ORACLE: Optimizing Reasoning Abilities of Large Language Models via Constraint-Led Synthetic Data Elicitation

Zhuojie Yang[1*], **Wentao Wan**[1*], **Keze Wang**[1,2,3†]

[1]School of Computer Science and Engineering, Sun Yat-sen University
[2]Peng Cheng Laboratory
[3]Guangdong Key Laboratory of Big Data Analysis and Processing
{yangzhj53, wanwt3}@mail2.sysu.edu.cn, wangkeze@mail2.sysu.edu.cn

Abstract

Training large language models (LLMs) with synthetic reasoning data has become a popular approach to enhancing their reasoning capabilities, while a key factor influencing the effectiveness of this paradigm is the quality of the generated multi-step reasoning data. To generate high-quality reasoning data, many recent methods generate synthetic reasoning paths and filter them based on final answer correctness, often overlooking flaws in intermediate reasoning steps. To enhance the verification of intermediate reasoning steps, prior work primarily resorts to code execution or symbolic reasoning engines. However, code-based validation is restricted to code or mathematical tasks, and reasoning engines require a well-structured and complete context. As a result, existing methods fail to function effectively in natural language reasoning tasks that involve ambiguous or incomplete contexts. In these tasks, synthetic data still lack reliable checks for verifying each reasoning step. To address this challenge, we introduce **ORACLE**, a structured data generation framework inspired by syllogistic reasoning. ORACLE integrates the generative strengths of LLMs with symbolic supervision: the LLM produces step-wise reasoning contexts, while a symbolic reasoning engine verifies the validity of each intermediate step. By employing a unified prompting template to elicit modular reasoning chains, ORACLE enables fine-grained, step-level validation, facilitating the construction of high-quality multi-step reasoning data. Across six logical, factual, and commonsense reasoning benchmarks, our ORACLE consistently outperforms strong baselines on multiple models.

Code — https://github.com/yangzhj53/ORACLE

Introduction

Training large language models (LLMs) (Chang et al. 2024a,b; Hadi et al. 2023; Liu et al. 2024) with synthetic reasoning data has emerged as a widely adopted paradigm for improving their performance on complex reasoning tasks such as multi-hop question answering, mathematical problem solving, and scientific inquiry(Zelikman et al. 2022; Yuan et al. 2023, 2024). This approach leverages the generative capabilities of LLMs to scale up the creation of multi-step reasoning examples, which are otherwise expensive and labor-intensive to annotate manually. To ensure correctness, recent methods (Madaan et al. 2023; Chen et al. 2023) commonly generate multiple synthetic reasoning paths and select those that yield the correct final answer. This answer-based filtering mechanism provides a scalable proxy for ground truth supervision, yet it suffers from a critical limitation: it implicitly assumes that a correct answer is indicative of a valid reasoning process. In practice, this assumption frequently breaks down, as models may arrive at the right answer via spurious, shortcut-driven, or logically invalid intermediate steps (Shum, Diao, and Zhang 2023; Bai et al. 2022). Such flawed reasoning paths, though superficially accurate, can propagate latent logical inconsistencies and reinforce brittle heuristics within the model.

Several efforts have been made to improve the quality of synthetic reasoning data through more rigorous verification. Code-based checking has been successfully applied in mathematical and code domains, where reasoning steps can be translated into executable programs (Lewkowycz et al. 2022; Biderman et al. 2023). Yet this technique falls short in natural language reasoning tasks that involve abstract semantics, ambiguous phrasing, or commonsense knowledge. Alternatively, symbolic reasoning engines offer a promising route for validating logical consistency by translating natural language into formal logic representations (Creswell, Shanahan, and Higgins 2023; Liang et al. 2021; Jiang, Fonseca, and Cohen 2024). Despite their precision, they depend heavily on sufficiently complete and unambiguous context, making them brittle in open-domain or under-specified scenarios. For natural language reasoning tasks that do not have complete and standardized context, there is currently no good method for intermediate process verification of synthetic data.

In this work, we focus on improving the quality of synthetic reasoning data under context-limited natural language reasoning settings, where traditional verification methods struggle. Inspired by classical syllogistic reasoning (Wan et al. 2025; Constant 2024), we propose a novel data generation framework that combines the generative power of LLMs with the precision of symbolic logic engines. LLMs possess rich world knowledge and the ability to actively identify reasoning directions, which provide the necessary contextual information for the engine. Meanwhile, the reasoning engine performs automated inference based on given premises and

[*]These authors contributed equally.
[†]Corresponding author.

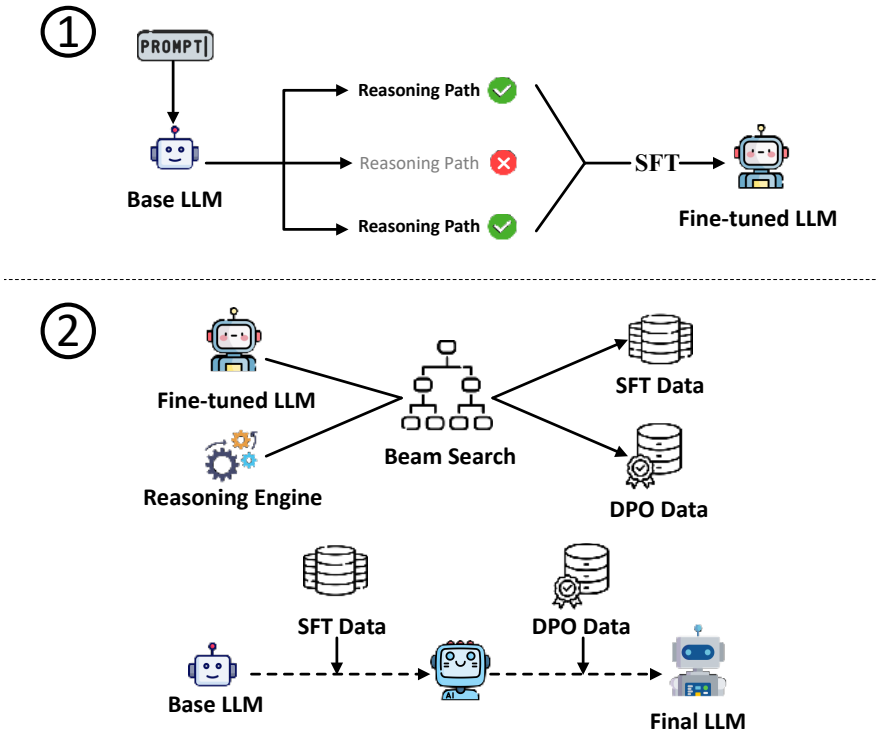

Figure 1: An overview of two-stage training pipeline of our ORACLE. ORACLE adopts a two-stage training pipeline: Stage 1 employs few-shot prompting and template-based reasoning generation, followed by answer-based and format-based filtering; Stage 2 integrates symbolic reasoning via beam search to produce high-quality reasoning data for supervised fine-tuning(SFT) and Direct Preference Optimization(DPO).

rules, ensuring logical rigor (Rabe et al. 2021; Liang et al. 2017). We aim to leverage the complementary strengths of LLMs and reasoning engines through fine-grained interactions to generate high-quality reasoning data.

Specifically, in this work, we propose ORACLE, a structured reasoning data generation framework that combines template-guided step-by-step generation with symbolic supervision to enable both interpretability and verifiability across multi-step chains of thought. ORACLE employs a unified response template that guides LLMs in discrete, modular steps—each comprising a clearly defined query, relevant facts, applied rules, and a revision process. These structured outputs not only facilitate automatic extraction and fine-grained analysis but also enable seamless integration with symbolic reasoning engines as external verifiers.

Our ORACLE consists of a two-stage training process. In the first stage, we use a small set of manually crafted demonstrations to bootstrap the generation of large-scale structured reasoning data via few-shot prompting. After rigorous filtering, this data is used to fine-tune the base model to internalize the step-wise reasoning format. In the second stage, we combine reasoning engine verification and large model evaluation to generate supervised fine-tuning data and preference data through a beam search strategy.

We conducted comprehensive experiments on six diverse reasoning datasets, covering symbolic, factual and commonsense tasks, using LLaMA, Mistral, and Qwen (Touvron et al. 2023; Jiang et al. 2023; Bai et al. 2023). Our ORACLE consistently achieves the best or near-best performance across all datasets and model variants, surpassing strong baseline methods. These results underscore the effectiveness and broad applicability of our ORACLE in enhancing the reasoning capabilities of large language models.

Related Work

Synthetic Reasoning Data Generation. To overcome the scarcity of high-quality reasoning data, numerous studies have explored leveraging large language models (LLMs) themselves to generate synthetic datasets (Zelikman et al. 2022; Yuan et al. 2023, 2024). A common strategy involves self-augmented prompting, where models generate chain-of-thought (CoT) explanations alongside answers, followed by filtering based on answer correctness. While effective in scaling up data, this method often retains reasoning paths that are spuriously correct, leading to models that memorize artifacts instead of acquiring genuine reasoning capabilities (Turpin et al. 2023; Zelikman et al. 2022). Our ORACLE addresses this issue by supervising the entire reasoning chain, not just the final answer, thereby mitigating shortcut learning.

Verification and Programmatic Supervision. Existing approaches to improving the quality of synthetic reasoning data often rely on programmatic supervision to verify intermediate steps. In mathematical and algorithmic domains, code-based execution offers an effective means of validation by aligning reasoning steps with executable semantics (Lewkowycz et al. 2022; Biderman et al. 2023; Leang et al. 2025). However, such methods are inherently domain-specific and fall short when applied to natural language reasoning tasks involving abstract concepts, ambiguous expressions, or commonsense knowledge. Alternatively, symbolic reasoning engines have been explored to assess logical consistency by converting natural language statements into formal logic representations (Creswell, Shanahan, and Higgins 2023; Jiang, Fonseca, and Cohen 2024; Kamoi et al. 2024). Despite their precision, these systems are fragile in practice, often requiring complete and unambiguous context that is difficult to obtain in open-domain scenarios. In contrast, ORACLE incorporates symbolic supervision in a soft and LLM-compatible manner by structuring reasoning into discrete, interpretable modules. This hybrid design facilitates automatic verification while preserving adaptability to diverse reasoning domains, thereby addressing key limitations of existing verification techniques.

LLM Supervision and Preference Training. Fine-tuning LLMs with high-quality, structured supervision has proven effective in aligning model outputs with human reasoning patterns (Ouyang et al. 2022; Rafailov et al. 2023). Beyond supervised fine-tuning, preference training via comparison data (Ziegler et al. 2019) has emerged as a powerful tool for refining generation quality. Our ORACLE first applies supervised fine-tuning (SFT) on verified reasoning paths to ground the model in faithful intermediate steps, and then leverages symbolic validation scores to construct preference pairs for Direct Preference Optimization (DPO), enabling alignment not only with correct outcomes but also with faithful reasoning.

Methodology

In this section, we will explain in detail how our ORACLE works.

Template Design

To facilitate structured reasoning and enable effective information extraction, our ORACLE design a fixed-format response template. This template guides the step-by-step reasoning process of large language models (LLMs), ensuring that key reasoning components can be easily identified and extracted using simple regular expressions. Figure 2 illustrates the complete structure of our template.

In our template, the <QUERY> field denotes the subproblem that needs to be solved at the current step. The field <FACTS> contains the relevant contextual information or premises used to solve the query. The field <RULE> specifies the reasoning principle or the logical rule applied during inference. The contents of <FACTS> and <RULE> conform to the expression form of the premises and rules in a syllogism. To improve robustness and interpretability,

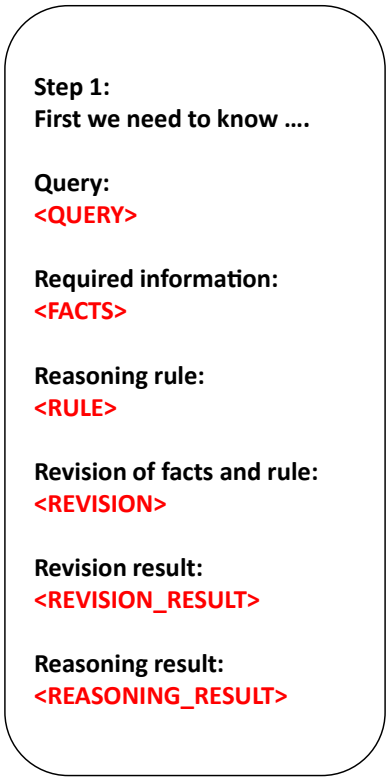

Figure 2: An overview of our structured reasoning template used during data generation of our ORACLE. Each reasoning step consists of modular fields: <QUERY>, <FACTS>, <RULE>, <REVISION>, <REVISION_RESULT>, and <REASONING_RESULT>. This design promotes interpretable reasoning, enables symbolic verification, and facilitates automatic extraction via pattern matching.

we also introduce a field of <REVISION>, which instructs the model to reflect on the appropriateness and sufficiency of the selected facts and the rule. The outcome of this reflective process is recorded in the <REVISION_RESULT> field, indicating whether the original input is retained or revised. Finally, the <REASONING_RESULT> field stores the conclusion derived from applying the reasoning rule to the given facts. During Stage 2, within the beam search process, the <REASONING_RESULT> field is populated with the output of the symbolic reasoning engine, conditioned on the successful execution of the reasoning step. The language model follows this template iteratively, filling in each field at every reasoning step until the final answer is reached.

Model Training

Our ORACLE divide the model training process into two stages, as illustrated in Figure 1. In the first stage, we use a small number of manually crafted reasoning examples formatted with our predefined template to prompt the base LLM in a few-shot manner. This process generates a larger set of synthetic reasoning samples. We then apply a strict filtering mechanism, retaining only those samples that strictly conform to the template structure and yield correct final an-

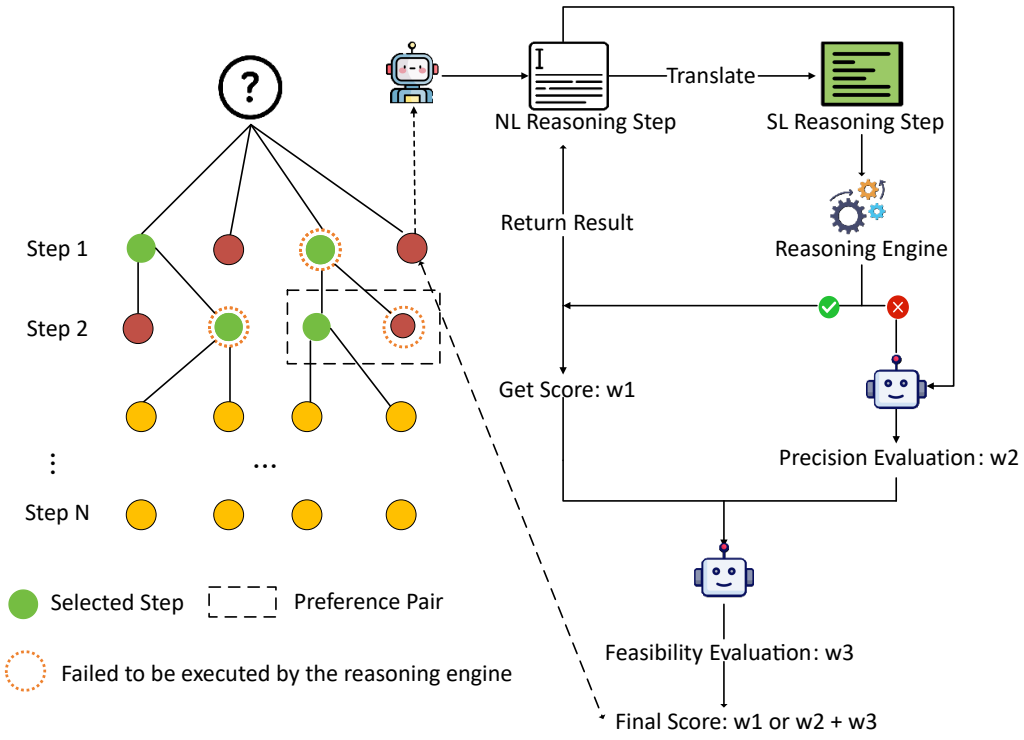

Figure 3: An overview of the beam search integrating a symbolic reasoning engine and LLM-based evaluation of our ORACLE. NL means natural language and SL means symbolic language. For each step, the LLM generates a candidate reasoning step in natural language, which is then translated into symbolic form and passed to the reasoning engine for execution. Each candidate is scored based on execution success (W_1), LLM's precision assessment (W_2), and feasibility estimation (W_3). Candidates that are successfully executed receive a final score of $W_1 + W_3$, while others are scored as $W_2 + W_3$. The top-K candidates based on these scores are selected and expanded in the next layer. Complete reasoning paths that produce correct final answers are collected as supervised fine-tuning(SFT) data. Additionally, preference pairs are constructed by comparing symbolically validated nodes with their invalid siblings for Direct Preference Optimization(DPO).

swers. The filtered data is used to fine-tune the base LLM. The objective of this stage is to encourage the model to internalize the step-by-step reasoning format imposed by the template and to produce structured outputs accordingly.

In the second stage, we integrate the fine-tuned LLM with a reasoning engine. We conduct multiple rounds of interaction between the model and the reasoning engine to generate two types of data: supervised fine-tuning data and preference data, based on the beam search strategy. These are used in succession to further train the base LLM via supervised fine-tuning(SFT) (Ouyang et al. 2022) and Direct Preference Optimization (DPO) (Rafailov et al. 2023). The goal of this stage is to use the reasoning engine as an external verifier to guide the generation of high-quality reasoning data.

Beam Search with LLM Evaluation and Reasoning Engine Validation

To guide the generation of high-quality reasoning paths and construct effective supervision and preference signals, our ORACLE incorporate a beam search strategy that integrates both the large language model and a symbolic reasoning engine, combining the generation ability of a large language model and the rigor of the reasoning engine.

Specifically, the generation of each node in beam search involves the following process. Given an input question q and a sequence of previously generated reasoning steps $x_1, x_2, \ldots$, the LLM fine-tuned in stage 1 will produce the next natural language reasoning step x_i. This step is then translated into a symbolic form by the base LLM with a designed prompt and passed to the reasoning engine. If the symbolic expression is executed successfully, the step receives a score W_1, and the result of the execution is integrated back into the natural language step. otherwise, the LLM performs a precision evaluation of x_i to produce a score W_2. Additionally, regardless of execution success, the LLM evaluates the feasibility of x_i, assigning a score W_3. The final score for the step is computed as $W_1 + W_3$ if it is successfully executed, or $W_2 + W_3$ otherwise.

Assume the beam width at each layer is W, and we select the top K candidates with the highest scores to expand at the next step. Each selected node is allowed to generate W/K children, ensuring that the overall beam width remains con-

Models	Methods	**ProntoQA**	**ProofWriter**	**BoolQ**	**CosmosQA**	**ScienceQA**	**StrategyQA**
Llama-3.1-8B-Instruct	CoT(zero-shot) (Wei et al. 2022)	94.3	53.7	85.0	74.2	90.5	89.9
	CoT(four-shot) (Wei et al. 2022)	92.4	60.3	83.2	79.5	91.9	88.6
	RFT (Yuan et al. 2023)	95.0	62.0	86.3	80.4	**92.4**	89.9
	ToT-SFT (Yao et al. 2023)	93.6	57.0	82.5	74.7	90.7	85.2
	self-rewarding (Yuan et al. 2024)	95.8	63.4	86.2	81.5	92.2	90.8
	ORACLE(Ours)	**97.2**	**67.5**	**87.7**	**82.4**	91.7	**91.8**
Mistral-7B-Instruct-v0.3	CoT(zero-shot) (Wei et al. 2022)	49.3	36.3	83.1	70.6	75.3	87.3
	CoT(four-shot) (Wei et al. 2022)	66.3	50.5	80.5	71.1	77.0	88.1
	RFT (Yuan et al. 2023)	81.3	53.6	84.4	74.0	81.0	89.0
	ToT-SFT (Yao et al. 2023)	82.8	54.8	80.9	72.4	77.4	84.4
	self-rewarding (Yuan et al. 2024)	81.6	58.8	85.6	71.7	80.2	89.0
	ORACLE(Ours)	**87.6**	**64.5**	**86.8**	**78.5**	**83.3**	**89.9**
Qwen-2.5-7B-instruct	CoT(zero-shot) (Wei et al. 2022)	97.0	60.0	82.8	75.9	82.6	86.9
	CoT(four-shot) (Wei et al. 2022)	98.0	63.8	83.0	77.1	84.9	89.9
	RFT (Yuan et al. 2023)	98.5	63.5	84.7	76.0	86.3	87.1
	ToT-SFT (Yao et al. 2023)	96.7	54.8	82.1	73.9	82.9	88.2
	self-rewarding (Yuan et al. 2024)	**98.8**	64.5	83.8	78.1	82.5	91.5
	ORACLE(Ours)	97.7	**68.2**	**87.5**	**82.1**	**87.3**	**92.6**

Table 1: Comparison of various reasoning and data generation methods across six datasets and three models. Our proposed method consistently achieves the best or near-best performance on all datasets and models, demonstrating its effectiveness across symbolic, factual, and commonsense reasoning tasks.

stant. Among the resulting reasoning paths, we retain only the complete paths that reach a final answer and produce a correct answer. These paths are then used as supervised fine-tuning data.

To construct preference pairs for DPO training, we backtrack each reasoning path that successfully leads to the correct final answer and identify intermediate steps (nodes) that can be successfully executed and verified by the symbolic reasoning engine. For each such validated node x_i, if there exists a sibling node x_j that fails engine verification, we construct a preference pair (x_i, x_j), indicating that x_i is preferred over x_j.

This process is illustrated in Figure 3, which shows how symbolic reasoning engine validation and language model evaluation are combined within the beam search.

Experiments

Experiment Setup

Datasets. We evaluate our ORACLE on six datasets covering diverse types of reasoning: ProntoQA (Boratko et al. 2020), ProofWriter (Tafjord, Dalvi, and Clark 2021), BoolQ (Clark et al. 2019), CosmosQA(Huang et al. 2019), ScienceQA(Lu et al. 2022), and StrategyQA(Geva et al. 2021). These datasets span multiple reasoning paradigms, including logical reasoning, factual reasoning, commonsense reasoning, and scientific multi-hop reasoning, allowing for comprehensive and effective evaluation of the reasoning capabilities of LLMs. Although ProntoQA and ProofWriter have standardized contexts, they are representative datasets for verifying logical reasoning ability. We hope to verify that our ORACLE can also achieve better results on these datasets.

Baselines. We compare our method with the following baselines: (1) **CoT** (Wei et al. 2022), where the model is prompted to generate step-by-step reasoning using chain-of-thought prompting without fine-tuned. We evaluate both zero-shot and four-shot settings with greedy decoding. (2) **RFT** (Yuan et al. 2023), where reasoning paths are generated via CoT prompting and filtered using rejection sampling to retain only those that lead to correct answers. (3) **ToT-SFT** (Yao et al. 2023), where multiple reasoning paths are explored using Tree-of-Thoughts, and those with correct final answers are selected as training data for supervised fine-tuning. In our work, we use the prompt in github (Hulbert 2023) to generate. (4) **Self-rewarding** (Yuan et al. 2024), where multiple reasoning paths are generated for each question via CoT prompting, and paths with correct answers are used for supervised fine-tuning. Additionally, the LLM is prompted to score these paths, and the highest- and lowest-scoring answers for the same question are used to construct preference pairs for DPO training.

Implementation details. Our experiments are conducted using the instruction-tuned variants of widely adopted open-source LLMs, including LLaMA-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and Qwen-2.5-7B-Instruct. For efficient training, we apply Low-Rank Adaptation (LoRA) (Hu et al. 2022) with a rank of 8 to all models. When generating data, for all baselines other than CoT, we use four-shot prompting and sample six completions per question. For our ORACLE, the data generation in the first stage uses two-shot prompting. In the second stage, we employ a beam search strategy with a beam width $w = 9$, selecting $k = 3$ top-scoring nodes at each step, each of which generates $w/k = 3$ children. During evaluation, all methods are applied in a zero-shot setting except for CoT. We use Pyke (Frederiksen 2008) as the symbolic reasoning engine. For scoring during beam search, if the symbolic reasoning step is successfully executed by the reasoning engine, it receives a score of $w_1 = 3$; otherwise, it is evaluated by the LLM for correctness, receiving

Models	Methods	ProntoQA	ProofWriter	BoolQ	CosmosQA	ScienceQA	StrategyQA
Llama-3.1-8B-Instruct	w/o engine	96.1	67.0	**87.9**	81.9	90.5	89.9
	w/o beam search	96.5	66.7	87.6	82.2	91.2	90.5
	w/o DPO	96.6	67.2	87.6	82.0	91.4	91.1
	ORACLE(Ours)	**97.2**	**67.5**	87.7	**82.4**	**91.7**	**91.8**
Mistral-7B-Instruct-v0.3	w/o engine	**88.0**	63.8	86.1	78.3	82.9	88.8
	w/o beam search	86.9	64.2	86.6	**79.1**	83.2	89.4
	w/o DPO	87.2	64.4	86.4	78.8	83.1	89.6
	ORACLE(Ours)	87.6	**64.5**	**86.8**	78.5	**83.3**	**89.9**
Qwen-2.5-7B-instruct	w/o engine	97.2	68.0	86.7	**82.4**	86.8	91.5
	w/o beam search	97.5	68.3	87.2	82.1	86.9	92.5
	w/o DPO	97.4	**68.6**	87.4	82.2	87.1	92.3
	ORACLE(Ours)	**97.7**	68.2	**87.5**	82.1	**87.3**	**92.6**

Table 2: Ablation study on key components of our framework. We compare our full method with three ablated variants to analyze the contribution of key components: w/o engine removes reasoning engine's guidance during data generation; w/o beam search disables beam-based search while retaining the reasoning engine; w/o DPO omits Direct Preference Optimization and saves other components during training. While our full method does not always yield the best score on every dataset-model pair, it consistently achieves strong performance and often outperforms ablations across most settings, validating the effectiveness of combining symbolic guidance, beam search, and preference learning.

$w_2 = 2$ if passed, or $w_2 = 0$ otherwise. In all cases, the LLM also conducts a feasibility evaluation, assigning $w_3 = 5$ if passed, and $w_3 = 0$ otherwise. The final score is the sum of w_1 or w_2 and w_3. During data generation, we use a temperature of 1.0, while for inference, the temperature is set to 0.01. The learning rates for SFT and DPO training are 5×10^{-6} and 1×10^{-4}, respectively. For all methods that use SFT or DPO, SFT used 12k samples; DPO used 2k preference pairs after generation and filtering. We use a batch size of 16 (with gradient accumulation) and optimize the model using AdamW. All experiments are conducted on NVIDIA A100 GPUs. Models are trained for 3 epochs.

Experimental Results and Analyses

Performance on ScienceQA. ScienceQA requires complex scientific and multi-hop reasoning. Our ORACLE achieves strong and stable performance across three LLMs, reaching 91.7%, 83.3%, and 87.3% accuracy on LLaMA-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and Qwen-2.5-7B-Instruct respectively. These results consistently surpass most baselines and demonstrate the effectiveness of our approach in enhancing scientific reasoning ability.

Performance on ProntoQA and ProofWriter. On the symbolic reasoning datasets ProntoQA and ProofWriter, our ORACLE significantly outperforms all compared methods. For ProntoQA, it improves accuracy by 1.4% to 6.0% across models, achieving up to 97.7%. On ProofWriter, the gains are even more pronounced, with increases of 3.7% to 5.7%, reaching a peak accuracy of 68.2%. These substantial improvements highlight the superiority of our method in handling complex logical inference tasks.

Performance on StrategyQA, BoolQ and CosmosQA. For factual verification and commonsense reasoning, our ORACLE leads with notable margins. It achieves up to 91.8%, 87.7%, and 82.4% on LLaMA-3.1-8B-Instruct, 89.9%, 86.8%, and 78.5% on Mistral-7B-Instruct-v0.3, and 92.6%, 87.5%, and 82.1% on Qwen-2.5-7B-Instruct for StrategyQA, BoolQ, and CosmosQA respectively. These improvements over the baselines, ranging from 0.9% to 6.1%, highlight the effectiveness of our approach in enhancing performance across a wide spectrum of reasoning tasks. The consistent gains demonstrate robust generalization capabilities, indicating that ORACLE can adapt well to diverse reasoning challenges with varying levels of complexity.

Overall, our ORACLE achieves consistent and significant performance gains across six diverse datasets and three models. This validates the effectiveness and generalization ability of our ORACLE in enhancing both logical reasoning, factual reasoning, commonsense reasoning, and scientific multi-hop reasoning capabilities.

Ablation Study

We perform an ablation study to assess the contributions of three key components in our ORACLE: the reasoning engine, beam search, and Direct Preference Optimization (DPO). Results across six datasets and the three models are shown in Table 2.

Reasoning Engine. Removing the symbolic reasoning engine leads to the most significant performance drops, especially on tasks involving multi-step reasoning such as ProofWriter and StrategyQA. For instance, on LLaMA-3.1-8B-Instruct, accuracy on StrategyQA declines from 91.8% to 89.9%, confirming the engine's role in enforcing logical consistency.

Beam Search. Disabling beam search results in moderate but consistent decreases across most tasks and models. This indicates that search-based generation enhances the model's ability to explore diverse and accurate reasoning paths, particularly when combined with symbolic validation.

Direct Preference Optimization. Removing the DPO stage results in slight but systematic declines in performance, with more pronounced effects observed on tasks that demand subtle semantic interpretation and contextual align-

Models	Iterations	**ProntoQA**	**ProofWriter**	**BoolQ**	**CosmosQA**	**ScienceQA**	**StrategyQA**
Llama-3.1-8B-Instruct	1st iteration	82.7	52.3	45.2	55.2	30.3	24.4
	2nd iteration	84.1	55.7	46.3	56.1	32.1	25.7
Mistral-7B-Instruct-v0.3	1st iteration	64.1	45.2	40.6	43.7	31.8	20.3
	2nd iteration	66.3	46.1	41.2	44.0	32.5	21.0
Qwen-2.5-7B-instruct	1st iteration	57.5	48.7	33.1	47.8	25.5	32.8
	2nd iteration	58.4	49.2	33.0	48.5	25.3	33.4

Table 3: The success rate of the reasoning steps executed by the reasoning engine in our ORACLE on six datasets. With more training iterations, the reasoning engine is able to successfully execute a greater proportion of reasoning steps.

Models	Error Reasons	**ProntoQA**	**ProofWriter**	**BoolQ**	**CosmosQA**	**ScienceQA**	**StrategyQA**
Llama-3.1-8B-Instruct	Generation Error	24.0	40.0	44.0	38.0	72.0	80.0
	Translation Error	76.0	60.0	56.0	62.0	28.0	20.0

Table 4: Error type proportions of our ORACLE on Llama-3.1-8B-Instruct across six datasets when the reasoning engine fails to execute. Generation Error means that the proposed facts or rules are overly complex or incorrectly formatted to be effectively translated into symbolic language. Translation Error means that the limited translation capability of the LLM leads to syntactic or semantic errors in the symbolic representation.

ment, such as BoolQ and ScienceQA. These tasks often require models to distinguish between superficially plausible and truly correct reasoning chains, a capability that is significantly enhanced by preference-based fine-tuning. By explicitly aligning model outputs with human-preferred reasoning patterns, DPO contributes to improving not only the final answer correctness but also the faithfulness and interpretability of intermediate steps.

Overall, our full system achieves the best or near-best results in almost all settings, highlighting the complementary benefits of symbolic guidance, search diversity, and preference alignment. We therefore retain all components in the final framework.

Verification of Reasoning Engine

We further investigate the verification capability of the reasoning engine within our ORACLE. Specifically, we analyze the interaction between the reasoning engine and 1,000 reasoning steps generated by each of models across six datasets. For each case, we record the success rate with which the reasoning engine executes the generated steps.

Table 3 presents the success rates of reasoning steps that are successfully executed by the symbolic reasoning engine across six datasets. Notably, tasks with a stronger emphasis on symbolic reasoning, such as ProofWriter and ProntoQA, exhibit significantly higher success rates, especially for Llama-3.1-8B-Instruct, reaching 52.3% and 82.7% respectively. This indicates that our template-driven framework effectively guides LLMs to generate reasoning steps that align well with formal symbolic logic, facilitating correct execution by the engine.

In contrast, datasets involving more diverse or commonsense reasoning, such as BoolQ, ScienceQA, and StrategyQA, show relatively lower success rates across all models. These tasks place higher demands on the model's inherent reasoning and translation abilities. The LLMs are required not only to generate appropriate premises and rules that conform to the norms of deductive reasoning but also to accurately translate natural language into the corresponding symbolic language.

However, as the numbers of iterations increase, the proportion of reasoning steps successfully executed by the reasoning engine also increases, indicating that through training, LLM not only enhances its reasoning ability, but also enhances its ability to generate standardized reasoning steps.

To analyze the reasons for the failure of the reasoning engine, we analyzed 50 failure examples of our ORACLE on Llama-3.1-8B-Instruct across each of six datasets, and finally summarized them into two main reasons: i) the proposed facts or rules are overly complex or ill-formed to be effectively translated into symbolic language; and ii) the model's limited translation capabilities resulted in syntactic or semantic errors in the symbolic representation. These are also future improvements we will pursue in ORACLE. Specific error type proportions can be found in Table 4.

Although the reasoning engine does not achieve a high execution success rate on certain tasks, our ORACLE enhances robustness by additionally leveraging the LLM to assess the correctness and feasibility of reasoning. The effectiveness of our ORACLE is further validated by the experimental results presented in Table 1.

Conclusion

In this work, we present ORACLE, a structured synthetic reasoning data generation framework that integrates fixed-format prompting, symbolic reasoning supervision, and beam search-based data selection. ORACLE focuses on generating high-quality, verifiable multi-step reasoning data for LLMs tuning to improve reasoning performance. Experimental results across multiple models and reasoning datasets confirm that training on ORACLE-generated data leads to significant improvements in reasoning accuracy. In future work, we plan to explore broader reasoning paradigms and verification strategies to further enhance the quality and applicability of synthetic supervision.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62276283, in part by the China Meteorological Administration's Science and Technology Project under Grant CMA-JBGS202517, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515012985, in part by Guangdong-Hong Kong-Macao Greater Bay Area Meteorological Technology Collaborative Research Project under Grant GHMA2024Z04, in part by Fundamental Research Funds for the Central Universities, Sun Yat-sen University under Grant 23hytd006, and in part by Guangdong Provincial High-Level Young Talent Program under Grant RL2024-151-2-11.

References

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Biderman, S.; Schoelkopf, H.; Anthony, Q. G.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E.; et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2397–2430. PMLR.

Boratko, M.; Li, X.; O'Gorman, T.; Das, R.; Le, D.; and McCallum, A. 2020. ProtoQA: A Question Answering Dataset for Prototypical Common-Sense Reasoning. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1122–1136. Online: Association for Computational Linguistics.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; Ye, W.; Zhang, Y.; Chang, Y.; Yu, P. S.; Yang, Q.; and Xie, X. 2024a. A Survey on Evaluation of Large Language Models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024b. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3): 1–45.

Chen, A.; Chen, A.; Chen, J.; Chi, E. H.; Narang, S.; Song, D.; Wei, J.; and Zhou, D. 2023. Teaching language models to reason with step-by-step feedback. *arXiv preprint arXiv:2304.02399*.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In Burstein, J.; Doran, C.; and Solorio, T., eds., *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)*, 2924–2936. Minneapolis, Minnesota: Association for Computational Linguistics.

Constant, A. 2024. A Bayesian model of legal syllogistic reasoning. *Artificial Intelligence and law*, 32(2): 441–462.

Creswell, A.; Shanahan, M.; and Higgins, I. 2023. Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning. In *The Eleventh International Conference on Learning Representations*.

Frederiksen, B. 2008. Applying expert system technology to code reuse with pyke. *PyCon: Chicago*.

Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D.; and Berant, J. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9: 346–361.

Hadi, M. U.; Qureshi, R.; Shah, A.; Irfan, M.; Zafar, A.; Shaikh, M. B.; Akhtar, N.; Wu, J.; Mirjalili, S.; et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*.

Huang, L.; Le Bras, R.; Bhagavatula, C.; and Choi, Y. 2019. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2391–2401. Hong Kong, China: Association for Computational Linguistics.

Hulbert, D. 2023. Using Tree-of-Thought Prompting to boost ChatGPT's reasoning. https://github.com/dave1010/tree-of-thought-prompting. Accessed: 2025-05-20.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

Jiang, D.; Fonseca, M.; and Cohen, S. B. 2024. Leanreasoner: Boosting complex logical reasoning with lean. *arXiv preprint arXiv:2403.13312*.

Kamoi, R.; Zhang, Y.; Zhang, N.; Han, J.; and Zhang, R. 2024. When can llms actually correct their own mistakes? a critical survey of self-correction of llms. *Transactions of the Association for Computational Linguistics*, 12: 1417–1440.

Leang, J. O. J.; Hong, G.; Li, W.; and Cohen, S. B. 2025. Theorem prover as a judge for synthetic data generation. *arXiv preprint arXiv:2502.13137*.

Lewkowycz, A.; Austin, J.; Zelikman, E.; Li, X.; Wei, J.; Dai, A.; Bosma, M.; Chowdhery, A.; Taylor, S.; Chi, E.; et al. 2022. Solving Quantitative Reasoning Problems with Language Models. *arXiv preprint arXiv:2206.14858*.

Liang, C.; Berant, J.; Le, Q.; Forbus, K. D.; and Lao, N. 2017. Neural Symbolic Machines: Learning Semantic

Parsers on Freebase with Weak Supervision. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 23–33. Vancouver, Canada: Association for Computational Linguistics.

Liang, P.; Bommasani, R.; Zhang, Y.; et al. 2021. The lesson of simplicity: Evidence from scientific flashcards. *arXiv preprint arXiv:2110.06261*.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to explain: multimodal reasoning via thought chains for science question answering. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713871088.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Gupta, S.; Majumder, B. P.; Hermann, K.; Welleck, S.; Yazdanbakhsh, A.; and Clark, P. 2023. SELF-REFINE: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Red Hook, NY, USA: Curran Associates Inc.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Rabe, M. N.; Lee, D.; Bansal, K.; and Szegedy, C. 2021. Mathematical Reasoning via Self-supervised Skip-tree Training. In *International Conference on Learning Representations*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741.

Shum, K.; Diao, S.; and Zhang, T. 2023. Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 12113–12139. Singapore: Association for Computational Linguistics.

Tafjord, O.; Dalvi, B.; and Clark, P. 2021. ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3621–3634. Online: Association for Computational Linguistics.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Turpin, M.; Michael, J.; Perez, E.; and Bowman, S. R. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wan, W.; Yang, Z.; Chen, Y.; Luo, C.; Wang, R.; Cai, K.; Kang, N.; Lin, L.; and Wang, K. 2025. SR-FoT: A Syllogistic-Reasoning Framework of Thought for Large Language Models Tackling Knowledge-based Reasoning Tasks. arXiv:2501.11599.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; and Zhou, D. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 24824–24837.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Yuan, W.; Pang, R. Y.; Cho, K.; Li, X.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2024. Self-rewarding language models. ICML'24. JMLR.org.

Yuan, Z.; Yuan, H.; Li, C.; Dong, G.; Lu, K.; Tan, C.; Zhou, C.; and Zhou, J. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Zelikman, E.; Wu, Y.; Goodman, N. D.; and Manning, C. D. 2022. STaR: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35: 24802–24816.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T.; Radford, A.; Amodei, D.; and Christiano, P. F. 2019. Fine-Tuning Language Models from Human Preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*.