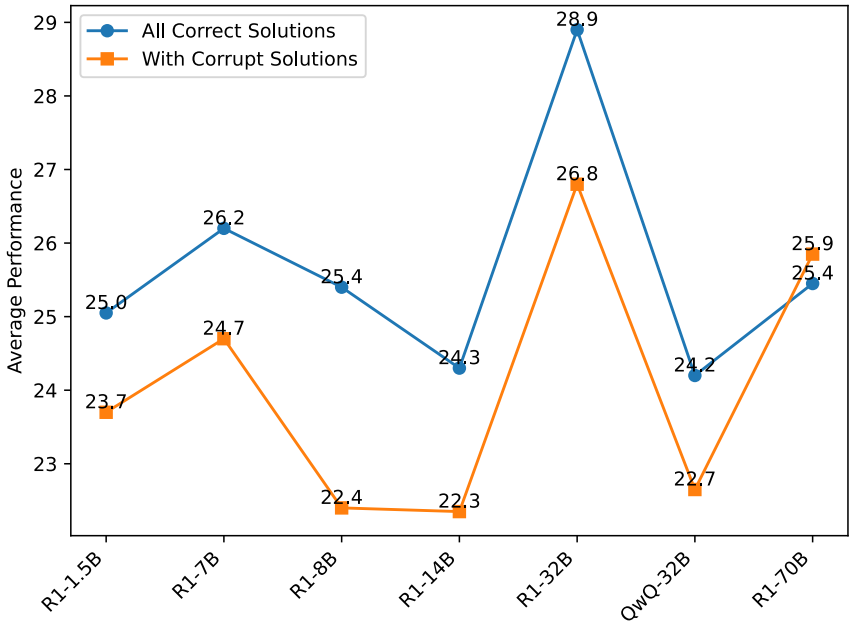

Figure 5.4: Average performance of Qwen2.5-1.5B-Instruct fine-tuned on correct-answer reasoning data ("All Correct Solutions") versus on all distilled samples ("With Corrupt Solutions"), across seven teacher reasoning models.

§ 5.5 Impact of Long Reasoning CoT Quality on Distillation Effectiveness

Figure 5.4 illustrates the performance of Qwen2.5-1.5B-Instruct when fine-tuned on reasoning datasets with varying solution quality across different teacher models. We compare two training conditions: "All Correct Solutions" where only samples (5,009 samples) with verified correct solutions are used for fine-tuning, and "With Corrupt Solutions" where the model is trained on the complete dataset (7,500 samples) including samples with incorrect solutions.

5.5.1 Quality-Quantity Trade-off in Long Reasoning CoT Distillation

Our results demonstrate a consistent performance advantage when training exclusively on correct solutions. Across all teacher models, the "All Correct" condition (averaging 25.1% to 28.9%) outperforms the "With Corrupt" condition (averaging 22.4% to 26.8%). This finding underscores the critical importance of solution quality in reasoning distillation, suggesting that the precision of reasoning demonstrations outweighs the benefits of increased training data volume when that additional data contains errors. The performance gap between conditions is particularly pronounced with certain teacher models. For instance, with R1-14B as the teacher, training on clean data yields an average performance of 24.3%, while including corrupt samples reduces performance to 22.4%, resulting in an 8.5% relative decrease. This indicates that some teacher models may be more sensitive to solution quality during knowledge transfer.

5.5.2 Teacher Model Sensitivity to Long Reasoning CoT Quality

Interestingly, the performance gap between training conditions varies non-uniformly across teacher models. The R1-32B teacher shows minimal difference between conditions (28.9% vs. 26.8%), suggesting that representations from larger models may be more robust to noise in training data. Conversely, mid-sized teachers like R1-14B exhibit larger performance deltas, indicating potentially higher sensitivity to solution quality. This pattern suggests that the impact of solution quality on distillation effectiveness depends on teacher model characteristics beyond mere scale. The representational robustness of teacher models may influence how effectively student models can filter

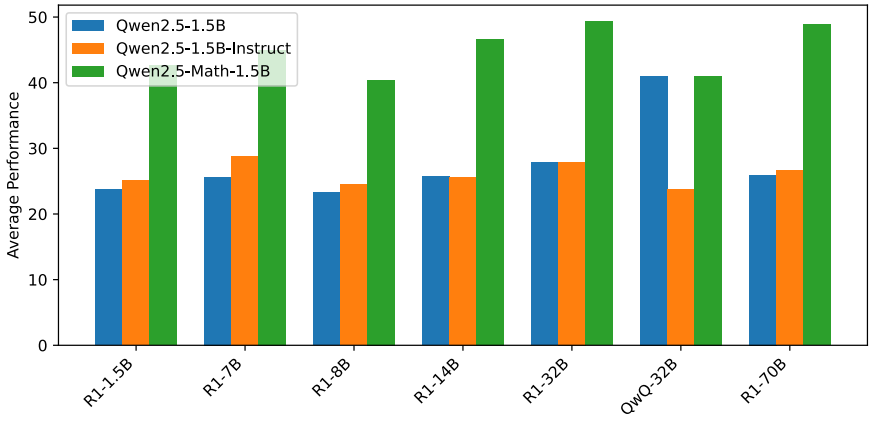

Figure 5.5: In-distribution performance of different types of base student models.

signal from noise during the distillation process.

These results have important implications for efficient knowledge distillation in long CoT reasoning tasks. They suggest prioritizing curating high-quality reasoning demonstrations over maximizing training data volume, particularly when working with mid-sized teacher models. This quality-first approach appears crucial for effectively transferring reasoning capabilities from large reasoning models to non-reasoning models.

§ 5.6 Impact of Student Model Pre-training on Long Reasoning CoT Distillation

Figure 5.5 and Figure 5.6 present an analysis of how pre-training objectives affect knowledge distillation outcomes across different teacher models. We compare three variants of the Qwen2.5-1.5B architecture: the base model, an instruction-tuned version, and a mathematics-specialized version, all fine-tuned on reasoning data distilled from various teacher models.

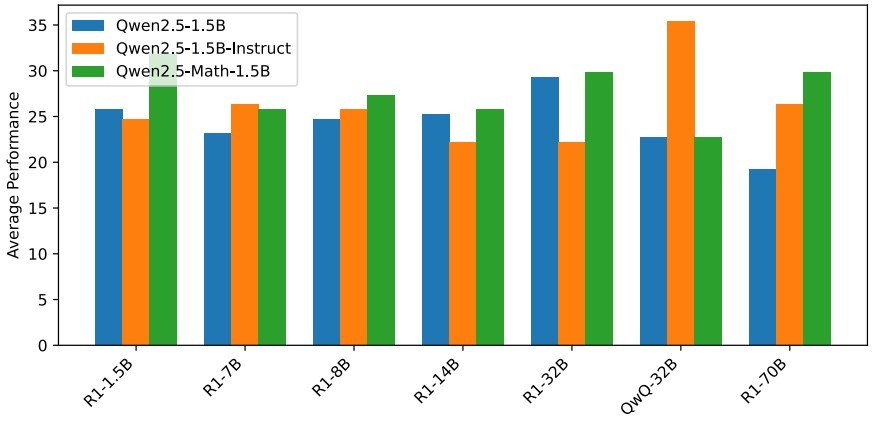

Figure 5.6: Out-of-distribution performance of different types of base student models.

5.6.1 Domain-Specific Pre-training Significantly Amplifies In-Distribution Performance

Figure 5.5 reveals that Qwen2.5-Math-1.5B substantially outperforms both the base and instruction-tuned variants across all teacher models on in-distribution mathematical reasoning tasks. The math-specialized model achieves average performance ranges of 42-53%, compared to 23-29% for the base model and 24-28% for the instruction-tuned variant. This dramatic performance gap (70-100% relative improvement) highlights the critical importance of domain-aligned pre-training in establishing the foundational capabilities needed to effectively absorb mathematical reasoning skills through distillation. Notably, this advantage persists regardless of teacher model size or architecture, suggesting that domain-specific pre-training creates a receptive substrate that consistently enhances the student model's ability to internalize mathematical reasoning patterns.

5.6.2 Diminished Pre-training Effects in Cross-Domain Transfer

In contrast, Figure 5.6 demonstrates that the advantages of specialized pre-training diminish considerably when evaluating on out-of-distribution reasoning tasks. The performance gap narrows significantly, with Qwen2.5-Math-1.5B achieving 22.7-31.8%, compared to 19.2-29.3% for the base model and 22.2-35.4% for the instruction-tuned variant. In several cases, particularly with QwQ-32B as teacher, the instruction-tuned model actually outperforms the math-specialized model on out-of-distribution tasks. This pattern suggests that while domain-specific pre-training enhances in-domain reasoning acquisition, it may offer limited benefits or even create constraints for cross-domain generalization. The instruction-tuned model's competitive out-of-distribution performance indicates that general-purpose instruction following might better facilitate transfer of reasoning capabilities across domains than highly specialized pre-training.

5.6.3 Teacher-Student Compatibility Patterns

Our analysis also reveals noteworthy interactions between teacher models and student pre-training objectives. For instance, QwQ-32B produces anomalous effects across student variants, yielding strong performance with the base model but underperforming with the instruction-tuned model on in-distribution tasks. Conversely, it produces the highest out-of-distribution performance when paired with the instruction-tuned student.

We demonstrate that effective distillation of long reasoning CoT capabilities depends not only on the individual properties of teacher and student models, but also on their compatibility. The optimal choice of student model architecture and pre-training may vary depending on the specific teacher model and the targeted reasoning domain, high-

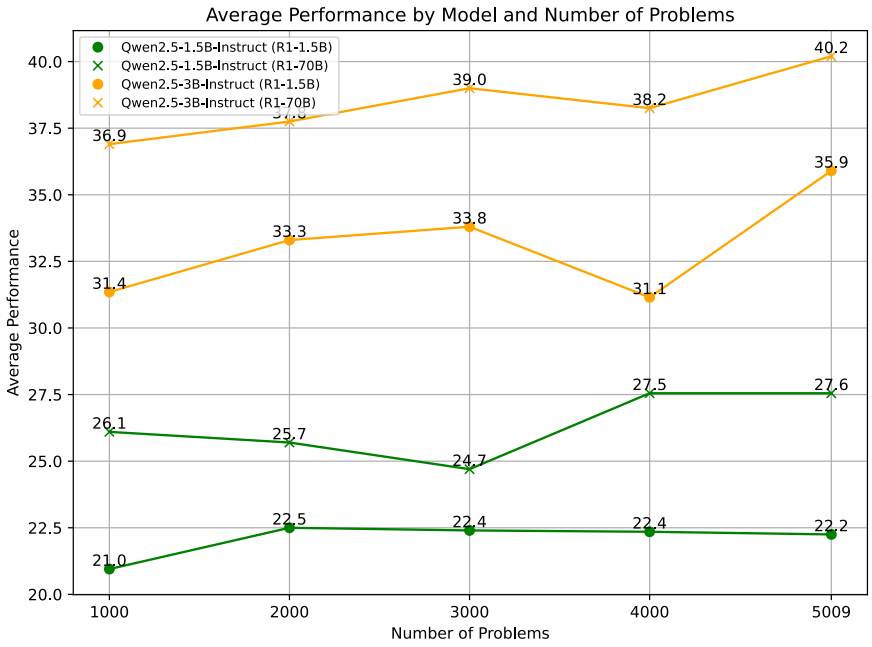

Figure 5.7: Average performance of two student models (Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct) fine-tuned on different sizes (1000, 2000, etc) of mathematical reasoning datasets distilled from two long CoT reasoning models: R1-1.5B and R1-70, respectively.

lighting the need for careful model selection in distillation pipelines.

§ 5.7 Impact of Training Data Size on Long Reasoning CoT Knowledge Transfer

Figure 5.7 illustrates how the volume of distilled reasoning problems affects knowledge transfer across different student and teacher model configurations. We examine the performance trajectories of two student models, Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct, when fine-tuned on increasingly larger samples of reasoning data distilled from either R1-1.5B or R1-70B teacher models.

5.7.1 Scaling Relationships in Reasoning Distillation

Our results demonstrate a consistent but modest upward trend in mathematical reasoning performance as training data increases from 1,000 to 5,009 problems. The performance of Qwen2.5-1.5B-Instruct improves from 21.0% to 22.2% (R1-1.5B teacher) and from 26.1% to 27.6% (R1-70B teacher), while Qwen2.5-3B-Instruct improves from 31.4% to 35.9% (R1-1.5B teacher) and from 36.9% to 40.2% (R1-70B teacher). This scaling pattern is notable for its nonlinearity: the performance gains from additional training data exhibit diminishing returns, particularly for the R1-70B teacher configurations. This suggests that beyond a certain threshold, the diversity and quality of reasoning examples may be more important than sheer quantity for effective knowledge transfer.

5.7.2 Teacher Model Size Influence

Notably, the performance gap between teacher models remains relatively stable as dataset size increases, indicating that the qualitative advantage of larger teacher models persists regardless of training data volume. This finding underscores the importance of teacher model selection in reasoning distillation pipelines and suggests that larger teachers provide inherently more effective reasoning demonstrations that cannot be compensated for simply by increasing the volume of examples from smaller teachers.

§ 5.8 Conclusion

This work presents the first systematic investigation of long CoT reasoning distillation across diverse model architectures, scales, and domains. Our findings challenge several conventional assumptions about knowledge transfer in reasoning tasks, and provide

actionable guidance for developing more capable yet computationally efficient reasoning models.

§ 5.9 Limitations

Despite the comprehensive nature of our investigation, several limitations should be acknowledged. First, our study focuses primarily on mathematical reasoning, while representative of structured problem-solving, which may not capture all aspects of reasoning encountered in other domains such as logical deduction, ethical reasoning, or causal inference. The patterns we observe might vary across different reasoning types. Second, our experiments are constrained to specific non-reasoning student model families (primarily Qwen2.5 and Llama), and the architectural compatibility effects we identify may not generalize to other model architectures. Additionally, while we explore reasoning teacher models ranging from 1B to 70B parameters, we do not investigate even larger teacher models (e.g., trillion-parameter models) that might reveal different distillation dynamics. Due to budget constraints, we do not train non-reasoning student models more than 8B parameters. Third, our evaluation methodology, may not fully capture all aspects of reasoning quality. In particular, our metrics focus primarily on answer correctness rather than evaluating the quality, coherence, or human-likeness of the reasoning process itself. This leaves open questions about whether the non-reasoning students faithfully reproduce the reasoning strategies of their teachers or merely learn to generate correct answers through different means. Fourth, we examine supervised fine-tuning as the primary distillation approach, without exploring alternative knowledge transfer techniques such as response-based distillation, intermediate feature matching, or

adversarial methods. Different distillation methodologies might yield different patterns in reasoning knowledge transfer.

Chapter 6

Conclusion and Future Work

§ 6.1 Conclusion

This thesis addressed three fundamental challenges in advancing the reasoning capabilities of large language models: generalization to novel domains, efficiency in complex reasoning tasks, and effective transfer of reasoning abilities across models. Using claim verification as a representative reasoning-intensive task, we explored solutions that integrate structured synthetic data generation, efficient multi-step reasoning, and systematic reasoning distillation.

First, we demonstrated that structured, domain-diverse synthetic data can substantially improve zero-shot generalization without relying on costly human annotation or domain-specific corpora. Second, we introduced an efficient reasoning framework that combines structured Chain-of-Thought distillation with reinforcement learning to balance accuracy and computational cost during inference time. Third, our comprehensive study of reasoning distillation uncovered non-trivial patterns in teacher–student transfer, highlighting the critical role of architecture-specific factors, reasoning style alignment,

and data quality.

Collectively, these contributions advance our understanding of how to design and train reasoning-optimized LLMs. The findings not only improve the performance of LLMs in claim verification but also provide general principles applicable to a broad range of reasoning-intensive NLP tasks.

§ 6.2 Future Work

While this thesis makes significant progress toward improving LLM reasoning, several promising research directions remain:

6.2.1 Synthetic Data Generation for Zero-Shot Claim Verification

Future research can explore integrating retrieval-augmented generation with synthetic data pipelines to better capture real-world factual grounding while maintaining domain diversity. Additionally, incorporating controlled noise or adversarial perturbations into synthetic datasets may improve model robustness against ambiguous or misleading claims. Another promising direction is to extend SYNTHVERIFY to multi-modal reasoning tasks, such as claim verification that involves both textual and visual evidence.

6.2.2 Efficient Reasoning for Context-Grounded Claim Verification

Building on the R1-ClaimVerifier framework, future work can investigate adaptive reasoning strategies that dynamically adjust the depth of reasoning based on input complexity. Integrating lightweight verification modules or early-exit mechanisms could further reduce inference costs without sacrificing accuracy. Moreover, extending

the framework to support online learning from user feedback could enable continual improvement in dynamic, real-world settings.

6.2.3 Rethinking Reasoning Distillation for Reasoning LLMs

Future work could expand the distillation study to cover more diverse reasoning paradigms, including abductive reasoning, analogical reasoning, and causal inference. Developing metrics for quantifying reasoning style compatibility between teachers and students could lead to more principled teacher selection. Furthermore, exploring hybrid distillation approaches that combine high-quality reasoning demonstrations with targeted self-improvement loops may yield even more effective transfer of reasoning capabilities.

In summary, the methods and insights developed in this thesis provide a strong foundation for future research in LLM reasoning, with clear pathways toward building models that are not only more capable, but also more efficient, robust, and adaptable.

A Synthetic Data Generation for Zero-Shot Claim Verification

§ A.1 Evaluation Benchmarks and Fine-Tuning Details

A.1.1 Dataset Statistics of the Evaluation Benchmarks

In this study, we used the following pre-processed datasets in Pan et al. (2023). The statistics of these dataset are shown in Table A.1. Claims are typically sentence-level statements, which can be either real-world natural claims sourced from websites, textbooks, and forums, or artificially generated by crowd-workers. Evidence, the key information used to validate a claim, is often drawn from textual sources such as news articles, academic papers, and Wikipedia documents. Claim labels vary, with com-

	Dataset	Domain	Claim	Evidence	Label	# Claims		Avg. # tokens	
						Train	Test	Claim	Evid.
I	FEVER	Wikipedia	artificial	sent-level	S (52%), R (22%), N (26%)	145,327	19,972	9.4	35.9
	VitaminC	Wikipedia	artificial	sent-level	S (50%), R (35%), N (15%)	370,653	63,054	12.6	29.5
	FoolMeTwice	Wikipedia	artificial	sent-level	S (49%), R (51%)	10,419	1,169	15.3	37.0
II	ClimateFEVER	Climate	natural	sent-level	S (25%), R (11%), N (64%)	6,140	1,535	22.8	33.8
	Sci-Fact	Science	natural	sent-level	S (43%), R (22%), N (35%)	868	321	13.8	61.9
	PubHealth	Health	natural	sent-level	S (60%), R (36%), N (4%)	8,370	1,050	15.7	137.6
	COVID-Fact	Forum	natural	sent-level	S (32%), R (68%)	3,268	818	12.4	82.5
	FAVIQ	Question	natural	doc-level	S (50%), R (50%)	17,008	4,260	15.2	304.9

Table A.1: List of the 8 fact verification datasets for our study and their characteristics.

mon formats being binary (supports/refutes) or three-class labels (supports/refutes/not enough info).

FEVER (Thorne et al., 2018) It treats Wikipedia as the primary evidence source and instructs crowdworkers to alter sentences from Wikipedia articles to create claims, which are then classified as supporting, refuting, or lacking sufficient information.

VitaminC (Schuster et al., 2021) It generates contrastive evidence pairs for each claim, where the pairs are nearly identical in wording and content, except that one supports the claim while the other does not.

FoolMeTwice (Eisenschlos et al., 2021) develops a multi-player game that encourages diverse strategies for constructing claims (*e.g.*, temporal inference) based on Wikipedia, leading to more complex claims with reduced lexical overlap with the evidence.

Climate-FEVER (Diggelmann et al., 2020) It consists of 1,535 real-world claims related to climate change, collected from the Internet. The five most relevant sentences from Wikipedia are retrieved as evidence, and human annotators label each sentence as supporting, refuting, or providing insufficient information for verification.

Sci-Fact (Wadden et al., 2020a) It includes 1.4K expert-generated scientific claims, each linked to abstracts containing supporting evidence, annotated with labels and sentence-level rationales.

PubHealth (Kotonya and Toni, 2020a) It contains 11.8K claims accompanied by gold standard judgments crafted by journalists to support or refute the claims. The claims

come from five fact checking platforms, news headlines, and news reviews. We pair each claim with its corresponding judgment text as evidence.

COVID-Fact (Saakyan et al., 2021) It consists of 4,086 claims related to the COVID-19 pandemic, collected from the *r/COVID19* subreddit. We use the sentence-level evidence annotated by crowd-workers as the supporting evidence.

FAVIQ (Park et al., 2022) It contains 26K claims derived from naturally occurring ambiguous questions posed by real users. Each claim is paired with an answer-containing Wikipedia paragraph as its document-level evidence.

Since many of the original datasets do not provide a publicly available test set, we use their original dev splits as our evaluation sets. Furthermore, we standardize label naming conventions to `supports`, `refutes`, and `not enough info`.

A.1.2 Fine-Tuning Details

We use LoRA (Hu et al., 2022) and FlashAttention-2 (Dao, 2024) with a cosine learning rate schedule and an initial learning rate of 1e−4 to fine-tune LLMs. The maximum sequence length is 1024 and the batch size is 4. The LoRA rank is 16 for all models. We use the Huggingface Transformers (Wolf et al., 2020) library to train all models.

§ A.2 Prompts

Prompt	Table Reference
Synthetic data generation	
Knowledge domain generation	Table A.3
Evidence plan generation	Table A.5
Evidence document synthesis	Table A.7
Key aspect extraction	Table A.9
Supported claim generation	Table A.11
Refuted claim generation	Table A.13
Not-Enough-Info claim generation	Table A.14
Model evaluation	
Zero-shot evaluation	Table A.15
Zero-shot Chain-of-Thought evaluation	Table A.16
Ablation studies	
Removing knowledge domain Specification	Table A.17
Removing Evidence Plan Generation	Table A.18
Removing Key Aspect Generation	Table A.19
Direct prompting	Table A.20

Table A.2: A collection of all prompts used in synthetic data generation, model evaluation and ablation studies.

This appendix provides the complete set of prompts used in our synthetic data generation pipeline, along with example outputs for each stage.

A.2.1 Knowledge Domain Generation

Please generate a list of diverse domains that is highly relevant for claim verification tasks. For each domain, provide: 1. The domain name. Do not include "&" or "and" in the generated domain name. Do not combine two or more words together to form a single domain name. 2. A concise description (1–2 sentences) highlighting its unique linguistic and factual characteristics and explaining why claims in this domain are challenging to verify. 3. A list of several domain properties. Format your response exactly as follows: Domain: [Domain Name] Description: [Your description here] Properties: [Property 1;Property 2;Property 3]

Table A.3: The prompt for knowledge domain generation.

Example Output: Domain: Climate Science Description: A dynamic domain involving long-term alteration of temperature and typical weather patterns. Claims are challenging to verify due to reliance on complex climate models and long-term observational data. Properties: ["Temperature data", "Weather patterns", "Climate models"]

Table A.4: The example output for knowledge domain generation.

A.2.2 Evidence Document Generation

You are an expert academic writer in the field of claim verification. Given the following knowledge domain:

Domain: [Domain Name]
Description: [A concise description (1–2 sentences) highlighting its unique linguistic and factual characteristics and explaining why claims in this domain are challenging to verify.]
Properties: [Property 1; Property 2; Property 3]

Please generate a concise bullet-point plan for a synthetic evidence document in this domain. Your plan should include:
• An Introduction that sets the description context.
• Key Factual Elements that reflect the domain properties.
• A Conclusion summarizing the key insight or challenge.

Table A.5: The prompt for evidence plan generation.

Example Output:
Domain: Political News
Description: Political news is characterized by rapid developments, partisan language, and selective reporting, making fact verification challenging.
Properties: Real-time data; Bias and spin; Official vs. independent sources

Evidence plan:
• Introduction: Briefly introduce a recent political event or debate.
• Key Factual Elements:
  - Reported statistic or claim by a political figure.
  - Data from official sources (e.g., government records) versus independent audits.
  - Mention of bias or discrepancies in reporting.
• Conclusion: Summarize the discrepancy between the claimed figure and the verified data, emphasizing its impact on public trust.

Table A.6: The example output for evidence plan generation.

You are a domain expert in claim verification. Expand the following evidence plan into a full evidence document suitable for a claim verification dataset. Ensure that the final document is coherent, detailed, and written in a formal academic style. Evidence plan: Insert the plan generated here. Please generate the complete evidence document following this structure.

Table A.7: The prompt for evidence document synthesis.

Example Output: Evidence Document: In a recent national debate, a prominent political figure claimed that public spending on infrastructure had increased by 12% over the last fiscal year. However, independent audits and official government records reveal that the actual increase was only 8%, largely due to delays in project implementation and discrepancies in budget reporting. This notable gap between the reported figure and the verified data highlights concerns over potential bias and spin in political reporting, which may ultimately undermine public trust in government accountability.

Table A.8: The example output for evidence document synthesis.

A.2.3 Claim Generation

You are a domain expert in claim verification. Given the following evidence document and its associated knowledge domain, please extract and list three key Aspects that capture different facets of the evidence. Each key aspect should be a concise description that highlights a distinct dimension (e.g., numerical details, source credibility, or temporal context) and aligns with the domain properties.

Evidence:
In a recent national debate, a prominent political figure claimed that public spending on infrastructure increased by 12% last fiscal year, while independent audits reported an actual increase of only 8%.

knowledge domain:
Domain: Political News
Description: Political news often features conflicting reports and varying interpretations due to partisan Aspects.
Properties: Real-time data; Bias and spin; Official vs. independent sources

Output Format:
1. [Key Aspect 1]
2. [Key Aspect 2]
3. [Key Aspect 3]

Table A.9: The prompt for the key aspect extraction from the evidence document.

Example Output:
Statistical Aspect: Focuses on the numerical difference between the 12% claimed and the 8% audited increase.
Source Reliability Aspect: Emphasizes the contrast between the official claim and the independent audits.
Contextual Aspect: Highlights the influence of political debate and potential bias in reporting.

Table A.10: The example output for the key aspect extraction from the evidence document.

You are a domain expert in claim verification. Based on the evidence document provided below and a specified key aspect, generate a supported claim that is factually consistent with the evidence while emphasizing the given aspect.

Evidence:
In a recent national debate, a prominent political figure claimed that public spending on infrastructure increased by 12% last fiscal year, while independent audits reported an actual increase of only 8%.

Key Aspect:
Statistical Aspect: Focuses on the numerical difference between the 12% claimed and the 8% audited increase.

Output Format:
Supported Claim: [Your claim here]

Table A.11: The prompt for generating supported claims based on key aspects.

Example Output:
Supported Claim: Independent audits confirm that public spending on infrastructure actually increased by only 8% last fiscal year, highlighting a significant statistical discrepancy compared to the 12% claimed by the political figure.

Table A.12: The example output for generating supported claims based on key aspects.

You are a highly skilled claim verification expert. Your task is to generate a refuted claim from a given supported claim by applying controlled perturbations that introduce discrepancies between the claim and the underlying evidence. In doing so, consider the following perturbation strategies:

1. Entity Substitution: Replace critical entities in the claim with alternative, contextually plausible candidates.
2. Temporal Modification: Adjust any time-related references, while maintaining overall historical coherence.
3. Relationship Reversal: Invert the relational dynamics between entities, altering the claim's meaning.
4. Attribute Modification: Modify specific properties or characteristics to create a divergence from the original details.

Now, given the following information, generate a refuted claim.

Supported Claim: [Insert Supported Claim Here]
Evidence: [Insert Relevant Evidence Here]

Output your response in the following format:
Refuted Claim: [Your refuted claim here]

Table A.13: The prompt for generating the refuted claim based on the supported claim.

You are a domain expert in claim verification. Given a supported claim that is fully detailed and verifiable based on the evidence, your task is to generate a "Not-Enough-Info" (NEI) claim by intentionally omitting or generalizing the key verifiable elements. Replace precise numerical figures and explicit qualifiers with vague, qualitative, or uncertain language, so that the claim no longer contains enough concrete information for conclusive verification. For example: Supported Claim: The study shows a 70% reduction in hospitalization rates. Not-Enough-Info Claim: The study suggests a significant reduction in hospitalization rates. Now, please generate a Not-Enough-Info claim for the following input: Supported Claim: [Insert Supported Claim Here] Evidence: [Insert Evidence Here] Output your response in the following format: Not-Enough-Info Claim: [Your generated Not-Enough-Info claim here]

Table A.14: The prompt for generating the Not-Enough-Info claim.

A.2.4 Model Evaluation

You are a highly skilled claim verification expert. You are given a claim and evidence. Your task is to verify the claim based solely on the evidence provided. Analyze the claim and evaluate the evidence. Based on your evaluation, return one of the following labels: "supports", "refutes" or "not enough info". Evidence: {EVIDENCE} Claim: {CLAIM} Label: {LABEL}

Table A.15: The prompt used for zero-shot evaluation.

You are a highly skilled claim verification expert. You are given a claim and evidence. Your task is to verify the claim based solely on the evidence provided. Analyze the claim and evaluate the evidence. Based on your evaluation, return one of the following labels: "supports", "refutes" or "not enough info". Let's think step by step. Evidence: {EVIDENCE} Claim: {CLAIM} Label: {LABEL}

Table A.16: The prompt used for zero-shot chain-of-thought evaluation.

A.2.5 Prompts for Ablation Studies

You are an expert academic writer in the field of claim verification. Given the following domain: Domain: [Domain Name] Please generate a concise bullet-point plan for a synthetic evidence document in this domain.

Table A.17: The prompt for the ablation study of removing knowledge domain Specification.

You are an expert academic writer. Generate an evidence document based on the following knowledge domain specification. Ensure that the final document is coherent, detailed, and written in a formal academic style. Domain: [Domain Name] Description: [A concise description (1–2 sentences) highlighting its unique linguistic and factual characteristics and explaining why claims in this domain are challenging to verify.] Properties: [Property 1; Property 2; Property 3] Please generate the complete evidence document following this structure. Evidence Document:

Table A.18: The prompt for the ablation study of removing evidence plan generation.

You are a domain expert in claim verification. Based on the evidence document provided below, generate a supported claim that is factually consistent with the evidence. Evidence:[Insert the evidence document here.] Output Format: Supported Claim: [Your claim here]

Table A.19: The prompt for the ablation study of removing key aspect extraction.

You are a domain expert in claim verification. Given a predefine domain, generate one evidence document and three claims about that domain so that their labels are “supports”, “refutes” and “not enough info”, respectively. Domain: [Domain name] Output Format: Evidence Document: [Evidence Document] Supported Claim: [Supported Claim] Refuted Claim: [Refuted Claim] Not-Enough-Info Claim: [Not-Enough-Info Claim]

Table A.20: The prompt for the ablation study of the direct prompting method.

B Efficient Reasoning for Context-Grounded Claim Verification

§ B.1 The Prompt of Claim Verification CoT Templates

<think> 1. Parse the claim and decompose it into its essential components (e.g., key entities, events, numerical claims). 2. Scan the provided document to extract evidence segments that correspond to each component. 3. Summarize the extracted evidence succinctly for each sub-claim. 4. Map the evidence to the relevant parts of the claim to form an initial reasoning trace. 5. Conclude with a preliminary assessment of the claim's veracity based on the collected evidence. </think>

Table B.1: The prompt for distilling synthetic CoTs for the context-grounded claim verification.

§ B.2 Evaluation Benchmark Details

We conduct experiments on various context-grounded claim verifications, including text- and table-based documents. We use the accuracy as the evaluation metric. These tasks include some seen tasks:

DocNLI (Yin et al., 2021) A large-scale dataset for document-level NLI, covering news and wiki domains. Unlike traditional NLI datasets that focus on sentence-level inference, DocNLI evaluates a model's ability to determine entailment, contradiction, or neutrality between pairs of multi-sentence documents. It was constructed by transforming various summarization and document-pairing datasets (e.g., CNN/DailyMail, XSum, MultiNews) into NLI examples, where a summary and the corresponding document form the premise-hypothesis pairs. The goal is to encourage deeper understanding of discourse, context, and logical reasoning across longer texts.

PubHealth (Kotonya and Toni, 2020b) It contains 11.8K claims accompanied by gold standard judgments crafted by journalists to support or refute the claims. The claims come from five fact checking platforms, news headlines, and news reviews. We pair each claim with its corresponding judgment text as evidence.

SciFact (Wadden et al., 2020a) It includes 1.4K expert-generated scientific claims, each linked to abstracts containing supporting evidence, annotated with labels and sentence-level rationales.

Some unseen tasks are also included.

FinDVer(Zhao et al., 2024b) The FinDVer dataset is designed for financial document verification tasks. It contains pairs of financial claims and supporting evidence from official financial documents, such as earnings reports, press releases, or filings. The goal is to determine whether a claim made in a document is supported, contradicted, or neutral with respect to the evidence provided. The dataset helps in training models to assess the factual accuracy of claims within financial texts.

SciTab (Lu et al., 2023) The SciTab dataset is a large-scale collection designed for scientific tabular data interpretation. It consists of tables from scientific papers, paired with natural language descriptions or questions that require reasoning about the data in the tables. The goal is to enable models to understand and answer questions based on the structured data presented in the form of tables. SciTab focuses on tasks like table comprehension, information extraction, and data-to-text generation to bridge the gap between structured and unstructured information in scientific literature.

WiCE (Kamoi et al., 2023b) The WiCE dataset is designed for real-world claim verification and focuses on entailment in the context of Wikipedia. It contains naturally occurring claims extracted from Wikipedia articles and pairs them with evidence sentences, also from Wikipedia, to test whether the evidence supports, refutes, or provides not enough information (NEI) for the claim.

C Rethinking Reasoning Distillation for Reasoning LLMs

§ C.1 The Prompt for Large Reasoning Model Distillation

You will be given a problem. Please reason step by step, and put your final answer within boxed{}: problem

Table C.1: The prompt for distilling long reasoning CoTs for the training dataset.

§ C.2 The Distilled Long Reasoning CoTs from Different Large Reasoning Models

Problem: When Claire divides her cupcakes into groups of 5, she has 3 remaining, and when she divides her cupcakes into groups of 7, she has 4 remaining. If Claire has fewer than 60 cupcakes, what is the sum of all possible quantities of cupcakes that she could have?

Table C.2: One of the training data example used for distilling long reasoning CoTs.

§ C.3 Examples Outputs

Okay, so I have this problem where Claire has some cupcakes, and when she divides them into groups of 5, she has 3 left over. And when she divides them into groups of 7, she has 4 left over. Also, she has fewer than 60 cupcakes. I need to find all possible quantities she could have and then sum them up. Hmm, let me think. This sounds like a problem involving modular arithmetic or maybe the Chinese Remainder Theorem. I remember that when you have a remainder when dividing by different numbers, you can set up some equations. So, let's denote the number of cupcakes as N. According to the problem: When divided by 5, the remainder is 3. So, mathematically, that's N ≡ 3 mod 5. Which means N can be written as 5k + 3 for some integer k. Similarly, when divided by 7, the remainder is 4. So, N ≡ 4 mod 7. That means N can also be written as 7m + 4 for some integer m. Since both expressions equal N, I can set them equal to each other: 5k + 3 = 7m + 4. Let me rearrange this equation to find a relationship between k and m. Subtracting 3 from both sides gives 5k = 7m + 1. So, 5k - 7m = 1. Hmm, this is a linear Diophantine equation. I think I can solve this using the method of finding integer solutions. I remember that to solve equations like ax + by = c, you can use the extended Euclidean algorithm. Let me try to find integers k and m that satisfy 5k - 7m = 1. Alternatively, maybe I can find a particular solution and then find the general solution. Let's see. Looking for small integers k and m that satisfy 5k - 7m = 1. Let me try k=2: 5*2=10. So, 10 -7m=1 → 7m=9 → m=9/7, which is not an integer. Not good. k=3: 15 -7m=1 → 7m=14 → m=2. Oh, that works! So, k=3 and m=2 is a particular solution. So, the general solution for k and m can be written as: k = 3 + 7t m = 2 + 5t Where t is any integer. Because the coefficients of k and m are 5 and 7, respectively, which are coprime, so the solutions will repeat every 7 and 5 steps. Now, since N = 5k + 3, let's substitute k: N = 5*(3 + 7t) + 3 = 15 + 35t + 3 = 18 + 35t. So, N can be expressed as 18 + 35t. Now, we also know that N < 60. So, 18 + 35t < 60 → 35t < 42 → t < 42/35 → t < 1.2. Since t must be an integer, t can be 0 or 1. Let me check t=0: N=18+0=18. Let's verify: 18 divided by 5 is 3 with remainder 3. Correct. 18 divided by 7 is 2 with remainder 4. Correct. t=1: N=18+35=53. Let's check: 53 divided by 5 is 10 with remainder 3. Correct. 53 divided by 7 is 7 with remainder 4. Correct. t=2: N=18+70=88, which is more than 60, so we stop here. So, the possible values of N are 18 and 53. Now, the problem asks for the sum of all possible quantities. So, 18 + 53 = 71. Wait, let me double-check if there are any other possible values. Since t can be 0 and 1, and t=2 gives N=88 which is over 60, so yes, only 18 and 53. So, the sum is 71.
</think> To solve the problem, we set up the equations based on the given conditions: 1. \(N \equiv 3 \mod 5 \) implies \(N = 5k + 3 \).
2. \(N \equiv 4 \mod 7 \) implies \(N = 7m + 4 \). Equating the two expressions for \(N \):
\[
5k + 3 = 7m + 4 \implies 5k - 7m = 1
\]
Solving this linear Diophantine equation, we find a particular solution \(k = 3 \), \(m = 2 \). The general solution is:
\[
k = 3 + 7t, \quad m = 2 + 5t
\]
Substituting back, \(N = 5(3 + 7t) + 3 = 18 + 35t \). Given \(N < 60 \), \(t \) can be 0 or 1, yielding \(N = 18 \) and \(N = 53 \). The sum of these values is \(18 + 53 = 71 \). **Answer:** The sum of all possible quantities is \boxed{71}.

Figure C.1: The response of R1-1.5B for the example shown in Table C.2

§ C.4 Evaluation Benchmark Details

Following the prior work (Guo et al. 2025b; Ye et al., 2025; Muennighoff et al., 2025), we conduct a comprehensive evaluation on a wide-range of reasoning tasks with both in-distribution and out-of-distribution settings.

C.4.1 In-Distribution Setting

These include several mathematical reasoning benchmarks:

Okay, so I have this math problem here about Claire and her cupcakes. Let me try to figure it out step by step. Hmm, the problem says that when she divides her cupcakes into groups of 5, she has 3 remaining. And when she divides them into groups of 7, she has 4 remaining. Also, she has fewer than 60 cupcakes. I need to find the sum of all possible quantities she could have. Alright, let's break this down. So, the first condition is that when she divides the cupcakes into groups of 5, there's a remainder of 3. That means the number of cupcakes, let's call it N, satisfies N ≡ 3 mod 5. In other words, N can be written as 5k + 3 for some integer k. The second condition is that when she divides them into groups of 7, the remainder is 4. So, N ≡ 4 mod 7. That means N can also be written as 7m + 4 for some integer m. So, I need to find all numbers N less than 60 that satisfy both these conditions. Then, sum them up. Let me think about how to approach this. Maybe I can list out the numbers that satisfy each condition and then find the common numbers. Starting with the first condition: N ≡ 3 mod 5. So, numbers are 3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58, 63, etc. But since N is less than 60, I can stop at 58. Now, the second condition: N ≡ 4 mod 7. Let's list those numbers. Starting from 4, adding 7 each time: 4, 11, 18, 25, 32, 39, 46, 53, 60, 67, etc. But since N is less than 60, the numbers are 4, 11, 18, 25, 32, 39, 46, 53. Now, I need to find the numbers that appear in both lists. Let me compare the two lists. First list (mod 5): 3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58. Second list (mod 7): 4, 11, 18, 25, 32, 39, 46, 53. Looking for common numbers. Let's see: - 18 is in both lists.
- 53 is also in both lists. Wait, are there any others? Let me check. Looking at the first list: 3,8,13,18,23,28,33,38,43,48,53,58. Second list: 4,11,18,25,32,39,46,53. So, 18 and 53 are the common numbers. Are there any more? Wait, let me check 53: 53 divided by 5 is 10 with remainder 3, yes. 53 divided by 7 is 7*7=49, so 53-49=4, so remainder 4. Correct. 18: 18/5=3*5=15, remainder 3. 18/7=2*7=14, remainder 4. Correct. Are there any other numbers? Let's see. Next after 53 would be 18 + LCM(5,7)=35, so 18+35=53, then 53+35=88, which is over 60. So, only 18 and 53. Wait, but wait, let me make sure I didn't miss any. Let me check each number in the first list against the second list. 3: Not in second list. 8: Not in second list. 13: Not in second list. 18: Yes, in second list. 23: Not in second list. 28: Not in second list. 33: Not in second list. 38: Not in second list. 43: Not in second list. 48: Not in second list. 53: Yes, in second list. 58: Not in second list. So, only 18 and 53. So, the possible numbers are 18 and 53. Wait, but wait, let me think again. Maybe I should solve it algebraically to make sure I didn't miss any. So, N ≡ 3 mod 5, so N = 5k + 3. Also, N ≡4 mod7, so 5k +3 ≡4 mod7. So, 5k ≡1 mod7. We can solve for k: 5k ≡1 mod7. Multiplicative inverse of 5 mod7 is needed. Since 5*3=15≡1 mod7, so inverse is 3. So, k ≡3*1=3 mod7. So, k=7m +3, where m is integer. So, N=5*(7m +3)+3=35m +15 +3=35m +18. So, N=35m +18. Now, N must be less than 60. So, 35m +18 <60. 35m <42. m <42/35=1.2. So, m can be 0,1. So, m=0: N=18. m=1: N=35+18=53. m=2: 70+18=88>60, so stop. So, N=18 and 53. So, the possible numbers are 18 and 53. Therefore, the sum is 18+53=71. Wait, but let me check if there are any other numbers. For m=0, 18; m=1,53; m=2,88>60. So, only two numbers. So, the sum is 18+53=71. Wait, but let me make sure I didn't make a mistake in the algebra. We had N=35m +18. So, for m=0: 18. m=1: 53. m=2: 88, which is over 60. So, yes, only 18 and 53. So, the sum is 71. I think that's correct.
</think> The problem involves finding numbers less than 60 that satisfy two modular conditions. By solving the congruences, we find that the possible numbers are 18 and 53. Summing these gives 71. Therefore, the final answer is: $\boxed{71}$

Figure C.2: The response of R1-7B for the example shown in Table C.2

Okay, so I have this problem here about Claire and her cupcakes. Let me try to figure it out step by step. Hmm, the problem says that when she divides her cupcakes into groups of 5, she has 3 left over. And when she divides them into groups of 7, she has 4 left. Also, she has fewer than 60 cupcakes. I need to find the sum of all possible quantities she could have. Alright, let's break this down. So, if she divides by 5 and has a remainder of 3, that means the number of cupcakes, let's call it N, can be expressed as N = 5k + 3 for some integer k. Similarly, when she divides by 7, the remainder is 4, so N = 7m + 4 for some integer m. So, N ≡ 3 mod 5 and N ≡ 4 mod 7. I think this is a problem that involves the Chinese Remainder Theorem. But since the numbers are small, maybe I can find N by listing out the possible numbers that satisfy both conditions. Let me list numbers that are 3 more than a multiple of 5. Starting from the smallest possible, which would be 3, then 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58, 63, etc. But since she has fewer than 60, I can stop at 58. Now, from this list, I need to pick the numbers that also satisfy N ≡4 mod7. Let's check each one: - 3 divided by 7: 3 ÷7 is 0 with remainder 3. So 3 mod7 is 3, which is not 4. So no.
- 8 ÷7 is 1 with remainder 1. Not 4.
- 13 ÷7 is 1 with remainder 6. Not 4.
- 18 ÷7 is 2 with remainder 4. Oh, that's good. So 18 is a candidate.
- Next, 23 ÷7 is 3 with remainder 2. Not 4.
- 28 ÷7 is exactly 4, so remainder 0. Not 4.
- 33 ÷7 is 4 with remainder 5. Not 4.
- 38 ÷7 is 5 with remainder 3. Not 4.
- 43 ÷7 is 6 with remainder 1. Not 4.
- 48 ÷7 is 6 with remainder 6. Not 4.
- 53 ÷7 is 7 with remainder 4. So 53 is another candidate.
- 58 ÷7 is 8 with remainder 2. Not 4. So the numbers less than 60 that satisfy both conditions are 18 and 53. Now, I need to sum these possible quantities. 18 + 53 is 71. So the sum is 71. Wait, let me double-check. For 18: 18 ÷5 is 3 with remainder 3, and 18 ÷7 is 2 with remainder 4. Correct. For 53: 53 ÷5 is 10 with remainder 3, and 53 ÷7 is 7 with remainder 4. Also correct. So both are valid. Are there any others? Let me see, after 53, the next number would be 53 + LCM(5,7)=35, which is 88, but that's more than 60, so no. So only 18 and 53. Sum is 71.
</think> To solve the problem, we need to find all numbers \(N \) less than 60 that satisfy the following conditions:
1. \(N \equiv 3 \pmod{5} \)
2. \(N \equiv 4 \pmod{7} \) ### Step 1: List numbers that satisfy \(N \equiv 3 \pmod{5} \)
Numbers that are 3 more than a multiple of 5 are:
\[3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58 \] ### Step 2: Check which of these numbers also satisfy \(N \equiv 4 \pmod{7} \)
- \(3 \div 7 = 0 \) remainder \(3 \) → Not 4
- \(8 \div 7 = 1 \) remainder \(1 \) → Not 4
- \(13 \div 7 = 1 \) remainder \(6 \) → Not 4
- \(18 \div 7 = 2 \) remainder \(4 \) → **Valid**
- \(23 \div 7 = 3 \) remainder \(2 \) → Not 4
- \(28 \div 7 = 4 \) remainder \(0 \) → Not 4
- \(33 \div 7 = 4 \) remainder \(5 \) → Not 4
- \(38 \div 7 = 5 \) remainder \(3 \) → Not 4
- \(43 \div 7 = 6 \) remainder \(1 \) → Not 4
- \(48 \div 7 = 6 \) remainder \(6 \) → Not 4
- \(53 \div 7 = 7 \) remainder \(4 \) → **Valid**
- \(58 \div 7 = 8 \) remainder \(2 \) → Not 4 ### Step 3: Sum the valid numbers
The valid numbers are 18 and 53. Their sum is:
\[18 + 53 = 71 \] Therefore, the sum of all possible quantities of cupcakes that Claire could have is:
\[
\boxed{71}
\]

Figure C.3: The response of R1-8B for the example shown in Table C.2

Okay, so I've got this math problem here, and I need to figure out how to solve it. Let me read it again to make sure I understand. The problem says: When Claire divides her cupcakes into groups of 5, she has 3 remaining. And when she divides them into groups of 7, she has 4 remaining. Also, she has fewer than 60 cupcakes. I need to find the sum of all possible quantities of cupcakes she could have. Hmm, alright. So, this seems like a problem involving division with remainders. I remember that when a number leaves a remainder when divided by another number, we can express it using modular arithmetic. Let me think about how to model this. First, when she divides the cupcakes into groups of 5 and has 3 left, that means the number of cupcakes, let's call it N, satisfies the equation N ≡ 3 mod 5. Similarly, when she divides them into groups of 7 and has 4 left, that translates to N ≡ 4 mod 7. So, I can write these as two separate congruences:
1. N ≡ 3 (mod 5)
2. N ≡ 4 (mod 7) And N is less than 60. I think the way to solve this is by finding numbers that satisfy both conditions. Maybe I can list out numbers that fit the first condition and then check which of those also fit the second condition. Let me start with the first condition: N ≡ 3 mod 5. So, numbers that when divided by 5 leave a remainder of 3. These numbers can be written as 5k + 3, where k is a non-negative integer. So, possible N values are 3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58, etc. But since N must be less than 60, the maximum N here would be 58. Now, from this list, I need to pick the numbers that also satisfy the second condition: N ≡ 4 mod 7. So, when these numbers are divided by 7, the remainder is 4. Let me go through each number and check: 1. 3: 3 divided by 7 is 0 with remainder 3. Not 4. So, no.
2. 8: 8 ÷7 is 1 with remainder 1. Not 4.
3. 13: 13 ÷7 is 1 with remainder 6. No.
4. 18: 18 ÷7 is 2 with remainder 4. Oh, that works! So 18 is a possible number.
5. 23: 23 ÷7 is 3 with remainder 2. No.
6. 28: 28 ÷7 is 4 with remainder 0. No.
7. 33: 33 ÷7 is 4 with remainder 5. No.
8. 38: 38 ÷7 is 5 with remainder 3. No.
9. 43: 43 ÷7 is 6 with remainder 1. No.
10. 48: 48 ÷7 is 6 with remainder 6. No.
11. 53: 53 ÷7 is 7 with remainder 4. Yes! So 53 is another possible number.
12. 58: 58 ÷7 is 8 with remainder 2. No. So, from the list, the numbers that satisfy both conditions are 18 and 53. Let me check if there are any more beyond 58, but since N must be less than 60, 58 is the last one to check, and 53 is the next one before that. So, I think that's all. Wait, but let me make sure I didn't miss any. Let me think about another approach. Maybe using the Chinese Remainder Theorem? I remember that when two moduli are coprime, there's a unique solution modulo the product. But 5 and 7 are coprime, so their product is 35. So, the solutions should repeat every 35 numbers. So, the first solution is 18, then adding 35 gives 53, and the next would be 88, which is over 60. So, yes, only 18 and 53 are the solutions under 60. So, the possible quantities are 18 and 53. The problem asks for the sum of all possible quantities. So, 18 + 53 equals 71. Wait, let me double-check my calculations. 18 + 53 is indeed 71. Hmm, but wait, is there another number between 53 and 60 that I might have missed? Let me see. The next number after 53 in the 5k+3 sequence is 58, which I checked earlier and it didn't satisfy the second condition. So, no, 53 is the last one. So, the sum is 18 + 53 = 71. Therefore, the answer should be 71.
</think> To solve the problem, we need to find all numbers \(N \) less than 60 that satisfy the following conditions:
1. \(N \equiv 3 \pmod{5} \)
2. \(N \equiv 4 \pmod{7} \) From the first condition, \(N \) can be expressed as \(N = 5k + 3 \) for some integer \(k \). Substituting this into the second condition, we solve for \(k \) such that \(5k + 3 \equiv 4 \pmod{7} \). Simplifying, we find \(k \equiv 1 \pmod{7} \), so \(k = 7m + 1 \) for some integer \(m \). Substituting back, \(N = 5(7m + 1) + 3 = 35m + 8 \). The valid values of \(N \) less than 60 are 18 and 53. The sum of these values is \(18 + 53 = 71 \). Therefore, the final answer is: $\boxed{71}$.

Figure C.4: The response of R1-14B for the example shown in Table C.2

Okay, so I've got this math problem here, and I need to figure it out step by step. Let me read it again to make sure I understand. Claire divides her cupcakes into groups of 5 and has 3 remaining. When she divides them into groups of 7, she has 4 remaining. She has fewer than 60 cupcakes. I need to find the sum of all possible quantities she could have. Hmm, so this sounds like a problem involving division with remainders. I remember that when a number leaves a remainder when divided by another number, it's related to modular arithmetic. So, let me try to translate the problem into equations. First, when Claire divides her cupcakes into groups of 5, she has 3 left. So, mathematically, that means the number of cupcakes, let's call it N, satisfies N ≡ 3 mod 5. In other words, N = 5k + 3 for some integer k. Similarly, when she divides them into groups of 7, she has 4 left. So, N ≡ 4 mod 7, which means N = 7m + 4 for some integer m. And we know that N is less than 60. So, N < 60. So, I need to find all numbers N < 60 such that N ≡ 3 mod 5 and N ≡ 4 mod 7. Then, sum all those possible Ns. I think the way to approach this is to find numbers that satisfy both congruences. Maybe I can list out numbers that satisfy one condition and see which ones also satisfy the other. Let me start by listing numbers that are 3 more than a multiple of 5. So, starting from 3, adding 5 each time: 3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58. Wait, but since N < 60, the next one after 58 would be 63, which is over 60, so I can stop at 58. Now, from this list, I need to find which numbers also satisfy N ≡ 4 mod 7. So, for each number in the list, I'll check if when divided by 7, the remainder is 4. Let me go through each number: 1. 3: 3 divided by 7 is 0 with remainder 3. Not 4. So, no.
2. 8: 8 ÷ 7 is 1 with remainder 1. Not 4.
3. 13: 13 ÷ 7 is 1 with remainder 6. Not 4.
4. 18: 18 ÷ 7 is 2 with remainder 4. Oh, that works! So, 18 is one possible number.
5. 23: 23 ÷ 7 is 3 with remainder 2. Not 4.
6. 28: 28 ÷ 7 is 4 with remainder 0. Not 4.
7. 33: 33 ÷ 7 is 4 with remainder 5. Not 4.
8. 38: 38 ÷ 7 is 5 with remainder 3. Not 4.
9. 43: 43 ÷ 7 is 6 with remainder 1. Not 4.
10. 48: 48 ÷ 7 is 6 with remainder 6. Not 4.
11. 53: 53 ÷ 7 is 7 with remainder 4. That works! So, 53 is another possible number.
12. 58: 58 ÷ 7 is 8 with remainder 2. Not 4. So, from the list, the numbers that satisfy both conditions are 18 and 53. Wait, let me double-check that. For 18: 18 ÷ 5 is 3 with remainder 3, which matches the first condition. 18 ÷ 7 is 2 with remainder 4, which matches the second condition. So, yes, 18 is good. For 53: 53 ÷ 5 is 10 with remainder 3, which is correct. 53 ÷ 7 is 7 with remainder 4, which is also correct. So, 53 is good too. Are there any others? Let me see. The next number after 53 in the list is 58, which didn't work. The next would be 63, which is over 60, so we can't include that. So, the possible quantities are 18 and 53. Now, the problem asks for the sum of all possible quantities. So, 18 + 53 = 71. Wait, but let me make sure I didn't miss any numbers. Maybe I should approach this using the Chinese Remainder Theorem or another method to confirm. The Chinese Remainder Theorem says that if the moduli are coprime, there's a unique solution modulo the product. Here, 5 and 7 are coprime, so there should be a unique solution modulo 35. So, the solutions will be of the form N = 35t + x, where x is the particular solution. Let me find x such that x ≡ 3 mod 5 and x ≡ 4 mod 7. Let me solve these two congruences. We can write x = 5k + 3. Substitute into the second equation: 5k + 3 ≡ 4 mod 7 So, 5k ≡ 1 mod 7 We need to find k such that 5k ≡ 1 mod 7. Multiplicative inverse of 5 mod 7 is needed. Let's find it. 5 * 3 = 15 ≡ 1 mod 7, because 15 - 14 =1. So, inverse is 3. So, k ≡ 3 * 1 ≡ 3 mod 7. So, k = 7m + 3 for some integer m. Thus, x = 5*(7m +3) +3 = 35m +15 +3 = 35m +18. So, the general solution is N = 35m +18. Now, since N <60, let's find m such that 35m +18 <60. 35m <42 → m < 42/35 ≈1.2. So, m can be 0 or 1. For m=0: N=18. For m=1: N=35+18=53. For m=2: 35*2+18=88, which is over 60, so stop. So, the solutions are 18 and 53, same as before. So, the sum is 18+53=71. Therefore, the sum is 71.
</think> To solve the problem, we need to find all numbers \(N \) less than 60 that satisfy the following conditions:
1. \(N \equiv 3 \pmod{5} \)
2. \(N \equiv 4 \pmod{7} \) ### Step 1: List numbers satisfying \(N \equiv 3 \pmod{5} \)
Numbers of the form \(N = 5k + 3 \) for \(k \geq 0 \) and \(N < 60 \) are:
\[
3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58
\] ### Step 2: Check which of these numbers also satisfy \(N \equiv 4 \pmod{7} \)
- \(3 \div 7 \) leaves remainder 3 (not 4)
- \(8 \div 7 \) leaves remainder 1 (not 4)
- \(13 \div 7 \) leaves remainder 6 (not 4)
- \(18 \div 7 \) leaves remainder 4 (valid)
- \(23 \div 7 \) leaves remainder 2 (not 4)
- \(28 \div 7 \) leaves remainder 0 (not 4)
- \(33 \div 7 \) leaves remainder 5 (not 4)
- \(38 \div 7 \) leaves remainder 3 (not 4)
- \(43 \div 7 \) leaves remainder 1 (not 4)
- \(48 \div 7 \) leaves remainder 6 (not 4)
- \(53 \div 7 \) leaves remainder 4 (valid)
- \(58 \div 7 \) leaves remainder 2 (not 4) Thus, the valid numbers are \(18 \) and \(53 \). ### Step 3: Sum the valid numbers
\[
18 + 53 = 71
\] Therefore, the sum of all possible quantities of cupcakes is: $\boxed{71}$ I hope it is correct.

Figure C.5: The response of R1-32B for the example shown in Table C.2

Okay, so I've got this math problem here, and I need to figure it out step by step. Let me read it again to make sure I understand. Claire is dividing her cupcakes into groups. When she makes groups of 5, she has 3 left over. And when she makes groups of 7, she has 4 left. Also, she has fewer than 60 cupcakes. I need to find all possible numbers of cupcakes she could have and then add them up. Hmm, this sounds like a problem involving division with remainders. So, I think it's related to modular arithmetic. Let me break it down. First, when she divides by 5 and has a remainder of 3, that means the number of cupcakes, let's call it N, satisfies N ≡ 3 mod 5. Similarly, when she divides by 7 and has a remainder of 4, that means N ≡ 4 mod 7. So, I need to find all numbers N less than 60 that satisfy both these conditions. Then, sum them up. I remember that this is a system of congruences, and maybe the Chinese Remainder Theorem can help here. But I'm a bit fuzzy on the exact method, so I'll try to approach it step by step. Let me write down the two conditions: 1. N ≡ 3 mod 5
2. N ≡ 4 mod 7 I can express the first condition as N = 5k + 3, where k is some integer. Then, substitute this into the second condition. So, substituting into the second equation: 5k + 3 ≡ 4 mod 7. Let me solve for k. Subtract 3 from both sides: 5k ≡ 1 mod 7. Now, I need to find k such that 5k ≡ 1 mod 7. To find k, I can find the multiplicative inverse of 5 mod 7. The inverse of 5 is a number m where 5m ≡ 1 mod 7. Let me think: 5*3=15, which is 1 mod 7 because 15-14=1. So, m=3. So, multiplying both sides of 5k ≡1 mod7 by 3 gives k ≡3*1=3 mod7. So, k ≡3 mod7. That means k can be written as k=7m +3, where m is an integer. Now, substituting back into N=5k+3: N=5*(7m +3)+3 = 35m +15 +3 =35m +18. So, N=35m +18. Now, since N must be less than 60, let's find all possible m such that N<60. So, 35m +18 <60. Subtract 18: 35m <42. Divide by 35: m <42/35, which is 1.2. Since m must be an integer, m can be 0 or 1. So, m=0: N=35*0 +18=18. m=1: N=35*1 +18=53. Wait, m=2 would give 35*2+18=70+18=88, which is way over 60, so that's too big. So the possible N values are 18 and 53. Wait, but let me check if these satisfy the original conditions. For N=18: 18 divided by 5 is 3 with remainder 3. Correct. 18 divided by 7 is 2 with remainder 4. Correct. For N=53: 53 divided by 5 is 10 with remainder 3. Correct. 53 divided by 7 is 7*7=49, 53-49=4. So remainder 4. Correct. So both 18 and 53 are valid. Wait, but wait a second. Let me make sure I didn't miss any other possible m values. Because sometimes when dealing with moduli, there might be more solutions. Wait, the modulus here is 35 because 5 and 7 are coprime, so the solutions repeat every 35. So the next solution after 53 would be 53+35=88, which is over 60, so that's why m=2 is too big. So the possible Ns are 18 and 53. So the sum is 18 +53=71. Wait, but let me think again. Did I miss any numbers? Wait, let me list all numbers less than 60 that are ≡3 mod5 and ≡4 mod7. Alternatively, I can list numbers that are 3 mod5 and check which are 4 mod7. Numbers that are 3 mod5: 3,8,13,18,23,28,33,38,43,48,53,58. Now, check which of these are ≡4 mod7. Let's compute each mod7: 3 mod7=3 → not 4. 8 mod7=1 → no. 13 mod7=6 → no. 18 mod7: 7*2=14, 18-14=4 → yes. So 18 is good. 23 mod7: 7*3=21, 23-21=2 → no. 28 mod7=0 → no. 33 mod7: 7*4=28, 33-28=5 → no. 38 mod7: 7*5=35, 38-35=3 → no. 43 mod7: 7*6=42, 43-42=1 → no. 48 mod7: 7*6=42, 48-42=6 → no. 53 mod7: 7*7=49, 53-49=4 → yes. 58 mod7: 7*8=56, 58-56=2 → no. So only 18 and 53 satisfy both conditions. So the sum is 18+53=71. Wait, but wait another thing. Let me make sure I didn't miss any numbers. Let me think about the general solution again. The general solution is N=35m +18. For m=0, N=18; m=1, N=53; m=2, N=88 (too big). So yes, only two numbers. So the sum is 18+53=71. Therefore, the final answer is 71.
</think> Claire's problem involves finding the number of cupcakes, \(N \), that satisfies two conditions: 1. When divided into groups of 5, there is a remainder of 3, i.e., \(N \equiv 3 \pmod{5} \).
2. When divided into groups of 7, there is a remainder of 4, i.e., \(N \equiv 4 \pmod{7} \). Additionally, \(N \) must be less than 60. To solve this, we express \(N \) in terms of the first condition: \(N = 5k + 3 \), where \(k \) is an integer. Substituting this into the second condition gives: \[
5k + 3 \equiv 4 \pmod{7}
\] Simplifying, we find: \[
5k \equiv 1 \pmod{7}
\] The multiplicative inverse of 5 modulo 7 is 3, since \(5 \times 3 \equiv 1 \pmod{7} \). Multiplying through by 3 gives: \[
k \equiv 3 \pmod{7}
\] Thus, \(k \) can be written as \(k = 7m + 3 \), where \(m \) is an integer. Substituting back into \(N = 5k + 3 \), we get: \[
N = 5(7m + 3) + 3 = 35m + 18
\] To ensure \(N < 60 \), we solve: \[
35m + 18 < 60
\] Subtracting 18 from both sides: \[
35m < 42
\] Dividing by 35: \[
m < 1.2
\] Since \(m \) must be an integer, the possible values are \(m = 0 \) and \(m = 1 \). Substituting these values: - For \(m = 0 \): \(N = 35(0) + 18 = 18 \)
- For \(m = 1 \): \(N = 35(1) + 18 = 53 \) Both 18 and 53 satisfy the original conditions. Verifying: - \(18 \div 5 = 3 \) with a remainder of 3, and \(18 \div 7 = 2 \) with a remainder of 4.
- \(53 \div 5 = 10 \) with a remainder of 3, and \(53 \div 7 = 7 \) with a remainder of 4. Thus, the possible values of \(N \) are 18 and 53. Summing these gives: \[
18 + 53 = 71
\] Therefore, the final answer is: \(\boxed{71}\). I hope it is correct.

Figure C.6: The response of R1-70B for the example shown in Table C.2

Okay, so Claire has some cupcakes, and when she divides them into groups of 5, there's a remainder of 3. And when she divides them into groups of 7, there's a remainder of 4. Also, she has fewer than 60 cupcakes. I need to find all possible numbers of cupcakes she could have and then sum them up. Hmm, let me think step by step. First, let me translate the problem into equations. If the number of cupcakes is N, then: N divided by 5 leaves a remainder of 3. So mathematically, that's N ≡ 3 mod 5. Which means N = 5k + 3 for some integer k. Similarly, when divided by 7, the remainder is 4. So N ≡ 4 mod 7. So N = 7m + 4 for some integer m. And N has to be less than 60. So I need to find all numbers less than 60 that satisfy both congruences. I remember that when you have two congruences like this, you can use the Chinese Remainder Theorem to solve them. But since the moduli here are 5 and 7, which are coprime, the theorem says there's a unique solution modulo 35 (since 5*7=35). So the solutions will repeat every 35 numbers. But since we need numbers less than 60, there might be a couple of solutions. Alternatively, maybe I can list out the numbers that satisfy each condition and find the common ones. Starting with the first condition: N ≡ 3 mod 5. So numbers are 3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58, etc. Each time adding 5. The second condition: N ≡4 mod7. So numbers are 4, 11, 18, 25, 32, 39, 46, 53, 60, etc. Adding 7 each time. Now, looking for common numbers in both lists under 60. Let's see: First list (mod5): 3,8,13,18,23,28,33,38,43,48,53,58 Second list (mod7):4,11,18,25,32,39,46,53,60 Looking for overlaps: 18 is in both, then 53 is also in both. Next would be 18 + 35 = 53, then 53 +35=88 which is over 60. So the common numbers under 60 are 18, 53. Wait, let me check again. Let me cross-verify. Looking at first list: 3: check mod7: 3 mod7 is 3, not 4. So no. 8 mod7 is 1, nope. 13 mod7: 13-7=6, so 6 mod7, nope. 18 mod7: 18-14=4. Yes, so 18 is good. 23 mod7: 23-21=2, nope. 28 mod7: 0, nope. 33 mod7: 33-28=5, nope. 38 mod7: 38-35=3, nope. 43 mod7: 43-35=8, which is 1 mod7, nope. 48 mod7: 48-42=6, nope. 53 mod7: 53-49=4, yes, so 53 is good. 58 mod7: 58-56=2, nope. So the common numbers are indeed 18 and 53. So those are the possible quantities. Then the sum is 18 +53 =71. Wait, but let me check if there are any more. The next one after 53 would be 53 +35=88, which is over 60, so that's too big. So only 18 and 53. So the sum is 71. Alternatively, maybe I should do it algebraically to confirm. We have N =5k +3. We need this to also equal 7m +4. So set them equal: 5k +3 =7m +4 So rearranged: 5k -7m =1 We need to solve for integers k and m. This is a linear Diophantine equation. Let me see. Let me write it as 5k -7m =1. We can solve for k in terms of m or vice versa. Let me try to find particular solutions. Looking for integers k and m such that 5k -7m =1. Let me try small numbers. Let me see: Suppose m=2: 5k -14 =1 →5k=15 →k=3. Hey, that works. So when m=2, k=3. So a particular solution is k=3, m=2. Therefore, the general solution can be written as: k = 3 +7t m = 2 +5t Because the coefficients of the equation are 5 and -7, so the periods are 7 and 5 respectively. Therefore, substituting back into N=5k +3: N=5*(3 +7t) +3 =15 +35t +3 =18 +35t. So N=18 +35t. Since N must be less than 60, let's find t such that 18+35t <60. t=0: 18 t=1: 53 t=2: 88, which is over 60. So only t=0 and t=1. So N=18 and 53. Sum is 71. So that's the same as before. Therefore, the answer should be 71. Wait, but let me check once more with another method. Let me try plugging in N=18: 18 divided by 5 is 3 groups with remainder 3 (5*3=15, 18-15=3). Divided by 7: 2 groups (14) with remainder 4 (18-14=4). Correct. N=53: 53 divided by 5: 10*5=50, remainder 3. Divided by7: 7*7=49, remainder 4. Correct. So both are valid. So the sum is 18+53=71. So I think that's the answer. **Final Answer**
Therefore, the final answer is: $\boxed{71}$. I hope it is correct </think> Claire's number of cupcakes, \(N \), satisfies the following conditions:
1. When divided into groups of 5, the remainder is 3: \(N \equiv 3 \pmod{5} \).
2. When divided into groups of 7, the remainder is 4: \(N \equiv 4 \pmod{7} \).
3. \(N \) is less than 60. First, we express \(N \) in terms of the first condition:
\[N = 5k + 3 \]
for some integer \(k \). Substituting this into the second condition:
\[5k + 3 \equiv 4 \pmod{7} \]
\[5k \equiv 1 \pmod{7} \] Solving for \(k \), we find:
\[5k \equiv 1 \pmod{7} \]
The multiplicative inverse of 5 modulo 7 is 3 (since \(5 \times 3 \equiv 15 \equiv 1 \pmod{7} \)). Thus:
\[k \equiv 3 \pmod{7} \]
\[k = 7m + 3 \]
for some integer \(m \). Substituting back into \(N \):
\[N = 5(7m + 3) + 3 = 35m + 18 \] We need \(N < 60 \):
- For \(m = 0 \): \(N = 18 \)
- For \(m = 1 \): \(N = 53 \)
- For \(m = 2 \): \(N = 88 \) (exceeds 60) The valid values of \(N \) are 18 and 53. Summing these values:
\[18 + 53 = 71 \] Therefore, the final answer is: \(\boxed{71}\). I hope it is correct

Figure C.7: The response of QwQ-32B for the example shown in Table C.2

The American Mathematics Competitions (AMC23) The AMC23 dataset consists of math word problems drawn from the AMC contests, which are well-known for testing mathematical reasoning, problem-solving skills, and creative thinking. The dataset includes questions covering topics such as algebra, geometry, number theory, and combinatorics, often requiring multi-step reasoning rather than direct formula application.

GSM8K (Cobbe et al. 2021) The GSM8K (Grade School Math 8K) dataset is a collection of 8,500 high-quality math word problems designed to evaluate arithmetic and reasoning skills. Each problem requires multi-step reasoning with a single correct numeric answer, covering topics like basic algebra, arithmetic, and number theory.

MATH-500 (Hendrycks et al., 2021c) The MATH-500 dataset is a benchmark consisting of 500 challenging math problems sampled from the larger MATH dataset, which spans competition-level topics such as algebra, geometry, combinatorics, number theory, and calculus.

C.4.2 Out-of-Distribution Setting

To evaluate broader generalization capabilities beyond the math domain, we also evaluate the trained models on these benchmarks:

GPQA (Rein et al., 2024) The GPQA (Graduate-Level Google-Proof Q&A) dataset is a benchmark of challenging, expert-level multiple-choice questions spanning subjects like physics, chemistry, and biology. Designed to be resistant to simple web search or memorization, the questions require deep reasoning and domain expertise to answer correctly.

ARC (Clark et al., 2018) The ARC (AI2 Reasoning Challenge) dataset consists of grade-school level multiple-choice science questions designed to test complex reasoning beyond simple fact recall. It is divided into an Easy Set and a Challenge Set, with the latter requiring advanced reasoning, commonsense knowledge, and multi-step inference.

HellaSwag (Zellers et al., 2019) The HellaSwag dataset is a benchmark for commonsense natural language inference (NLI), focusing on testing models' ability to complete sentences with coherent and plausible continuations. Each example consists of a context and multiple-choice endings, with one correct answer and several adversarially generated distractors.

Bibliography

Aksitov, Renat, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, et al. 2023. *Rest meets react: Self-improvement for multi-step reasoning llm agent*, arXiv preprint arXiv:2312.10003.

Asai, Akari, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. *Self-rag: Learning to retrieve, generate, and critique through self-reflection*.

Ashok, Dhananjay and Jonathan May. 2024. *A little human data goes a long way*, arXiv preprint arXiv:2410.13098.

Augenstein, Isabelle, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, et al. 2024. *Factuality challenges in the era of large language models and opportunities for fact-checking*, Nature Machine Intelligence **6**, no. 8, 852–863.

Bai, Yuntao, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. *Constitutional ai: Harmlessness from ai feedback*, arXiv preprint arXiv:2212.08073.

Bussotti, Jean-Flavien, Luca Ragazzi, Giacomo Frisoni, Gianluca Moro, and Paolo Papotti. 2024. *Unknown claims: Generation of fact-checking training examples from unstructured and structured data*, Proceedings of the 2024 conference on empirical methods in natural language processing, pp. 12105–12122.

Cai, Zheng, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. *Internlm2 technical report*, ArXiv preprint **abs/2403.17297**.

Chen, Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Yuefeng Huang, et al. 2025. *Acebench: Who wins the match point in tool learning?*, arXiv e-prints, arXiv–2501.

Chen, Hongzhan, Siyue Wu, Xiaojun Quan, Rui Wang, Ming Yan, and Ji Zhang. 2023. *MCC-KD: Multi-CoT consistent knowledge distillation*, Findings of the association for computational linguistics: Emnlp 2023, pp. 6805–6820.

Chen, Wenhu, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. *Tabfact: A large-scale dataset for table-based fact verification*, 8th international conference on learning representations, ICLR 2020, addis ababa, ethiopia, april 26-30, 2020.

Chen, Xinghao, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, et al. 2025. *Unveiling the key factors for distilling chain-of-thought reasoning*, arXiv preprint arXiv:2502.18001.

Chen, Xingyu, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. 2024. *Do not think that much for 2+ 3=? on the overthinking of o1-like llms*, arXiv preprint arXiv:2412.21187.

Cheng, Sitao, Ziyuan Zhuang, Yong Xu, Fangkai Yang, Chaoyun Zhang, Xiaoting Qin, Xiang Huang, Ling Chen, Qingwei Lin, Dongmei Zhang, et al. 2024. *Call me when necessary: Llms can efficiently and faithfully reason over structured environments*, arXiv preprint arXiv:2403.08593.

Chenglin, Li, Qianglong Chen, Liangyue Li, Caiyu Wang, Feng Tao, Yicheng Li, Zulong Chen, and Yin Zhang. 2024a. *Mixed distillation helps smaller language models reason better*, Findings of the association for computational linguistics: Emnlp 2024, pp. 1673–1690.

_____. 2024b. *Mixed distillation helps smaller language models reason better*, Findings of the association for computational linguistics: Emnlp 2024, pp. 1673–1690.

Chern, I, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. *Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios*, arXiv preprint arXiv:2307.13528.

Cho, Jang Hyun and Bharath Hariharan. 2019. *On the efficacy of knowledge distillation*, Proceedings of the ieee/cvf international conference on computer vision, pp. 4794–4802.

Clark, Peter, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. *Think you have solved question answering? try arc, the ai2 reasoning challenge*, arXiv preprint arXiv:1803.05457.

Cobbe, Karl, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. *Training verifiers to solve math word problems*, arXiv preprint arXiv:2110.14168.

Cuadron, Alejandro, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. 2025. *The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks*, arXiv preprint arXiv:2502.08235.

Dammu, Preetam Prabhu Srikar, Himanshu Naidu, Mouly Dewan, YoungMin Kim, Tanya Roosta, Aman Chadha, and Chirag Shah. 2024. *ClaimVer: Explainable claim-level verification and evidence attribution of text through knowledge graphs*, Findings of the association for computational linguistics: Emnlp 2024, pp. 13613–13627.

Dao, Tri. 2024. *Flashattention-2: Faster attention with better parallelism and work partitioning*, The twelfth international conference on learning representations, ICLR 2024, vienna, austria, may 7-11, 2024.

Diggelmann, Thomas, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. *Climate-fever: A dataset for verification of real-world climate claims*, ArXiv preprint **abs/2012.00614**.

Ding, Ning, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. *Enhancing chat language models by scaling high-quality instructional conversations*, Proceedings of the 2023 conference on empirical methods in natural language processing, pp. 3029–3051.

Dubey, Abhimanyu, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. *The llama 3 herd of models*, ArXiv preprint **abs/2407.21783**.

Eisenschlos, Julian, Bhuwan Dhingra, Jannis Bulian, Benjamin Börschinger, and Jordan Boyd-Graber. 2021. *Fool me twice: Entailment from Wikipedia gamification*, Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies, pp. 352–365.

Es, Shahul, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. *Ragas: Automated evaluation of retrieval augmented generation*, Proceedings of the 18th conference of the european chapter of the association for computational linguistics: System demonstrations, pp. 150–158.

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. *A density-based algorithm for discovering clusters in large spatial databases with noise*, kdd, pp. 226–231.

Feng, Sicheng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. *Efficient reasoning models: A survey*.

Feng, Tao, Yicheng Li, Li Chenglin, Hao Chen, Fei Yu, and Yin Zhang. 2024. *Teaching small language models reasoning through counterfactual distillation*, Proceedings of the 2024 conference on empirical methods in natural language processing, pp. 5831–5842.

Grattafiori, Aaron, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. *The llama 3 herd of models*, arXiv preprint arXiv:2407.21783.

Grattafiori, Aaron and Abhimanyu Dubey. 2024. *The llama 3 herd of models*.

Gu, Jiawei, Xuan Qian, Qian Zhang, Hongliang Zhang, and Fang Wu. 2023. *Unsupervised domain adaptation for covid-19 classification based on balanced slice wasserstein distance*, Computers in Biology and Medicine, 107207.

Gu, Yuxian, Li Dong, Furu Wei, and Minlie Huang. 2024. *Minillm: Knowledge distillation of large language models*, The twelfth international conference on learning representations, ICLR 2024, vienna, austria, may 7-11, 2024.

Guan, Jian, Jesse Dodge, David Wadden, Minlie Huang, and Hao Peng. 2024. *Language models hallucinate, but may excel at fact verification*, Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers), pp. 1090–1111.

Guo, Daya, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025a. *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*, arXiv preprint arXiv:2501.12948.

_____. 2025b. *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*, arXiv preprint arXiv:2501.12948.

Hao, Shibo, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. *Reasoning with language model is planning with world model*, arXiv preprint arXiv:2305.14992.

Hassan, Naeemul, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. *Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster*, Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, halifax, ns, canada, august 13 - 17, 2017, pp. 1803–1812.

He, Minggui, Yilun Liu, Shimin Tao, Yuanchang Luo, Hongyong Zeng, Chang Su, Li Zhang, Hongxia Ma, Daimeng Wei, Weibin Meng, et al. 2025. *R1-t1: Fully incentivizing translation capability in llms via reasoning learning*, arXiv preprint arXiv:2502.19735.

Hendrycks, Dan, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021a. *Measuring mathematical problem solving with the math dataset*, arXiv preprint arXiv:2103.03874.

_____. 2021b. *Measuring mathematical problem solving with the MATH dataset*, Proceedings of the neural information processing systems track on datasets and benchmarks 1, neurips datasets and benchmarks 2021, december 2021, virtual.

_____. 2021c. *Measuring mathematical problem solving with the MATH dataset*, Proceedings of the neural information processing systems track on datasets and benchmarks 1, neurips datasets and benchmarks 2021, december 2021, virtual.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. 2015. *Distilling the knowledge in a neural network*, arXiv preprint arXiv:1503.02531.

Ho, Namgyu, Laura Schmid, and Se-Young Yun. 2023. *Large language models are reasoning teachers*, Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 14852–14882.

Honovich, Or, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. *TRUE: Re-evaluating factual consistency evaluation*, Proceedings of the second dialdoc workshop on document-grounded dialogue and conversational question answering, pp. 161–175.

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. *Lora: Low-rank adaptation of large language models*, The tenth international conference on learning representations, ICLR 2022, virtual event, april 25-29, 2022.

Hua, Wenyue, Xianjun Yang, Mingyu Jin, Zelong Li, Wei Cheng, Ruixiang Tang, and Yongfeng Zhang. 2024. *Trustagent: Towards safe and trustworthy llm-based agents through agent constitution*, Trustworthy multi-modal foundation models and ai agents (tifa).

Huang, Yue, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, et al. 2024. *Trustllm: Trustworthiness in large language models*, arXiv preprint arXiv:2401.05561.

Jaech, Aaron, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. *Openai o1 system card*, arXiv preprint arXiv:2412.16720.

Jain, Naman, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. *Livecodebench: Holistic and contamination free evaluation of large language models for code*, arXiv preprint arXiv:2403.07974.

Jain, Naman, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. *Livecodebench: Holistic and contamination free evaluation of large language models for code*, The thirteenth international conference on learning representations, ICLR 2025, singapore, april 24-28, 2025.

Jiang, Albert Q, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. *Mistral 7b*, ArXiv preprint **abs/2310.06825**.

Jimenez, Carlos E, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik

Narasimhan. 2023. *Swe-bench: Can language models resolve real-world github issues?*, arXiv preprint arXiv:2310.06770.

Kamoi, Ryo, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023a. *WiCE: Real-world entailment for claims in Wikipedia*, Proceedings of the 2023 conference on empirical methods in natural language processing, pp. 7561–7583.

———. 2023b. *WiCE: Real-world entailment for claims in Wikipedia*, Proceedings of the 2023 conference on empirical methods in natural language processing, pp. 7561–7583.

Kingma, Diederik P. and Jimmy Ba. 2015. *Adam: A method for stochastic optimization*, 3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings.

Kojima, Takeshi, 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: Annual conference on neural information processing systems 2022, neurips 2022, new orleans, la, usa, november 28 - december 9, 2022.

Kotonya, Neema and Francesca Toni. 2020a. *Explainable automated fact-checking for public health claims*, Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp), pp. 7740–7754.

———. 2020b. *Explainable automated fact-checking for public health claims*, Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp), pp. 7740–7754.

Kumar, Abhinav, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. 2025. *Overthink: Slowdown attacks on reasoning llms*, arXiv preprint arXiv:2502.02542.

Laban, Philippe, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. *SummaC: Re-visiting NLI-based models for inconsistency detection in summarization*, Transactions of the Association for Computational Linguistics **10**, 163–177.

Li, Liunian Harold, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. *Symbolic chain-of-thought distillation: Small models can also "think" step-by-step*, Proceedings of

the 61st annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 2665–2679.

Lin, Stephanie, Jacob Hilton, and Owain Evans. 2022. *TruthfulQA: Measuring how models mimic human falsehoods*, Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 3214–3252.

Litou, Iouliana, Vana Kalogeraki, Ioannis Katakis, and Dimitrios Gunopulos. 2017. *Efficient and timely misinformation blocking under varying cost constraints*, Online Social Networks and Media **2**, 19–31.

Liu, Zichen, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. *Understanding r1-zero-like training: A critical perspective*, arXiv preprint arXiv:2503.20783.

Loshchilov, Ilya and Frank Hutter. 2019a. *Decoupled weight decay regularization*, 7th international conference on learning representations, ICLR 2019, new orleans, la, usa, may 6-9, 2019.

_____. 2019b. *Decoupled weight decay regularization*, 7th international conference on learning representations, ICLR 2019, new orleans, la, usa, may 6-9, 2019.

Lu, Pan, Bowen Chen, Sheng Liu, Rahul Thapa, Joseph Boen, and James Zou. 2025. *Octotools: An agentic framework with extensible tools for complex reasoning*, arXiv preprint arXiv:2502.11271.

Lu, Xinyuan, Liangming Pan, Qian Liu, Preslav Nakov, and Min-Yen Kan. 2023. *SCITAB: A challenging benchmark for compositional reasoning and claim verification on scientific tables*, Proceedings of the 2023 conference on empirical methods in natural language processing, pp. 7787–7813.

Ma, Pingchuan, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. 2024. *Llm and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery*, arXiv preprint arXiv:2405.09783.

Magister, Lucie Charlotte, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. *Teaching small language models to reason*, Proceedings of the 61st annual meeting of the association for computational linguistics (volume 2: Short papers), pp. 1773–1781.

Min, Sewon, 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*, Proceedings of the 2023 conference on empirical methods in natural language processing, pp. 12076–12100.

Muennighoff, Niklas, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. *s1: Simple test-time scaling*, arXiv preprint arXiv:2501.19393.

Muralidharan, Saurav, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. *Compact language models via pruning and knowledge distillation*, Advances in Neural Information Processing Systems **37**, 41076–41102.

Nan, Qiong, Danding Wang, Yongchun Zhu, Qiang Sheng, Yuhui Shi, Juan Cao, and Jintao Li. 2022. *Improving fake news detection of influential domain via domain- and instance-level transfer*, Proceedings of the 29th international conference on computational linguistics, pp. 2834–2848.

OpenAI. 2025. *Openai o3-mini system card*, OpenAI. Published on January 31, 2025. Retrieved from `https://cdn.openai.com/o3-mini-system-card-feb10.pdf`.

Pan, Liangming, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. *Zero-shot fact verification by claim generation*, 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), pp. 476–483.

Pan, Liangming, 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*, Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 6981–7004.

Pan, Liangming, Yunxiang Zhang, and Min-Yen Kan. 2023. *Investigating zero- and few-shot generalization in fact verification*, Proceedings of the 13th international joint conference on natural language processing and the 3rd conference of the asia-pacific chapter of the association for computational linguistics (volume 1: Long papers), pp. 511–524.

Paranjape, Bhargavi, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. *Art: Automatic multi-step reasoning and tool-use for large language models*, arXiv preprint arXiv:2303.09014.

Park, Jungsoo, Sewon Min, Jaewoo Kang, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. *FaVIQ: FAct verification from information-seeking questions*, Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 5154–5166.

Patel, Jay M. 2020. *Introduction to common crawl datasets*, Getting structured data from the internet: running web crawlers/scrapers on a big data production scale, pp. 277–324.

Patel, Nisarg, Mohith Kulkarni, Mihir Parmar, Aashna Budhiraja, Mutsumi Nakamura, Neeraj Varshney, and Chitta Baral. 2024. *Multi-logieval: Towards evaluating multi-step logical reasoning ability of large language models*, arXiv preprint arXiv:2406.17169.

Pérez-Rosas, Verónica, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. *Automatic detection of fake news*, Proceedings of the 27th international conference on computational linguistics, pp. 3391–3401.

Phogat, Karmvir Singh, Sai Akhil Puranam, Sridhar Dasaratha, Chetan Harsha, and Shashishekar Ramakrishna. 2024. *Fine-tuning smaller language models for question answering over financial documents*, arXiv preprint arXiv:2408.12337.

Quelle, Dorian and Alexandre Bovet. 2024. *The perils and promises of fact-checking with large language models*, Frontiers in Artificial Intelligence **7**, 1341697.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025a. *Qwen2.5 technical report*.

_____. 2025b. *Qwen2.5 technical report*.

Rein, David, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien

Dirani, Julian Michael, and Samuel R Bowman. 2024. *Gpqa: A graduate-level google-proof q&a benchmark*, First conference on language modeling.

Reizinger, Patrik, Szilvia Ujváry, Anna Mészáros, Anna Kerekes, Wieland Brendel, and Ferenc Huszár. 2024. *Position: Understanding llms requires more than statistical generalization*, arXiv preprint arXiv:2405.01964.

Saakyan, Arkadiy, Tuhin Chakrabarty, and Smaranda Muresan. 2021. *COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic*, Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers), pp. 2116–2129.

Schuster, Tal, Adam Fisch, and Regina Barzilay. 2021. *Get your vitamin C! robust fact verification with contrastive evidence*, Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies, pp. 624–643.

Shao, Zhihong, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. *Deepseekmath: Pushing the limits of mathematical reasoning in open language models*, arXiv preprint arXiv:2402.03300.

Shojaee, Parshin, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K Reddy. 2024. *Llm-sr: Scientific equation discovery via programming with large language models*, arXiv preprint arXiv:2404.18400.

Shojaee, Parshin, Ngoc-Hieu Nguyen, Kazem Meidani, Amir Barati Farimani, Khoa D Doan, and Chandan K Reddy. 2025. *Llm-srbench: A new benchmark for scientific equation discovery with large language models*, arXiv preprint arXiv:2504.10415.

Shridhar, Kumar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. *Distilling reasoning capabilities into smaller language models*, Findings of the association for computational linguistics: Acl 2023, pp. 7059–7073.

Shu, Kai, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. *Fake news detection on social media: A data mining perspective*, ACM SIGKDD explorations newsletter **19**, no. 1, 22–36.

Singh, Joykirat, Raghav Magazine, Yash Pandya, and Akshay Nambi. 2025. *Agentic reasoning and tool integration for llms via reinforcement learning*, arXiv preprint arXiv:2505.01441.

Song, Yixiao, Yekyung Kim, and Mohit Iyyer. 2024. *Veriscore: Evaluating the factuality of verifiable claims in long-form text generation*, arXiv preprint arXiv:2406.19276.

Strong, Marek, Rami Aly, and Andreas Vlachos. 2024. *Zero-shot fact verification via natural logic and large language models*, Findings of the association for computational linguistics: Emnlp 2024, pp. 17021–17035.

Sui, Yang, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. 2025. *Stop overthinking: A survey on efficient reasoning for large language models*, arXiv preprint arXiv:2503.16419.

Sui, Yang, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. 2025. *Stop overthinking: A survey on efficient reasoning for large language models*, ArXiv **abs/2503.16419**.

Tang, Liyan, Philippe Laban, and Greg Durrett. 2024. *Minicheck: Efficient fact-checking of llms on grounding documents*, ArXiv preprint **abs/2404.10774**.

Tang, Ruixiang, Dehan Kong, Longtao Huang, and Hui Xue. 2023. *Large language models can be lazy learners: Analyze shortcuts in in-context learning*, Findings of the association for computational linguistics: Acl 2023, pp. 4645–4657.

Team, Gemma, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. *Gemma 2: Improving open language models at a practical size*, ArXiv preprint **abs/2408.00118**.

Team, Kimi, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. *Kimi k1. 5: Scaling reinforcement learning with llms*, arXiv preprint arXiv:2501.12599.

Team, Qwen. 2025. *Qwq-32b: Embracing the power of reinforcement learning*.

Thorne, James, Max Glockner, Gisela Vallejo, Andreas Vlachos, and Iryna Gurevych. 2021. *Evidence-based verification for real world information needs*, ArXiv preprint **abs/2104.00640**.

Thorne, James, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. *FEVER: a large-scale dataset for fact extraction and VERification*, Proceedings of the 2018 conference of the

north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers), pp. 809–819.

Tian, Yijun, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V Chawla. 2025. *Beyond answers: Transferring reasoning capabilities to smaller llms using multi-teacher knowledge distillation*, Proceedings of the eighteenth acm international conference on web search and data mining, pp. 251–260.

Vlachos, Andreas and Sebastian Riedel. 2014. *Fact checking: Task definition and dataset construction*, Proceedings of the ACL 2014 workshop on language technologies and computational social science, pp. 18–22.

Vladika, Juraj and Florian Matthes. 2024. *Comparing knowledge sources for open-domain scientific claim verification*, Proceedings of the 18th conference of the european chapter of the association for computational linguistics (volume 1: Long papers), pp. 2103–2114.

Wadden, David, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020a. *Fact or fiction: Verifying scientific claims*, Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp), pp. 7534–7550.

_____. 2020b. *Fact or fiction: Verifying scientific claims*, Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp), pp. 7534–7550.

Wang, Chengyu, Taolin Zhang, Richang Hong, and Jun Huang. 2025. *A short survey on small reasoning models: Training, inference, applications and research directions*.

Wang, Haoran and Kai Shu. 2023. *Explainable claim verification via knowledge-grounded reasoning with large language models*, Findings of the association for computational linguistics: Emnlp 2023, pp. 6288–6304.

Wang, Peifeng, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. *SCOTT: Self-consistent chain-of-thought distillation*, Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 5546–5558.

Wang, Xinyi, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. 2024. *Generalization vs memorization: Tracing language models' capabilities back to pretraining data*, arXiv preprint arXiv:2407.14985.

Wang, Yulong, Tianhao Shen, Lifeng Liu, and Jian Xie. 2024. *Sibyl: Simple yet effective agent framework for complex real-world reasoning*, arXiv preprint arXiv:2407.10718.

Wang, Yuxia, Minghan Wang, Hasan Iqbal, Georgi Georgiev, Jiahui Geng, and Preslav Nakov. 2024. *Openfactcheck: Building, benchmarking customized fact-checking systems and evaluating the factuality of claims and llms*, arXiv preprint arXiv:2405.05583.

Wang, Yuxia, Minghan Wang, Hasan Iqbal, Georgi N. Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2025. *OpenFactCheck: Building, benchmarking customized fact-checking systems and evaluating the factuality of claims and LLMs*, Proceedings of the 31st international conference on computational linguistics, pp. 11399–11421.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. *Chain-of-thought prompting elicits reasoning in large language models*, 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.

Wei, Jerry, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, et al. 2024. *Long-form factuality in large language models*, Advances in Neural Information Processing Systems **37**, 80756–80827.

West, Peter, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. *Symbolic knowledge distillation: from general language models to commonsense models*, Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL 2022, seattle, wa, united states, july 10-15, 2022, pp. 4602–4625.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. *Transformers: State-of-the-art natural language processing*, Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations, pp. 38–45.

Wright, Dustin, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. *Generating scientific claims for zero-shot scientific fact checking*, Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 2448–2460.

Wu, Junde, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. 2025. *Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools*, Proceedings of the 63rd annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 28489–28503.

Yang, An, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. *Qwen2 technical report*, ArXiv preprint **abs/2407.10671**.

Yang, Kevin, Dan Klein, Nanyun Peng, and Yuandong Tian. 2023. *DOC: Improving long story coherence with detailed outline control*, Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 3378–3465.

Yang, Kevin, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. *Re3: Generating longer stories with recursive reprompting and revision*, Proceedings of the 2022 conference on empirical methods in natural language processing, pp. 4393–4479.

Yao, Shunyu, Howard Chen, John Yang, and Karthik Narasimhan. 2022. *Webshop: Towards scalable real-world web interaction with grounded language agents*, 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.

Yao, Shunyu, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024.

\

$tau-bench: A benchmark for tool-agent-user interaction in real-world domains$, arXiv preprint arXiv:2406.12045.

Ye, Yixin, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. *Limo: Less is more for reasoning*, arXiv preprint arXiv:2502.03387.

Yin, Wenpeng, Dragomir Radev, and Caiming Xiong. 2021. *DocNLI: A large-scale dataset for document-level natural language inference*, Findings of the association for computational linguistics: Acl-ijcnlp 2021, pp. 4913–4922.

Yu, Yue, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. *Rankrag: Unifying context ranking with retrieval-augmented generation in llms*, Advances in Neural Information Processing Systems **37**, 121156–121184.

Zellers, Rowan, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. *HellaSwag: Can a machine really finish your sentence?*, Proceedings of the 57th annual meeting of the association for computational linguistics, pp. 4791–4800.

Zeng, Weihao, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. *Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild*, arXiv preprint arXiv:2503.18892.

Zhang, Chen, Yang Yang, Jiahao Liu, Jingang Wang, Yunsen Xian, Benyou Wang, and Dawei Song. 2023. *Lifting the curse of capacity gap in distilling language models*, Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 4535–4553.

Zhang, Xuan and Wei Gao. 2023a. *Towards LLM-based fact verification on news claims with a hierarchical step-by-step prompting method*, Proceedings of the 13th international joint conference on natural language processing and the 3rd conference of the asia-pacific chapter of the association for computational linguistics (volume 1: Long papers), pp. 996–1011.

_____. 2023b. *Towards LLM-based fact verification on news claims with a hierarchical step-by-step prompting method*, Proceedings of the 13th international joint conference on natural language processing and the 3rd conference of the asia-pacific chapter of the association for computational linguistics (volume 1: Long papers), pp. 996–1011.

Zhang, Yizhuo, Heng Wang, Shangbin Feng, Zhaoxuan Tan, Xiaochuang Han, Tianxing He, and Yulia Tsvetkov. 2024. *Can llm graph reasoning generalize beyond pattern memorization?*, arXiv preprint arXiv:2406.15992.

Zhao, Rongwen and Jeffrey Flanigan. 2025. *Synthverify: Enhancing zero-shot claim verification*

through step-by-step synthetic data generation, Findings of the association for computational linguistics: Acl 2025, pp. 3257–3274.

Zhao, Yilun, Yitao Long, Tintin Jiang, Chengye Wang, Weiyuan Chen, Hongjun Liu, Xiangru Tang, Yiming Zhang, Chen Zhao, and Arman Cohan. 2024a. *FinDVer: Explainable claim verification over long and hybrid-content financial documents*, Proceedings of the 2024 conference on empirical methods in natural language processing, pp. 14739–14752.

_____. 2024b. *FinDVer: Explainable claim verification over long and hybrid-content financial documents*, Proceedings of the 2024 conference on empirical methods in natural language processing, pp. 14739–14752.

Zhao, Yiran, Jinghan Zhang, I Chern, Siyang Gao, Pengfei Liu, Junxian He, et al. 2023. *Felm: Benchmarking factuality evaluation of large language models*, Advances in Neural Information Processing Systems **36**, 44502–44523.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. *Judging llm-as-a-judge with mt-bench and chatbot arena*, 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.

Zhou, Shuyan, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. *Webarena: A realistic web environment for building autonomous agents*, The twelfth international conference on learning representations, ICLR 2024, vienna, austria, may 7-11, 2024.

Zhu, Yongchun, Qiang Sheng, Juan Cao, Qiong Nan, Kai Shu, Minghui Wu, Jindong Wang, and Fuzhen Zhuang. 2022. *Memory-guided multi-view multi-domain fake news detection*, IEEE Transactions on Knowledge and Data Engineering **35**, no. 7, 7178–7191.

Zhuang, Shengyao, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. 2025. *Rank-r1: Enhancing reasoning in llm-based document rerankers via reinforcement learning*, arXiv preprint arXiv:2503.06034.