REASONIE: BETTER LLMS FOR SCIENTIFIC INFORMATION EXTRACTION WITH REINFORCEMENT LEARNING AND DATA AUGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) are good at reasoning in math and coding but underperform smaller, supervised models on structured Scientific Information Extraction (SciIE) tasks. This gap rises from a limited domain data and SciIE requires a combination of knowledge memorization and complex reasoning. To bridge this gap, we propose ReasonIE, a novel two-stage training framework. First, we use LLM-driven data augmentation to generate additional domain-specific training data, mitigating data limitation. We then introduce MimicSFT, a supervised fine-tuning method that uses structured reasoning templates to teach logical patterns without human-annotated chains-of-thought, followed by R^{2}GRPO, an RLVR algorithm optimized with a composite reward function that jointly scores factual relevance and logical consistency. Evaluated on SciIE benchmarks, our approach enables a general-purpose Qwen2.5-7B model to become competitive with specialized supervised baselines with less training data, demonstrating that RLVR and LLM-based data augmentation can successfully enhance both the knowledge retention and structured reasoning capacities of LLMs. The implementation is available at: `https://anonymous.4open.science/r/R2GRPO-48B5`

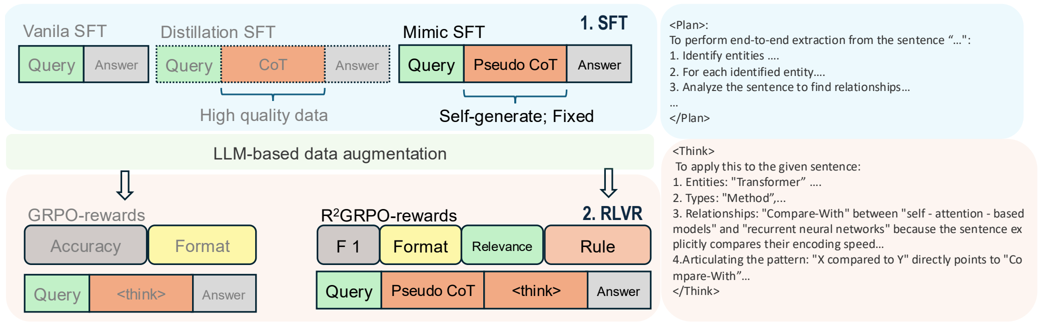
1 INTRODUCTION

Reasoning Large Language Models (LLMs) Guo et al. (2025); Jaech et al. (2024); El-Kishky et al. (2025), trained with advanced post-training methods like Reinforcement Learning from Verifiable Rewards (RLVR), have achieved remarkable success in mathematical reasoning and code generation. However, a significant performance gap remains in structured prediction tasks like Information Extraction (IE), particularly in scientific domains (SciIE). Even state-of-the-art LLMs underperform smaller, supervised BERT-based models Devlin et al. (2019); Beltagy et al. (2019); Yan et al. (2023); Ye et al. (2021); Zhong & Chen (2020) on benchmarks like SciER Zhang et al. (2024).

This gap stems from a fundamental data limitation. Supervised baselines like SciBERT Beltagy et al. (2019) are pretrained on massive, in-domain scientific corpora (e.g., 1.14M papers, 3.1B tokens) and fine-tuned on annotated IE datasets, giving them a strong advantage in domain-specific knowledge memorization. In contrast, general-purpose LLMs are typically applied to SciIE in a low-data regime, with access to only limited training examples (e.g., 5K sentences). SciIE itself demands a hybrid of *knowledge memorization* for precise entity recognition and *contextual rule reasoning* for inferring implicit relations.

However, LLMs' strong reasoning and general natural language procssing ability might help to mitigate the issue of limited domain data in information extraction. Recent studies suggest that RLVR may not impart new reasoning capacities but merely optimize existing output distributions Yue et al. (2025). We posit that SciIE provides an ideal testbed to challenge this view, particularly when combined with data augmentation strategies. LLMs' generative capabilities make them ideal for creating additional training samples for domain-specific IE, helping overcome data scarcity. Furthermore, the current reasoning model are trained for math or coding but these reasoning ability is not good for the information extraction.

Based on this, we propose **ReasonIE**, a two-stage training framework designed to enhance LLMs for IE in data-scarce settings. ReasonIE combines data augmentation with advanced training tech-



<Plan>:
To perform end-to-end extraction from the sentence "…":
1. Identify entities ….
2. For each identified entity….
3. Analyze the sentence to find relationships…
…
</Plan>

<Think>
To apply this to the given sentence:
1. Entities: "Transformer" ….
2. Types: "Method",…
3. Relationships: "Compare-With" between "self - attention - based models" and "recurrent neural networks" because the sentence explicitly compares their encoding speed…
4.Articulating the pattern: "X compared to Y" directly points to "Compare-With"…
</Think>

Figure 1: Our two-stage ReasonIE framework for scientific IE on the left and the sample output of the model. The 'plan' part is for the Pseudo CoT and 'think' part is for the reasoning.

niques: 1) **Data augmentation**: We first leverage LLMs' generative capabilities to create additional domain-specific IE training samples, addressing the data scarcity problem. 2) **MimicSFT** uses structured reasoning templates as prompting strategies during supervised fine-tuning, providing a strong foundation without needing human-annotated chain-of-thought data. 3) **R^{2}GRPO** applies reinforcement learning with a composite reward function combining *relevance* (for factual grounding) and *rule-induced* rewards (for logical consistency).

Our experiments on SciIE benchmarks demonstrate that the ReasonIE framework significantly closes the performance gap with specialized supervised models. A general-purpose Qwen2.5-7B model trained with ReasonIE surpasses other reasoning LLMs and becomes competitive with supervised benchmarks, despite using only the limited SciIE training data. This indicates that RLVR can indeed enhance both the knowledge retention and reasoning capacities of LLMs for complex IE tasks. Our key contributions are:

- We identify and analyze the data disparity challenge in adapting LLMs to SciIE, and propose using LLM-driven data augmentation to generate additional training samples for domain-specific IE.
- We introduce MimicSFT, a method using structured reasoning templates as prompting strategies for effective SFT without high-quality human CoT data.
- We propose R^{2}GRPO, an RLVR method with a novel reward function that jointly optimizes for relevance and rule-based reasoning.

2 Related Work

Post-training Adaptation of LLMs. Large Language Models (LLMs) Achiam et al. (2023); Kaplan et al. (2020) require post-training to align with specific tasks. Common approaches include Supervised Fine-Tuning (SFT) Radford et al. (2018); Brown et al. (2020); Wei et al. (2021); Chung et al. (2024); Zhou et al. (2023) and Reinforcement Learning (RL) Ziegler et al. (2019); Ouyang et al. (2022); Guo et al. (2025). While SFT excels at memorization Chu et al. (2025), RL methods like GRPO Shao et al. (2024) enhance reasoning through self-generated chains Wei et al. (2022). Recent debates question whether RLVR primarily optimizes existing reasoning paths Yue et al. (2025) or learns new patterns Li et al., with effectiveness varying by task type Sprague et al. (2024). Our work examines this SFT-RL distinction in Information Extraction, where both knowledge and reasoning matter.

Scientific Information Extraction (SciIE) aims to extract structured information from unstructured text. Traditional approaches often rely on supervised learning withTransformer-based models (e.g., BERT) trained on domain-specific annotated datasets Devlin et al. (2019). Recently, LLMs have been explored for IE tasks, leveraging their zero-shot Wei et al. (2023); Li et al. (2023); Lu et al. (2023); Xie et al. (2023); Yuan et al. (2023) or in-context learning capabilities Bi et al. (2024); Zhang & Soh (2024); Zhu et al. (2024) or undergoing supervised fine-tuning Wang et al. (2023); Dagdelen et al. (2024); Ning & Liu (2024). While LLMs offer flexibility, they often under-perform specialized supervised models Zhong & Chen (2020); Yan et al. (2023); Ye et al. (2021), Beltagy et al. (2019); Zhang et al. (2024); Dagdelen et al. (2024). Since the training data for these domain is usually limited. Few studies have study how to adapt the reasoning ability of LLMs through post-training in the limited data scenario to improve the performance on SciIE.

3 Methodology

3.1 Problem Formulation

We focus on two fundamental information extraction tasks: Named Entity Recognition (NER) and Relation Extraction (RE).

Named Entity Recognition (NER): Given an input text $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, NER identifies entity spans $e_i = \{x_j, \ldots, x_k\}$ and assigns each a type $t_i \in \mathcal{T}$, where $\mathcal{T}$ is a predefined set of entity types (e.g., Task, Method, Dataset in scientific literature).

Relation Extraction (RE): For a pair of entities (e_i, e_j) identified in text, RE determines whether a relation exists and, if so, classifies it into a relation type $r_{ij} \in \mathcal{R}$, where $\mathcal{R}$ is the set of possible relation types (e.g., Used-For, Compare-With).

End-to-End IE: This combines both tasks, requiring models to first identify entities and then determine relations between them, making it particularly challenging as errors in entity recognition propagate to relation extraction.

Constrained Generation View: We can view IE as a constrained generation problem where the model must generate outputs y that satisfy both:

- *Schema constraints*: Answers must conform to predefined entity and relation types and follow the required structure (e.g. valid json format).
- *Factual constraints*: Answers must come from the original content.

Formally, we can define the constrained generation problem as finding:

$$y^* = \arg\max_{y \in \mathcal{Y}} P(y|x;\theta) \quad \text{s.t.,} \quad C_{\text{schema}}(y) = 1 \wedge C_{\text{factual}}(y, x) = 1, \tag{1}$$

where C_{schema} and C_{factual} are binary constraint functions. This formulation is challenging for standard LLMs as they must simultaneously satisfy structural constraints while maintaining factual accuracy. All fine-tuning is performed using Low-Rank Adaptation (LoRA) (Hu et al., 2022) for computational efficiency.

3.2 Supervised Fine-Tuning and MimicSFT

Standard SFT adapts a pre-trained LLM by maximizing the conditional probability of target outputs given inputs:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{(x,y) \in \mathcal{D}_{\text{SFT}}} \log P(y|x;\theta), \tag{2}$$

where $\mathcal{D}_{\text{SFT}}$ is the supervised fine-tuning dataset. In terms of Equation 11, $o = y$, $\mathcal{D} = \mathcal{D}_{\text{SFT}}$, and $GC(x, y, t, \pi_{\text{ref}}) = 1$ for all tokens.

To improve generalization, we decompose IE into distinct sub-tasks (NER only, RE with Gold Entities, RE only, End-to-End IE) and employ a multi-task learning approach:

$$\mathcal{L}_{\text{MT-SFT}}(\theta) = -\sum_{k=1}^{K} \sum_{(x,y) \in \mathcal{D}_{\text{SFT},T_k}} \log P(y|x, T_k;\theta) \tag{3}$$

where T_k indicates the task type in the prompt.

MimicSFT: Structured Reasoning Without CoT Data. We introduce MimicSFT to encourage structured reasoning without requiring high-quality CoT annotations. The model is trained to produce a templated reasoning block z (enclosed in `<reasoning>...</reasoning>` tags) before generating the final output y:

$$\mathcal{L}_{\text{MimicSFT}}(\theta) = -\sum_{(x,y') \in \mathcal{D}_{\text{MimicSFT}}} \log P(y'|x;\theta), \tag{4}$$

where $y' = (z, y)$ is the concatenation of reasoning steps and final output. The reasoning template follows a general IE process (e.g., 1. Identify entities, 2. Consider relations, 3. Formulate extraction).

3.3 R^{2}GRPO: Reinforcement Learning with Relevance and Rule-Induction

3.3.1 GRPO Framework

R^{2}GRPO builds on Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a PPO variant that normalizes rewards based on group performance. The GRPO objective is:

$$
\begin{aligned}
J_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|q)} \Big[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\mathcal{A}_{\text{clip}}(o_{i,t}, q, \hat{A}_{i,t}) \\
- \beta D_{\text{KL}}(\pi_\theta(\cdot|q, o_{i,<t})||\pi_{\text{ref}}(\cdot|q, o_{i,<t}))\Big],
\end{aligned}
\tag{5}
$$

where: , q is an input prompt from the IE dataset , $\{o_i\}_{i=1}^{G}$ is a group of G outputs sampled from policy $\pi_{\theta_{\text{old}}}$, $\mathcal{A}_{\text{clip}}(o_{i,t}, q, \hat{A}_{i,t}) = \min(r_t(\theta)\hat{A}_{i,t}, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,t})$, $r_t(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the probability ratio , $\hat{A}_{i,t} = \frac{R_i - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$ is the normalized advantage ,β controls the KL divergence penalty from reference policy π_{ref}.

3.3.2 Composite Reward Function

R^{2}GRPO's composite reward function combines four components:

$$
R(o_i, x, y_{\text{gold}}) = w_1 R_{\text{F1}} + w_2 R_{\text{span}} + w_3 R_{\text{relevancy}} + w_4 R_{\text{rule}},
\tag{6}
$$

where w_j are tunable weights. The components are:

F1 Score $R_{\text{F1}}(o_i, y_{\text{gold}}) = \text{F1-score}(o_i, y_{\text{gold}})$ measures extraction accuracy.

Entity Span $R_{\text{span}}(o_i, y_{\text{gold}}) = \frac{1}{N_e}\sum_{j=1}^{N_e} \text{Jaccard}(\text{span}(e_{\text{pred},j}), \text{span}(e_{\text{gold},j}))$ evaluates boundary precision Niwattanakul et al. (2013).

Rule-pattern $R_{\text{rule}}(o_i, x) = \sum_k w_k \cdot \mathbb{I}(\text{pattern}_k \text{ satisfied})$ encourages logical reasoning patterns.

Relevancy $R_{\text{relevancy}}(o_i, x) = \text{Map}(c_i, \text{evidence}_{\text{gold}}) - \lambda_{\text{penalty}} \cdot \left(\frac{|c_i|}{|x|}\right)^2 \cdot \mathbb{I}(|c_i| > \text{threshold})$ promotes evidence-based extraction with length penalty.

The system prompt for R2GRPO training:

System Prompt

Respond in the following format:
<reasoning>
Provide step-by-step reasoning to solve the task based on the given instructions and sentence.
</reasoning>
<thinking>
Cite the specific sentence part (e.g., phrase, verb, or structure) supporting the relation. Articulate a symbolic pattern you discovered (e.g., ”The verb ’achieves’ suggests a Method is applied to a Task, implying a relation”). Explain how this pattern leads to the predicted relation, referencing the relationship definition. Use concise, logical chains (e.g., ”X performs Y → relation Z because of definition”).
</thinking>
<answer>
Provide the final answer in JSON format as specified in the instruction.
</answer>

3.4 Training Strategy

Our training methodology employs a dual-phase approach combining data augmentation with structured reinforcement learning. First, we generate augmented training data by prompting the base LLM to produce both in-domain examples matching the original schema and cross-domain samples from related scientific fields. Each generated instance undergoes rigorous validation for factual accuracy, structural conformity, and diversity, ensuring $\mathcal{D}_{\text{aug}}$ maintains the original data distribution while expanding coverage.

The simplified prompt for data generation:

Data generation prompt

```
You are an expert in scientific data annotation. Your task is to generate new training samples for Named Entity Recognition (NER) and Relation Extraction (RE) tasks in science domain with given entity and relation types.
## Diversity Requirements:
...
## Output Format:
Generate exactly num_to_generate samples in the following JSON format: ”doc_id”: ”UniqueID”, ”sentence”: ”Generated sentence text”, ”ner”: [[”Entity1”, ”Type”], [”Entity2”, ”Type”]], ”rel”: [[”Subject”, ”Relation”, ”Object”]]]
```

For the full version, you can refer to the!A.3. By encouraging the model to generate both in-domain and diverse samples, it could improve the domain adaptation and generalization ability a lot. Different from the math or coding, information extraction data generation is a relatively straightforward tasks for current LLMs.

Our training methodology employs a progressive reinforcement learning approach that systematically increases task complexity. The RL stage begins with low-difficulty samples containing minimal entities and simple relations, allowing the model to establish fundamental extraction patterns. We define difficulty through the relation complexity metric $d(x) = \alpha n_e + \beta n_r$, where n_e counts entities and n_r counts relations per instance. As training progresses, we gradually introduce more challenging samples through an automated curriculum scheduler. Throughout all stages, we maintain distributional alignment with the original task requirements through constrained sampling that preserves the original entity and relation type frequencies.

3.5 Why Structured Reasoning Works

Constraint Satisfaction Through Decomposition. The hierarchical reasoning approach transforms the constrained generation problem into a more tractable form by decomposing it into stages. For MimicSFT with a single reasoning level z_1:

$$P(y|x;\theta) \approx \sum_{z_1 \in \mathcal{Z}_1} P(y|z_1, x;\theta) P(z_1|x;\theta), \tag{7}$$

where $\mathcal{Z}_1$ is the space of valid reasoning templates. This decomposition allows the model to first focus on generating valid reasoning (z_1) that satisfies intermediate constraints before producing the final output (y). For R^{2}GRPO with two reasoning levels:

$$P(y|x;\theta) \approx \sum_{z_1 \in \mathcal{Z}_1} \sum_{z_2 \in \mathcal{Z}_2(z_1)} P(y|z_2, z_1, x;\theta) P(z_2|z_1, x;\theta) P(z_1|x;\theta), \tag{8}$$

where $\mathcal{Z}_2(z_1)$ is the space of valid second-level reasoning conditioned on z_1. This further decomposition allows for more refined constraint satisfaction: z_1 (`<reasoning>...</reasoning>`) establishes the general reasoning framework, addressing schema constraints, z_2 (`<think>...</think>`) refines the reasoning with task-specific details, addressing factual constraints ,y produces the final structured output based on both reasoning levels

3.5.1 Multi-Level Reasoning in R^{2}GRPO

R^{2}GRPO extends MimicSFT by adding a second level of reasoning optimization. If z_1 is the fixed reasoning template (from MimicSFT) and z_2 is the RL-optimized reasoning, the full generation becomes $y' = (z_1, z_2, y)$, creating a hierarchical structure:

Improved Constraint Satisfaction. We can show that this hierarchical approach improves constraint satisfaction probability. Let $\mathcal{C} = \{y : C_{\text{schema}}(y) = 1 \wedge C_{\text{factual}}(y, x) = 1\}$ be the set of outputs satisfying all constraints. The probability of generating a valid output is:

$$P(y \in \mathcal{C}|x;\theta) = \sum_{y \in \mathcal{C}} P(y|x;\theta). \tag{9}$$

For the hierarchical model with reasoning steps z_1 and z_2, we assume that:

$$P(y \in \mathcal{C}|x;\theta_{\text{hier}}) \geq P(y \in \mathcal{C}|x;\theta_{\text{direct}}), \tag{10}$$

when the reasoning steps are optimized to guide the model toward constraint satisfaction. We will verify this later through experiments as shown in Figure 2.

4 Experiments

4.1 Experimental Setup

Training Settings *Base Model* All our fine-tuning experiments are conducted by adapting the Qwen2.5-7B-Instruct model Yang et al. (2024). *SFT (Supervised Fine-Tuning):* Standard fine-tuning on the target IE tasks. *MimicSFT (Multi-Task):* An SFT approach that encourages pseudo-reasoning steps and leverages multi-task learning across different IE sub-tasks (e.g., NER only, RE with Gold Entities, End-to-End IE) as described in Section 3.2 and 3.2. *GRPO-only:* Reinforcement learning using Group Relative Policy Optimization Shao et al. (2024) with a basic F1 score as the reward signal. *R^{2}GRPO:* Our proposed Reinforcement Learning framework, R^{2}GRPO (Relevance and Rule-Induction Group Relative Policy Optimization), incorporating a composite reward function as detailed in Section 3.3. The overall prompt can be seen in the appendix A.3.

Implementation Details All models are fine-tuned using the LoRA approach with a rank of 16 and alpha of 32 for SFT and a rank of 64 and alpha of 128 for R^{2}GRPO, applied to all linear layers in the transformer blocks. For SFT and MimicSFT, we train for 3 epochs with a learning rate of 2×10^{-5} and a batch size of 32 (accumulated over gradient accumulation steps). For R^{2}GRPO, the learning rate for the policy updates is set to 1×10^{-6}. More detail can be found in the appendix.

Evaluation Metrics For Named Entity Recognition (NER) and Relation Extraction (Rel and Rel+), we report the standard micro F1-score. NER: An entity is correct if its span and type match a gold entity. Rel: A relation is correct if the types and spans of both entities and the relation type match a gold relation. Rel+: It further requires the entity type is correct in the triples.

To understand the upperbound and average performance characteristics, especially for RL-finetuned models, we employ metrics analogous to pass@K used in mathematical reasoning. We report: Best F1@K: The best F1 score among K generated outputs for a given input. This helps assess the model's capability to produce a correct extraction within its top K hypotheses. Avg@K: The average F1 score over K generated outputs, providing insight into the general quality and consistency of the model's generations. Unless otherwise specified, K is set to 1 for Best F1@K in main result tables. For the detailed Best F1@K analysis in Section 4.3, we explore a wider range of K values.

For the main results of our models, we set temperature at 0. For the baseline models we use there default setting in their documents. For the Best@K performance to allow better exploration, we set temperature 1.0 for all the compared models. We show more analysis about temperature in the experiment part.

Baseline Models We compare with:

Proprietary or large (>72B) LLMs: regular LLMs like Gemini2.0-flash, DeepSeekV3 and reasoning LLMs like DeepSeek R1,Gemini2.0-flash-thinking;

Small regular LLMs(<=72B): Qwen2.5-7B-Instruct (our base model), Qwen2.5-32B-Instruct, Small reasoning LLMs through distillation(<=72B): deepseek-r1-distill-Qwen2.5-7B, 32B.

Supervised BERT-based models: PURE Zhong & Chen (2020), PL-Marker Ye et al. (2021), HGERE Yan et al. (2023). General-purpose LLMs are evaluated using zero-shot.

Dataset We conduct our experiments primarily on the SciER dataset and OOD datasets Zhang et al. (2024). SciER is a benchmark for information extraction in the scientific domain. It contains 24k

Table 1: Test F1 scores of different baselines on SciER and OOD setting. "Rel" and "Rel+" represent the relation extraction under boundaries and strict evaluation, respectively. ReasonIE is our method with data augmentation and ReasonIE* refer to using test time compute. Our training is based on Qwen2.5-7B-Instruct.

Methods	SciER			OOD		
	NER	Rel	Rel+	NER	Rel	Rel+
Supervised Baselines						
PURE	81.60	53.27	52.67	71.99	50.44	49.46
PL-Marker	83.31	60.06	59.24	73.93	59.02	56.68
HGERE	86.85	62.32	61.10	81.32	61.31	58.32
Zero-Shot LLMs-based Baselines						
DeepSeek-V3	42.45	18.76	18.76	57.40	22.66	22.02
DeepSeek-R1	60.27	27.98	27.16	65.95	32.82	32.25
Gemini2.0	69.85	38.38	38.12	58.53	27.74	26.93
Gemini2.0 thinking	61.43	32.30	31.44	64.75	30.62	30.33
Qwen2.5-32B	56.67	17.10	17.10	36.85	8.72	8.72
DeepSeek-R1-Distill-Qwen-32B	57.63	17.62	17.11	49.00	10.79	9.98
Qwen2.5-7B	41.24	7.09	7.09	44.88	4.20	4.20
DeepSeek-R1-Distill-Qwen-7B	32.01	4.60	4.60	30.25	2.88	2.88
Fine-tuned LLMs						
SFT	80.76	42.22	41.01	70.13	19.45	18.12
GRPO	76.18	48.84	48.02	68.93	42.34	41.76
Ours						
Mimic-SFT	81.70	56.02	55.34	73.71	50.74	49.95
R2GRPO	77.55	54.59	53.65	70.05	45.72	44.67
ReasonIE(w/o aug)	84.36	66.81	65.95	77.84	55.08	54.29
ReasonIE	85.18	67.68	66.82	83.24	63.87	61.90
ReasonIE*	**89.72**	**75.48**	**74.83**	**85.02**	**69.94**	**67.51**

entities and 12k relations over 106 scientific publications. It features diverse entity types (e.g., Task, Method, Datasets) and relation types (e.g., Used-For, Compare-with, Feature-Of, Evaluate-with, ...). We use the standard splits for training. The detail dataset statistics can is shown in Table 3.

We generate 2K data as data augmentation with the same backbone model.

4.2 Main Results

We present the overall performance for Named Entity Recognition (NER) and End-to-End Relation Extraction (Rel) on the SciER test set and OOD dataset in Table 1.

The results in Table 1 show that our ReasonIE boosts the performance of Qwen2.5-7B-Instruct on SciER and OOD for both NER and RE tasks significantly. Especially on Relation extraction in SciER, it outperforms all the supervised baselines. Mimic-SFT achieve higher relation extraction score than SFT one show the pseudo CoT can activate model's 'reasoning' ability or constrained generation refine the reasoning path. Not we use 0 temperature for our models and default setting for the baseline models. For the results for supervised baselines, we use the reported results from the original paper Zhang et al. (2024). Similarly, R^2GRPO outperform GRPO is this case. MimicSFT also shows strong performance, often outperforming standard SFT, highlighting the benefit of the proposed structured pseudo-reasoning. For test-time-compute enhancement, we select the best of 5 results with temperature 1.

4.3 What Do Reasoning Models Learn? Analysis of Best F1@K

To delve deeper into what reasoning models learn, particularly through Reinforcement Learning with Verifiable Rewards (RLVR) like R^2GRPO, we analyze the Best F1@K performance. We selected a subset of 50 challenging samples from the SciER test set and evaluated model outputs with K values ranging from 1, 4, 16, 32, 64, 128, 512, up to 1024. This analysis aims to understand the upper-bound

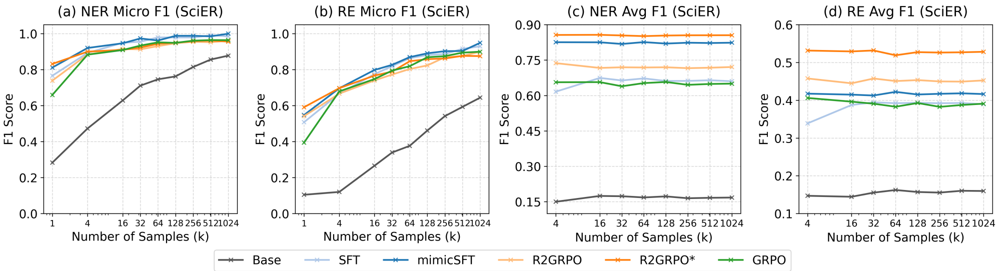


Figure 2: Best F1@K scores representing the reasoning capacity and Avg@K scores representing the reasoning ability for NER and RE on SciER (small). R2GRPO* is the ReaonIE without data augmentation.

capabilities of the models and how SFT and RLVR shape their knowledge and reasoning. The results for NER and RE are visualized in Figure 2.

RLVR and SFT both Enhance Reasoning Capacity: From Figure 2, both RLVR-based (GRPO, R^{2}GRPO) and SFT-based models (SFT, MimicSFT) significantly outperform the base Qwen2.5-7B-Instruct model across all K values. This contradicts the hypothesis that RLVR merely optimizes path selection without improving underlying capabilities Yue et al. (2025). Instead, our results demonstrate that both SFT and RLVR enable models to acquire domain-specific knowledge and enhance reasoning capabilities relevant to IE tasks. The consistent improvement in Best F1@K scores, even at large K values, indicates a genuine expansion of the model's knowledge boundaries rather than just better prioritization of existing knowledge.

Hierarchical Reasoning Improves Knowledge Integration: MimicSFT consistently outperforms standard SFT, and similarly, R^{2}GRPO outperforms basic GRPO across both in-domain and OOD settings. This validates our theoretical analysis in Section 3.5 that structured decomposition of reasoning facilitates better constraint satisfaction. The templated reasoning approach creates intermediate representations that guide the model toward valid outputs, effectively narrowing the search space while maintaining exploration capabilities. This verify our assumption in Eq. 10.

Complementary Effects of SFT and RLVR: While SFT-based models (particularly MimicSFT) achieve slightly higher Best F1@K at very large K values, RLVR models demonstrate superior Avg@K and Best F1@1 scores. This reveals a fundamental trade-off: SFT expands the model's knowledge boundaries, while RLVR optimizes the probability distribution to prioritize high-quality outputs. The combination in R^{2}GRPO* achieves the best of both worlds—maintaining high Best F1@K (knowledge breadth) while significantly improving Best F1@1 (practical performance). **Structured Reasoning Enhances Generalization:** The performance gap between our methods and baselines widens in OOD settings, demonstrating that hierarchical reasoning improves generalization. This aligns with our theoretical framework—by decomposing complex extraction tasks into structured sub-problems, models learn more generalizable patterns rather than memorizing specific input-output mappings. The structured attention allocation mechanism described in Section 3.5 enables more effective feature extraction across different domains.

Exploration-Exploitation Balance: The slightly lower Best F1@K of R^{2}GRPO compared to MimicSFT at very large K values reflects an intentional trade-off. R^{2}GRPO optimizes for high-reward trajectories within a practical exploration budget, focusing computational resources on promising reasoning paths. This is particularly valuable in real-world applications where generating hundreds of candidates is impractical. The higher Avg@K scores of R^{2}GRPO indicate more consistent performance across generations, making it more reliable in production environments.

4.3.1 Ablation Studies and Parameter Sensitivity

Impact of Data Augmentation: Table 1 reveals the significant benefits of our data augmentation strategy, particularly for out-of-domain (OOD) generalization. The performance improvement can be attributed to two key factors: Domain Diversity: Our cross-domain augmentation generates samples covering broader scientific disciplines (biology, chemistry, physics) compared to the original computer science-focused training sets. Structural Variation: The augmented data contains varied sentences for similar entities.

Temperature Sensitivity: Figure 3 shows that our models consistently outperform baselines across different temperature settings. The optimal performance is at lower temperatures (¡0.6). This indicates that SciIE benefits from more deterministic generation since the task requires precise entity boundary detection and relation inference based on the content. The performance degradation at higher temperatures suggests that excessive exploration introduces more noise in the structured extraction process. We also found that the completion length increases with higher temperature in Figure 6a. This suggests the thinking length increase with the noise that leads to unstable thinking content can harm the performance.

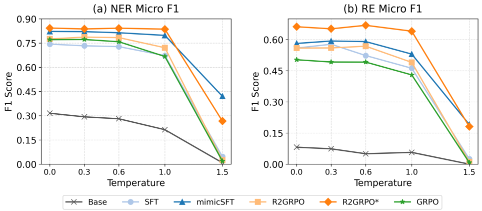
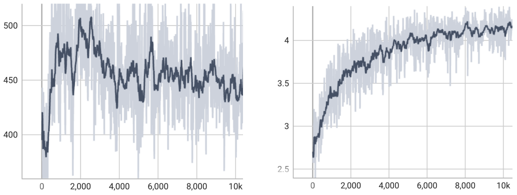


Figure 3: Performance v.s temperature. R2GRPO* is the ReaonIE without data augmentation.

Response Length Analysis: In Figure 5, as training goes, the response length first increase than decrease. This differs from the results of Shao et al. (2024) where response length increase. Figure 4 reveals an interesting relationship between response length and performance. The performance does not generally benefit from longer response. The long response length at high temperature(1.5) leads to bad performance. This suggest for tasks like SciIE, the constrained reasoning process is better than the long but noisy think content. This supports our assumption—effective IE requires concise, targeted reasoning that focuses on relevant constraints rather than exhaustive exploration. The hierarchical reasoning approach in R^{2}GRPO guides the model to generate more efficient reasoning paths, avoiding unnecessary elaboration while maintaining extraction accuracy. More training detail is in Figure 6.



(a) Response length

(b) Reward

Figure 5: Response length(a) and Reward(b) v.s. training steps for R^{2}GRPO

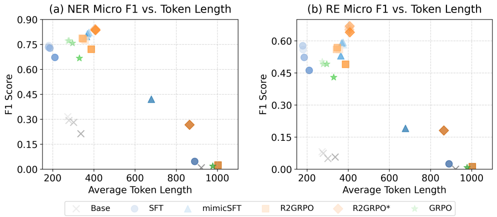


Figure 4: Response token length. The deeper the color, the higher the temperature.R2GRPO* is the ReaonIE without data augmentation.

4.4 Efficiency Comparison

While ReasonIE is based on a 7B parameter model, our approach demonstrates superior efficiency, processing sentences more than 2× faster than HGERE (a 0.4B model) on the SciER dataset as shown in Table 2. HGERE's processing time per entity becomes particularly time-consuming for long sentences with multiple entities, as noted in their original paper Yan et al. (2023). In contrast, our method simply prompts the trained LLM, maintaining consistent processing speed as long as the input remains within the token length limit.

Table 2: Efficiency

Model	**Speed (sent/s)**
ReasonIE	13.6
HGERE	5.2

5 Limitation and Future Work

Our study mainly focuses on the SciIE. The effectiveness of MimicSFT's pseudo-reasoning templates and R^{2}GRPO might vary across different types of information extraction tasks or languages. In the future, we would explore the adaptability of R^{2}GRPO to broader domains and investigate more automated methods for generating or refining reasoning templates. Moreover, further research can also focus on how structured reasoning influences knowledge acquisition and path selection in diverse LLM architectures and explore the scalability of our approach to even larger models or datasets of different domains.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. Codekgc: Code language model for generative knowledge graph construction. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(3):1–16, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, et al. Competitive programming with large reasoning models. *arXiv preprint arXiv:2502.06807*, 2025.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhamaneshi, Shishir G Patil, Matei Zaharia, et al. Llms can easily learn to reason from demonstrations structure, not content, is what matters!, 2025. *URL https://arxiv. org/abs/2502.07374*.

Guozheng Li, Peng Wang, and Wenjun Ke. Revisiting large language models as zero-shot relation extractors. *arXiv preprint arXiv:2310.05028*, 2023.

Jiaxiang Li, Siliang Zeng, Hoi-To Wai, Chenliang Li, Alfredo Garcia, and Mingyi Hong. Getting more juice out of the sft data: Reward learning from human demonstration improves sft for llm alignment. *Advances in Neural Information Processing Systems*, 37:124292–124318, 2024.

Keming Lu, Xiaoman Pan, Kaiqiang Song, Hongming Zhang, Dong Yu, and Jianshu Chen. Pivoine: Instruction tuning for open-world information extraction. *arXiv preprint arXiv:2305.14898*, 2023.

Yansong Ning and Hao Liu. Urbankgent: A unified large language model agent framework for urban knowledge graph construction. *arXiv preprint arXiv:2402.06861*, 2024.

Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pp. 380–384, 2013.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

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

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*, 2023.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

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

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. Zero-shot information extraction via chatting with chatgpt. *arXiv e-prints*, pp. arXiv–2302, 2023.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Empirical study of zero-shot ner with chatgpt. *arXiv preprint arXiv:2310.10035*, 2023.

Zhaohui Yan, Songlin Yang, Wei Liu, and Kewei Tu. Joint entity and relation extraction with span pruning and hypergraph neural networks. *arXiv preprint arXiv:2310.17238*, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. Packed levitated marker for entity and relation extraction. *arXiv preprint arXiv:2109.06067*, 2021.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. Zero-shot temporal relation extraction with chatgpt. *arXiv preprint arXiv:2304.05454*, 2023.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Bowen Zhang and Harold Soh. Extract, define, canonicalize: An llm-based framework for knowledge graph construction. *arXiv preprint arXiv:2404.03868*, 2024.

Qi Zhang, Zhijia Chen, Huitong Pan, Cornelia Caragea, Longin Jan Latecki, and Eduard Dragut. Scier: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents. *arXiv preprint arXiv:2410.21155*, 2024.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zexuan Zhong and Danqi Chen. A frustratingly easy approach for joint entity and relation extraction. *arXiv preprint arXiv:2010.12812*, 2020.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *World Wide Web*, 27(5):58, 2024.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Appendix

A.1 Dataset Statistics

We show here the detail statistic of the datasets used. For the Best@K evaluation, we select 50 samples from both the SciER and OOD test set. And for the rest evaluation we use the full datasets.

A.2 R^{2}GRPO Training

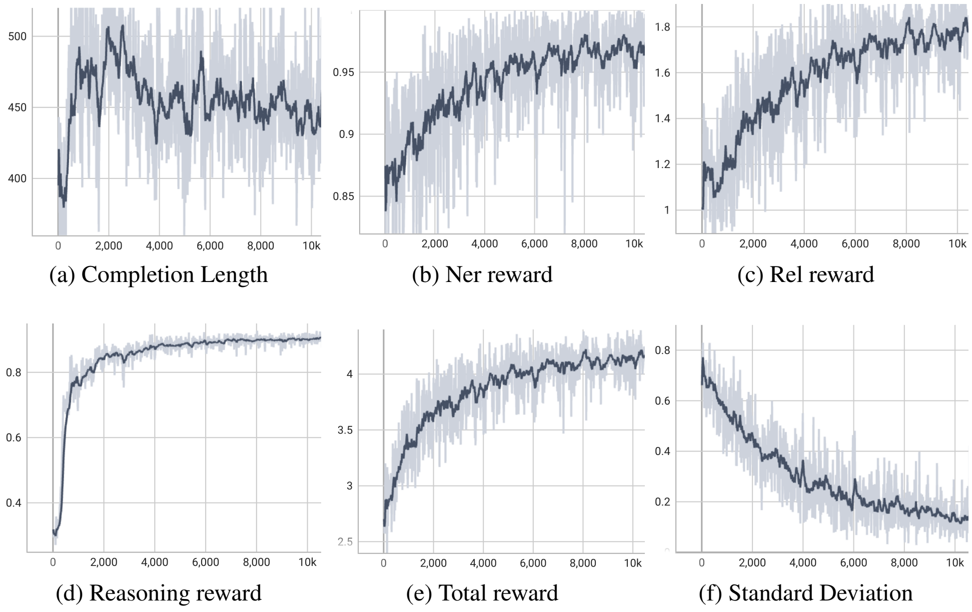
Unified Gradient Framework Both SFT and RL update model parameters θ by following a gradient. Following Shao et al. (2024), we conceptualize these post-training algorithms under a unified gradient expression:

$$\nabla_\theta J(\theta) = \mathbb{E}_{(x,o)\sim\mathcal{D}} \left[\frac{1}{|o|} \sum_{t=1}^{|o|} GC(x, o, t, \pi_{\text{ref}}) \nabla_\theta \log \pi_\theta(o_t|x, o_{<t}) \right], \tag{11}$$

where (x, o) is an input-output pair from distribution $\mathcal{D}$, $\pi_\theta(o_t|x, o_{<t})$ is the probability of generating token o_t given input x and previous tokens $o_{<t}$, $GC(x, o, t, \pi_{\text{ref}})$ is the gradient coefficient determining update magnitude and direction, and $\mathcal{D}$ represents the data source (human-annotated for SFT, model-generated for RL).

Table 3: Detail Distribution of Datasets

Entity/Relation Type	**Train**	**Dev**	**SciER Test**	**OOD Test**	**Total**
		Entity Types			
`Method`	11424	1549	1890	1018	15881
`DATASET`	3220	269	370	83	3942
`TASK`	3397	416	688	194	4695
Total	18041	2234	2948	1295	24518
		Relation Types			
`PART-OF`	1865	214	304	111	2494
`USED-FOR`	2398	343	546	167	3454
`EVALUATED-WITH`	863	78	131	49	1121
`SYNONYM-OF`	880	76	170	89	1215
`COMPARE-WITH`	875	175	114	54	1218
`SUBCLASS-OF`	697	114	176	73	1060
`BENCHMARK-FOR`	551	64	85	28	728
`SUBTASK-OF`	210	31	65	9	315
`TRAINED-WITH`	404	37	35	2	478
Total	8743	1132	1626	582	12083



(a) Completion Length

(b) Ner reward

(c) Rel reward

(d) Reasoning reward

(e) Total reward

(f) Standard Deviation

Figure 6: R^2GRPO training detail v.s. steps

For SFT, $GC = 1$ for all tokens in the target sequence, while for RL methods like GRPO, GC is derived from reward signals and advantage estimates. Based on this, both SFT and GRPO can update the model parameter based on the data. Since GRPO is can refine the reasoning path and SFT(with distillation) can improve the reasoning capacity. SFT should also be able to refine the reasoning process in a simple way. And GRPO should also be able to improve the reasoning capacity and memorize knowledge from the input data. Our method is one step towards this.

We trained the model on a sub set of the SciER with 1K sample for RL. The selection is based on the balanced distribution of the entity and relation of different types and samples with different length total number of entities and relation triples. The training detail is shown in Fig. 6.

Training can be done within 24GB vram with the lora adapter. However, larger group size require larger vram We train for 3 epochs on the full SciER datasets and 10 epochs for the RL on the 1K subset.

A.3 Prompt

We adapt the instruction from SciER Zhang et al. (2024).

Ner Background

Extract specific entities from the following sentence. The entities to be identified are: 'Dataset', 'Task', and 'Method'.
Entity Definitions:
- 'Task': A task in machine learning refers to the specific problem or type of problem that a ML/AI model/method is designed to solve. Tasks can be broad, like classification, regression, or clustering, or they can be very specific, such as Pedestrian Detection, Autonomous Driving, Sentiment Analysis, Named Entity Recognition, and Relation Extraction.
- 'Method': A method entity refers to the approach, algorithm, or technique used to solve a specific task/problem. Methods encompass the computational algorithms, model architectures, and the training procedures that are employed to make predictions or decisions based on data. For example, Convolutional Neural Networks, Dropout, data augmentation, recurrent neural networks.
- 'Dataset': A realistic collection of data that is used for training, validating, or testing the algorithms. These datasets can consist of various forms of data such as text, images, videos, or structured data. For example, MNIST, COCO, AGNews, IMDb.

Other Notes: - Generics cannot be used independently to refer to any specific entities, e.g., 'This task', 'the dataset', and 'a public corpus' are not entities.
- The determiners should not be part of an entity span. For example, given span 'the SQuAD v1.1 dataset', where the determiner 'the' should be excluded from the entity span.
- If both the full name and the abbreviation are present in the sentence, annotate the abbreviation and its corresponding full name separately. For instance, '20-newsgroup (20NG)'.
- Only annotate "factual, content-bearing" entities. Task, dataset, and method entities normally have specific names and their meanings are consistent across different papers. For example, "CoNLL03", "SNLI" are factual entities. Annotators should annotate only the minimum necessary to represent the original meaning of task/dataset/metric (e.g., "The", "dataset", "public", 'method', 'technique' are often omitted).

Based on the given sentence and the entities with their types, determine the relationship between each pair. The potential relations are: ['Part-Of', 'SubClass-Of', 'SubTask-Of', 'Benchmark-For', 'Trained-With', 'Evaluated-With', 'Synonym-Of', 'Used-For', 'Compare-With']. If no relationship exists between a pair, do not include it in the output.

Rel Background

```
### Relationship Definitions:
- ’Part-Of’: This relationship denotes that one entity (e.g., a Method) is a component or a part of another entity (e.g., another Method).
- ’SubClass-Of’: Specifies that one entity is a subclass or a specialized version of another entity.
- ’SubTask-Of’: Indicates that one Task is a subset or a specific aspect of another broader Task.
- ’Benchmark-For’: Shows that a Dataset serves as a standard or benchmark for evaluating the performance of a Method on a Task.
- ’Trained-With’: Indicates that a Method is trained using a Dataset.
- ’Evaluated-With’: This relationship denotes that a Method is evaluated using a Dataset to test its performance or conduct experiments.
- ’Synonym-Of’: Indicates that two entities are considered to have the same or very similar meaning, such as abbreviations.
- ’Used-For’: Shows that one entity (e.g., a Method) is utilized for achieving or performing another entity (e.g., a Task). This relationship is highly flexible.
- ’Compare-With’: This relationship is used when one entity is compared with another to highlight differences, similarities, or both.
### Notes:
- Determine the ’Relationship’ that best describes how the subject and object are related, based on the sentence context.
- Please do not annotate negative relations (e.g., X is not used in Y).
- Annotate a relationship only if there is direct evidence or clear implication in the text. Avoid inferring relationships that are not explicitly mentioned or clearly implied.
```

Task

```
Given the sentence: ”sentence”
Extract entities and their relations.
### Instruction:
- Think step-by-step to identify entities (’Dataset’, ’Task’, ’Method’) and their relationships.
- Return the results in JSON format with:
- ”ner”: a list of [entity, type] pairs.
- ”rel”: a list of [subject, relation, object] triples.
```

In general, the prompt consists of the background definition of the entity, relation and the instruction on the tasks.

Data generation prompt:

Data generation prompt

```
You are an expert in scientific data annotation. Your task is to generate new training samples for Named Entity Recognition (NER) and Relation Extraction (RE) tasks in science domain.
## Entity Definitions:
- ’Task’: A task refers to the specific problem or type of problem that a scientific model/method is designed to solve.
- ’Method’: A method entity refers to the approach, algorithm, or technique used to solve a specific task/problem.
- ’Dataset’: A realistic collection of data that is used for training, validating, or testing the algorithms/method.

## Relationship Definitions:
- ’Part-Of’: One entity is a component or part of another entity.
- ’SubClass-Of’: One entity is a subclass or specialized version of another.
- ’SubTask-Of’: One Task is a subset or specific aspect of another broader Task.
- ’Benchmark-For’: A Dataset serves as a standard for evaluating a Method on a Task.
- ’Trained-With’: A Method is trained using a Dataset.
- ’Evaluated-With’: A Method is evaluated using a Dataset.
- ’Synonym-Of’: Two entities have the same or very similar meaning.
- ’Used-For’: One entity is utilized for performing another entity.
- ’Compare-With’: One entity is compared with another.

## Annotation Guidelines:
- Only annotate factual, content-bearing entities with specific names
- Exclude determiners from entity spans
- Annotate both full names and abbreviations separately
- Only annotate relationships with direct evidence in the text

## Diversity Requirements:
- Use varied sentence structures (simple, compound, complex)
- Cover different scientific domains
- Include different types of entities (acronyms, full names, combinations)
- Vary the position of entities in sentences
- Use different verb tenses and voices (active/passive)
- Use COMPLETELY DIFFERENT entity names from the examples
- Vary sentence structures significantly
- Include different combinations of entity types

## Output Format:
Generate exactly num_to_generate samples in the following JSON format: ”doc_id”: ”UniqueID”, ”sentence”: ”Generated sentence text”, ”ner”: [[”Entity1”, ”Type”], [”Entity2”, ”Type”]], ”rel”: [[”Subject”, ”Relation”, ”Object”]]]
```

A.4 LLM USAGE

We use LLM to aid writing including polishing and grammar check.