SemCoT: Accelerating Chain-of-Thought Reasoning through Semantically-Aligned Implicit Tokens

Yinhan He[*]
University of Virginia
Charlottesville, VA
nee7ne@virginia.edu

Wendy Zheng[*]
University of Virginia
Charlottesville, VA
ncd9cf@virginia.edu

Yaochen Zhu[*]
University of Virginia
Charlottesville, VA
uqp4qh@virginia.edu

Zaiyi Zheng
University of Virginia
Charlottesville, VA
sjc4fq@virginia.edu

Lin Su
LinkedIn Inc.
Sunnyvale, CA
lsu@linkedin.com

Sriram Vasudevan
LinkedIn Inc.
Sunnyvale, CA
svasudevan@linkedin.com

Qi Guo
LinkedIn Inc.
Sunnyvale, CA
qguo@linkedin.com

Liangjie Hong
LinkedIn Inc.
Sunnyvale, CA
liahong@linkedin.com

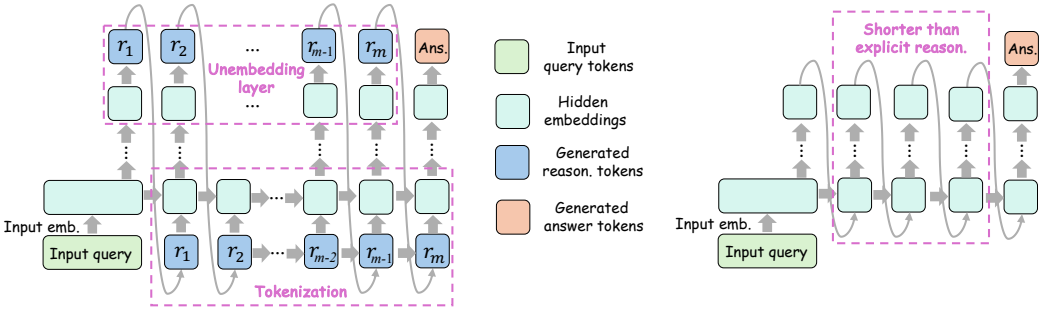
Jundong Li
University of Virginia.
Charlottesville, VA
jl6qk@virginia.edu

Abstract

Chain-of-Thought (CoT) enhances the performance of Large Language Models (LLMs) on reasoning tasks by encouraging step-by-step solutions. However, the verbosity of CoT reasoning hinders its mass deployment in efficiency-critical applications. Recently, implicit CoT approaches have emerged, which encode reasoning steps within LLM's hidden embeddings (termed "implicit reasoning") rather than explicit tokens. This approach accelerates CoT reasoning by reducing the reasoning length and bypassing some LLM components. However, existing implicit CoT methods face two significant challenges: (1) they fail to preserve the semantic alignment between the implicit reasoning (when transformed to natural language) and the ground-truth reasoning, resulting in a significant CoT performance degradation, and (2) they focus on reducing the length of the implicit reasoning; however, they neglect the considerable time cost for an LLM to generate one individual implicit reasoning token. To tackle these challenges, we propose a novel semantically-aligned implicit CoT framework termed `SemCoT`. In particular, for the first challenge, we design a contrastively trained sentence transformer that evaluates semantic alignment between implicit and explicit reasoning, which is used to enforce semantic preservation during implicit reasoning optimization. To address the second challenge, we introduce an efficient implicit reasoning generator by finetuning a lightweight language model using knowledge distillation. This generator is guided by our sentence transformer to distill ground-truth reasoning into semantically aligned implicit reasoning, while also optimizing for accuracy. `SemCoT` is the first approach that enhances CoT efficiency by jointly optimizing token-level generation speed and preserving semantic alignment with ground-truth reasoning. Extensive experiments demonstrate the superior performance of `SemCoT` compared to state-of-the-art methods in both efficiency and effectiveness. Our code can be found at `https://github.com/YinhanHe123/SemCoT`.

[*]These authors contributed equally to this work.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Original CoT (b) Implicit CoT

Figure 1: Illustration of how implicit CoT approaches improve CoT efficiency. "Ans." is the answer. Curl arrows represent that the tokens are autoregressively generated. r_is are explicit reasoning tokens.

1 Introduction

Chain-of-Thought (CoT) [50] is a technique where Large Language Models (LLMs) demonstrate step-by-step reasoning by breaking down complex questions into sequential steps. This approach can be implemented through finetuning with specialized datasets [26, 55] or providing crafted prompt instructions [50]. CoT has achieved remarkable performance in various reasoning-intensive NLP tasks such as mathematical problem solving [21, 32, 29] and symbolic reasoning [53, 29, 43]. In light of the impressive performance gains achieved through Chain-of-Thought (CoT) prompting, several advanced reasoning models have recently been developed, including OpenAI's o1/o3 models [19, 34] and DeepSeek R1 [14]. These models have been specifically trained to tackle problems that require complex, long-chain reasoning. However, as CoT reasoning becomes particularly verbose for LLMs with extensive parameters, it also significantly extends LLMs' reasoning time. For example, ChatGPT-4o [18] may generate as many as around five hundred tokens during CoT reasoning (taking up to 21.37 seconds [3]) to complete a grade school-level math problem, whose answer is simply a numerical number with two to four tokens. Several works aim to improve CoT efficiency, with one primary strategy being *implicit CoT* [8, 15, 41, 3, 44, 54, 42, 9], as shown in Fig. 1 [1]. Implicit CoT substitutes the explicit reasoning tokens with a small number of LLM's first-layer hidden embeddings (each embedding is called an implicit CoT token). Implicit CoT can improve efficiency because it not only pursues more concise reasoning (box in the middle of Fig. 1 (b)) but also avoids going through unembedding layers (box at the upper side in Fig. 1 (a)) to decode from last-layer hidden embeddings to tokens, and the tokenization (box at the lower side in Fig. 1 (a)) of reasoning tokens.

However, current implicit CoT methods encounter two significant challenges: (1) **Gap towards Maintaining Effectiveness**: Current implicit CoT methods struggle to establish semantic alignment between implicit embeddings (when transformed to natural language) and ground-truth reasoning. This "semantic alignment difficulty" stems from a fundamental format mismatch: implicit CoT exists as hidden embeddings while ground-truth reasoning consists of natural language. Existing approaches inadequately address this by either: (*i*) completely discarding ground-truth reasoning [13, 54], (*ii*) matching only keywords from ground-truth reasoning [3], or (*iii*) first finetune LLMs with ground-truth reasoning, then optimizing implicit reasoning only for generating correct answers [15, 8]. (2) **Gap towards Enhancing Efficiency**: Although some implicit CoT methods [3, 42, 15, 9] reduce reasoning length, they neglect the high computational cost for generating each reasoning token with the LLM. This challenge becomes particularly important when LLMs scale to substantial sizes (hundreds of billions of parameters) nowadays. For example, generating one token can cost as high as approximately 0.1s for DeepSeek-R1 [7, 36]. The computational overhead accumulates significantly during reasoning, especially for complex problems requiring extensive step-by-step thinking.

To tackle the above challenges, in this paper, we propose a novel framework named `SemCoT` (Semantically-aligned Implicit CoT) to accelerate implicit CoT while effectively preserving the performance benefits of traditional CoT methods. Specifically, to tackle the first challenge, we propose a customized sentence transformer to measure the semantic alignment between implicit reasoning and ground-truth reasoning. The customized sentence transformer converts each of the im-

[1] The figure shows how most (**not all**) implicit CoT methods perform inference with a query.

plicit reasoning and ground-truth reasoning into an embedding vector that characterizes its semantics. Thus, we can compare their semantic alignment by cosine similarity [2]. The well-trained, customized sentence transformer is utilized to optimize implicit CoT generation to be semantically aligned with ground-truth reasoning. To tackle the second challenge, we adopt a lightweight language model (LM) [2], which is an off-the-shelf LM that is distilled or sheared (i.e., pruned) from the original LLM, to generate implicit reasoning. This approach dramatically reduces the time cost of generating each single CoT token. The lightweight LM is guided by our sentence transformer to distill ground-truth reasoning into semantically aligned implicit reasoning, while also optimized for answer accuracy [12].

The main contribution of this paper is summarized as follows. (1) **Problem Identification.** We uncover two fundamental gaps in existing implicit CoT: the failure to preserve semantic alignment between implicit and explicit reasoning and the computational cost of generating individual reasoning tokens. (2) **Method Design.** We propose a novel framework named `SemCoT`, which jointly optimizes the token-level generation speed of implicit reasoning while maintaining semantic alignment with ground-truth reasoning. (3) **Experimental Evaluation.** We conduct comprehensive experiments to test `SemCoT` and state-of-the-art implicit CoT reasoning baselines on multiple LLMs and real-world NLP tasks to verify the effectiveness and efficiency of the proposed framework `SemCoT`.

2 Preliminaries and Problem Definition

Preliminaries. We introduce the terminology throughout the work. Let Q denote a query requiring complex reasoning, and $Y = [y_1, ..., y_N]$ represent the ground-truth answer of N tokens. The original CoT reasoning process generates M ground-truth explicit reasoning tokens $R = [r_1, ..., r_M]$ before producing Y. In contrast, our approach leverages implicit reasoning tokens, denoted as $\mathbf{Z}$, which are embedding vectors that encode reasoning information without requiring the generation of explicit tokens. The *white-box* LLM with parameters g is represented as $\mathcal{F}_g$, and within the LLM $\mathcal{T}_g$ denotes the mapping converting sentences into LLM input embeddings. The lightweight implicit reasoning generator is denoted as $\mathcal{I}_\psi$ with parameters ψ. We use $\mathcal{C}_\phi$ to represent our customized sentence transformer with parameters ϕ, which evaluates semantic alignment between the ground-truth reasoning R and the implicit reasoning $\mathbf{Z}$. The probability of the LLM generating token y_i given preceding tokens $y_{1:i-1}$, query Q, and implicit reasoning $\mathbf{Z}$ is denoted as $\mathcal{P}_\mathcal{F}(y_i|y_{1:i-1}, Q, \mathbf{Z})$.

Problem 1. ***Efficient CoT with Implicit Reasoning.*** *Given a query Q, we aim to minimize the total time T for an white-box LLM to generate the implicit reasoning $\mathbf{Z}$ before achieving the answer Y, i.e., $\min_\psi T(Q, \mathbf{Z}, Y)$, subject to maximal retention of answer accuracy compared with the original CoT.*

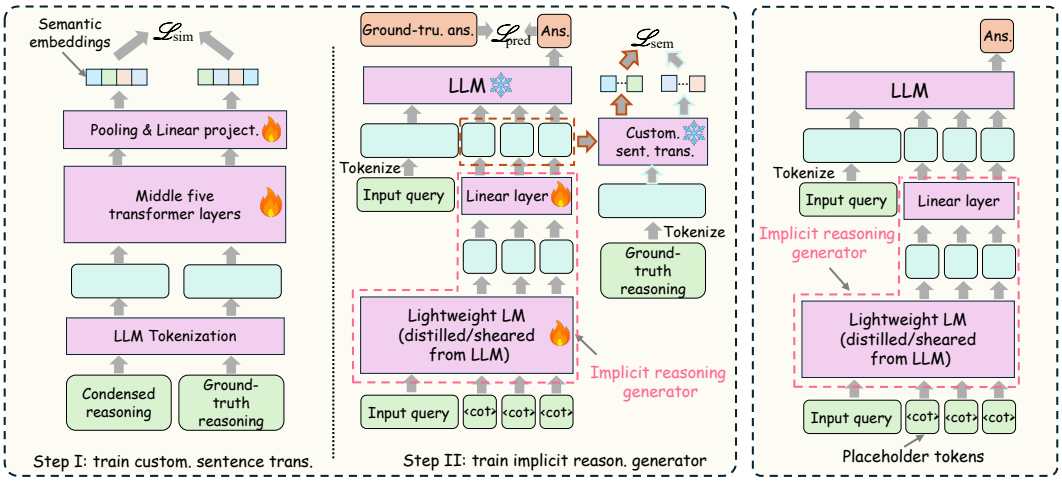
3 Methodology

In this section, we first present an overview of the proposed framework `SemCoT`, followed by the detailed elaboration on its two steps performed sequentially: *implicit v.s. ground-truth reasoning semantic alignment assessment* and *efficient implicit reasoning generation*.

3.1 Overview

We introduce the workflow of our proposed framework `SemCoT`, as shown in Fig. 2. `SemCoT` addresses two key challenges through a two-step process: The first step, *implicit vs. ground-truth reasoning semantic alignment assessment*, trains a *customized sentence transformer* using contrastive learning. This sentence transformer assesses how semantically aligned the implicit reasoning is with the ground-truth reasoning, addressing our first challenge: the gap towards maintaining CoT effectiveness, by ensuring semantic alignment between implicit and ground-truth reasoning. In the second step, *efficient implicit reasoning generation*, we finetune a *lightweight implicit reasoning generator* to produce implicit reasoning efficiently. This addresses our second challenge: enhancing CoT efficiency. The generator is optimized for two objectives simultaneously: (1) semantic alignment with explicit reasoning (guided by our trained sentence transformer) and (2) answer correctness when the LLM generates answers using implicit reasoning. In conclusion, `SemCoT` efficiently generates implicit reasoning that preserves the semantics of ground-truth reasoning, successfully addressing both challenges of maintaining effectiveness while improving efficiency.

[2] As the lightweight model may not be a large language model, we only call it a language model.



(a) Training stage (b) Inference stage

Figure 2: Overview of the proposed SemCoT. Each *cyan box* is a hidden text embedding within model components, with the text content and model type varying based on the box's position in the figure. *Fire* and *snowflake* signs mean the component is trained and frozen, respectively.

3.2 Implicit vs. Ground-truth Reasoning Semantic Alignment Assessment

Motivation. As introduced in Section 1, existing methods struggle to fully capture ground-truth reasoning information in implicit reasoning. We address this challenge with a natural assumption: optimal implicit CoT performance [3] is achieved when the implicit CoT semantically aligns with the ground-truth CoT. This alignment occurs when the implicit CoT, if translated from the LLM's embedding space to human language, would be semantically equivalent to the ground-truth reasoning. However, this "translation" presents a fundamental challenge: we cannot formulate it in explicit formulations due to its complexity, nor can we model it with neural networks because human language space is discrete and therefore not differentiable. As a practical alternative, we measure this alignment within the LLM's embedding space by comparing the input embedding of ground-truth reasoning (what we call "embedded ground-truth reasoning") with the implicit reasoning. Although one might consider directly using vector distance metrics to measure semantic alignment between implicit reasoning and embedded ground-truth reasoning, such approaches have significant limitations: (1) LLM embeddings are optimized for next-token prediction rather than sentence semantics, vector distance may not faithfully reveal semantic similarity between sentences[40]. (2) LLM embeddings are high dimensional and suffer from the curse of dimensionality [48], where distance metrics become less meaningful as dimensionality increases [17].

Customized Sentence Transformer. Recent advances in sentence transformers [40, 1, 10] offer state-of-the-art methods to measure semantic relationships between sentences. They utilize a transformer-based architecture combined with pooling and linear layers to produce semantic vector representations of input sentences, enabling the measurement of sentence similarity via cosine similarity. Although traditional sentence transformers cannot be directly applied in our context—since implicit reasoning steps are not explicitly formulated as sentences—they inspire our approach. A naive solution would involve bypassing the tokenization step of sentence transformers and directly feeding implicit or embedded ground-truth reasoning into these models. However, this approach is inadequate, as standard sentence transformers and the LLM we examine typically employ different mappings from tokens to input embeddings, resulting in embedding spaces that differ semantically. To address this issue, we design a customized sentence transformer specifically tailored to handle embeddings from LLMs, thereby accurately measuring the semantic alignment between reasoning.

We design a customized sentence transformer, shown in Fig. 2 (a), denoted as $\mathcal{C}_\phi$ (with parameters ϕ), tailored to comparing semantics between implicit reasoning and ground-truth reasoning. First, we extract the middle five layers of the LLM that we perform CoT on to serve as the backbone of our customized sentence transformer, as these layers have been shown in prior work [27] to possess optimal language modeling ability and transferability across tasks. Next, we add a pooling layer on

[3] "Implicit CoT performance" refers to the LLM answer accuracy with the implicit CoT reasoning.

top of the transformer layers to aggregate token embeddings from the entire reasoning sequence into a unified vector representation. Finally, we employ a linear layer to project this unified vector into a lower-dimensional semantic embedding space, facilitating more efficient similarity comparisons.

The customized sentence transformer is trained with a specially crafted reasoning pair dataset $\mathcal{G}$, where each data point $(R, S) \in \mathcal{G}$ is a reasoning pair with R being the ground-truth reasoning and S a GPT-4o-mini [33] generated condensed semantically aligned reasoning of R. The prompt to generate S with GPT-4o-mini [33] is "*Please generate the most condensed reasoning text which is semantically aligned with the following reasoning text: R.*" We train the sentence transformer with a contrastive learning [23] strategy. Specifically, we compute the semantic embeddings of the ground-truth reasoning $\mathcal{C}_\phi(\mathcal{T}_\mathcal{F}(R_i))$ and condensed reasoning $\mathcal{C}_\phi(\mathcal{T}_\mathcal{F}(S_i))$, where $\mathcal{T}_\mathcal{F}$ is the mapping from tokens to input embeddings of the LLM. The sentence transformer is trained using contrastive learning to maximize the similarity between embeddings of positive pairs (R_i, S_i), while minimizing similarities between all negative pairs (R_i, S_j) for $j \neq i$ within the same minibatch. Formally, we learn the sentence transformer with

$$\mathcal{L}_{sim} = -\frac{1}{|\mathcal{G}|} \sum_{(R_i,S_i)\in\mathcal{G}} \log \frac{\exp(sim(e_{R_i}, e_{S_i})/\tau)}{\sum_{j=1}^{|\mathcal{G}|} \exp(sim(e_{R_i}, e_{S_j})/\tau)}, \tag{1}$$

where $e_{L_i} := \mathcal{C}_\phi(\mathcal{T}_\mathcal{F}(L_i))$ and $e_{S_i} := \mathcal{C}_\phi(\mathcal{T}_\mathcal{F}(S_i))$, $sim(e_{L_i}, e_{S_i})$ is the cosine similarity between these embeddings, and τ controls the distribution concentration over negative samples.

3.3 Token-level Efficient Implicit Reasoning Generation

Motivation. LLM typically requires significant time to generate even a single implicit CoT token. Thus, we propose using lightweight language models to generate implicit reasoning efficiently. However, a key problem arises: the hidden embeddings of lightweight models do not naturally align with the embedding space of LLMs. This misalignment occurs for two reasons: (1) the embedding dimensions may differ, and (2) the semantic distributions between the embedding spaces of the two models can vary significantly. Nevertheless, if we carefully select a compatible lightweight model whose semantic distribution resembles a linear projection of the LLM's distribution, we can potentially learn a simple linear transformation layer to align their embedding spaces effectively.

Lightweight Implicit Reasoning Generator. We build the lightweight implicit reasoning generator, shown on the right in Fig. 2 (a), based on an off-the-shelf pruned or distilled LM (e.g., Sheared-LLaMA-1.3B [52]) from the LLM (e.g., Llama-2-7b-chat-hf [47]). We choose to use a pruned or distilled LMs of the original LLMs because existing research [46, 49] suggests that those models preserve crucial semantic properties of their original LLMs in their embedding spaces. Then, we incorporate a linear projection layer to map the last-layer hidden embedding generated by the lightweight LM into the embedding space of the LLM to serve as implicit reasoning for the LLM.

More specifically, given a query Q, we first append the query with k `<CoT>` tokens, with k being the length of the implicit reasoning. Here, the `<CoT>` token is a special token added to the lightweight implicit reasoning generator's vocabulary. Then, the implicit reasoning generator, denoted as $\mathcal{I}_\psi$ with parameters ψ, processes the concatenated text and collects the last-layer hidden embeddings of all the `<CoT>` tokens. Finally, we linearly transform the embeddings to obtain the generated implicit reasoning $\mathbf{Z} = \mathcal{I}_\psi(Q)$. We learn to generate the ground-truth answer $Y = [y_1, ..., y_N]$ (N is the length of the answer measured in tokens) with cross-entropy loss:

$$\mathcal{L}_{pred}(\psi) = -\frac{1}{N|\mathcal{D}|} \sum_{j=1}^{|\mathcal{D}|} \sum_{i=1}^{N} \log P_\mathcal{F}(y_i | y_{1:i-1}, Q_j, \mathbf{Z}_j). \tag{2}$$

Here, y_i is the ith token of the ground-truth answer, and $P_\mathcal{F}(y_i|y_{1:i-1}, Q, \mathbf{Z})$ is the LLM's probability of predicting y_i when provided the query, implicit reasoning and all answer tokens before y_i. In addition, we adopt a knowledge distillation training objective to encourage the implicit reasoning to be semantically aligned to the ground-truth reasoning. First, we tokenize the ground-truth reasoning R with the LLM and produce its input embedding $\mathcal{T}_\mathcal{F}(R)$. Then, we measure the semantic alignment between the embedded ground-truth and the implicit reasoning by computing their semantic embedding vectors generated by the well-trained sentence transformer $\mathcal{C}_\phi$. Formally, we train the implicit

reasoning with the following semantic alignment loss $\mathcal{L}_{sem}$

$$\mathcal{L}_{sem}(\psi) = -\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} sim(\mathcal{C}_\phi(\mathcal{T}_\mathcal{F}(R_i)), \mathcal{C}_\phi(Z_i)). \tag{3}$$

Therefore, we train the implicit reasoning generator with $\mathcal{L}_{total} = \lambda\mathcal{L}_{sem} + (1-\lambda)\mathcal{L}_{pred}$, where the λ is a hyperparameter controlling the degree to which semantic alignment between the implicit and ground-truth reasoning is emphasized during the training of the implicit reasoning generator.

3.4 Training and Inference Strategy

Training Strategy. Based on our elaborations above, here we present a summary of the training strategy for the proposed `SemCoT`. For the first step (left-hand side of Fig. 2 (a)), we optimize the parameters of the sentence transformer ϕ with contrastive learning loss $\mathcal{L}_{sim}$. For the second step (right-hand side of Fig. 2 (a)), we optimize the parameters of the implicit reasoning generator ψ with the aid of a customized sentence transformer trained in the first step. In this second step, the parameters of the sentence transformer ϕ are frozen. The parameters of the implicit reasoning generator ψ is trained with loss $\mathcal{L}_{total}$, which combines $\mathcal{L}_{pred}$ for enhancing answer accuracy and $\mathcal{L}_{sem}$ to semantically align the implicit reasoning with the ground-truth reasoning.

At each training step, we begin with a warm-up phase that trains only the final linear layer (see Fig. 2 (a)) of the component being updated — either ϕ_{linear} for the customized sentence transformer in the first step, or ψ_{linear} for the implicit reasoning generator in the second step.

Inference Strategy. Fig. 2 (b) illustrates the inference strategy. During inference, given a query, we first concatenate the query with k `CoT` tokens. The concatenated text is then processed by the implicit reasoning generator to generate implicit reasoning. Finally, we concatenate the implicit reasoning to the end of the query's input embedding and input it into the LLM to generate the final answer.

4 Experimental Evaluations

In this section, we first introduce the experiment setup. Then, we discuss the evaluation results of the `SemCoT`. Specifically, we aim to answer the following research questions: **RQ1**: How well can `SemCoT` improve the CoT reasoning efficiency and retain CoT effectiveness compared with the state-of-the-art efficient CoT baselines? **RQ2**: To what extent does each component of `SemCoT` contribute to the overall CoT reasoning performance? **RQ3**: How do hyperparameters such as the number of implicit reasoning tokens affect SemCoT's performance? **RQ4**: Are there evidence to prove that `SemCoT` can learn semantically-aligned implicit reasoning while baselines cannot?

4.1 Experiment Settings

We introduce the experiment settings. More details (e.g., hardware information) are in Appendix C.

Datasets. We adopt five representative datasets from three different semantic domains used for benchmarking CoT performance. Specifically, we apply mathematical reasoning datasets GSM8K [5], SVAMP [38], MultiArith [4], commonsense reasoning dataset CommonsenseQA [45], and symbolic reasoning dataset CoinFlip [22]. Please see Appendix C for the metadata of the five used datasets.

Examined LLM & Implicit Reasoning Generator. We examine `SemCoT` on two representative open source LLMs, Llama-2-7b-chat-hf [47] and Mistral-7B-Instruct-v0.2 [20]. For the lightweight LM within the implicit reasoning generator, we utilize the distilled/sheared LMs from their corresponding examined LLMs. Specifically, in the cases that the LLM is Llama-2-7b-chat-hf [47], we employ Sheared-LLaMA-1.3B [52] as the lightweight LM within the implicit reasoning generator. When the backbone LLM is Mistral-7B-Instruct-v0.2 [20], we apply mistral-1.1b-testing [35] accordingly.

Baselines. We adopt state-of-the-art baselines improving CoT efficiency on LLMs. Specifically, (1) *Pause* [13] uses identical implicit reasoning tokens to substitute ground-truth reasoning. Their training strategy includes pretraining and finetuning the LLM for optimal answer accuracy with implicit reasoning. We only implement the finetuning strategy due to the significant cost of LLM pretraining. (2) Progressive encoding methods, such as *ICoT-SI* [8] and *COCONUT* [54], use implicit

Table 1: Performance of SemCoT vs. baselines across datasets (best results in bold). Green highlight indicates the best values, and blue highlight indicates the runner-up.

			COCONUT	CODI	ICoT-SI	Pause	SoftCoT	**SemCoT** (ours)
Llama	CoinFlip	Acc (%)	77.67 ± 5.10	**97.33 ± 3.77**	65.17 ± 2.39	77.67 ± 5.44	58.33 ± 0.85	87.00 ± 17.33
		Time (s)	1.58 ± 0.01	1.60 ± 0.01	1.47 ± 0.01	1.50 ± 0.01	1.08 ± 0.01	**1.06 ± 0.01**
	Common	Acc (%)	94.67 ± 0.85	93.83 ± 3.66	98.00 ± 0.41	88.67 ± 1.43	97.00 ± 1.08	**98.33 ± 1.03**
		Time (s)	1.65 ± 0.01	1.66 ± 0.05	1.46 ± 0.00	1.20 ± 0.03	**1.01 ± 0.05**	1.06 ± 0.02
	GSM8K	Acc (%)	3.00 ± 0.71	5.00 ± 0.41	4.50 ± 0.41	4.33 ± 2.32	8.83 ± 1.03	**9.83 ± 0.24**
		Time (s)	1.60 ± 0.01	1.59 ± 0.02	1.48 ± 0.01	1.46 ± 0.01	1.05 ± 0.00	**1.02 ± 0.08**
	MultiArith	Acc (%)	9.63 ± 0.69	11.85 ± 2.95	8.70 ± 1.31	11.85 ± 6.19	7.96 ± 0.69	**15.93 ± 1.14**
		Time (s)	1.62 ± 0.04	1.58 ± 0.01	1.48 ± 0.01	1.45 ± 0.00	1.03 ± 0.01	**0.94 ± 0.11**
	SVAMP	Acc (%)	33.00 ± 1.87	15.50 ± 1.87	26.00 ± 1.08	20.83 ± 8.34	38.17 ± 2.90	**46.33 ± 0.85**
		Time (s)	1.67 ± 0.05	1.58 ± 0.01	1.48 ± 0.01	1.48 ± 0.01	1.06 ± 0.00	**1.00 ± 0.13**
Mistral	CoinFlip	Acc (%)	50.00 ± 4.02	66.67 ± 47.14	41.83 ± 3.92	63.50 ± 3.24	49.17 ± 9.40	**82.17 ± 9.53**
		Time (s)	1.75 ± 0.04	1.85 ± 0.02	1.75 ± 0.02	1.70 ± 0.05	1.31 ± 0.02	**1.30 ± 0.07**
	Common	Acc (%)	97.00 ± 0.00	99.17 ± 0.47	99.33 ± 0.62	**99.67 ± 0.47**	98.50 ± 1.41	**99.67 ± 0.24**
		Time (s)	1.38 ± 0.01	1.84 ± 0.01	1.69 ± 0.03	1.67 ± 0.03	**1.18 ± 0.13**	1.27 ± 0.05
	GSM8K	Acc (%)	5.83 ± 1.89	6.00 ± 2.94	10.17 ± 2.05	14.83 ± 1.03	12.17 ± 2.62	**18.50 ± 2.94**
		Time (s)	1.93 ± 0.02	1.85 ± 0.00	1.69 ± 0.03	1.76 ± 0.00	**1.35 ± 0.01**	**1.35 ± 0.01**
	MultiArith	Acc (%)	3.33 ± 0.45	5.00 ± 7.07	1.85 ± 0.26	**40.93 ± 5.02**	18.70 ± 0.94	38.89 ± 12.53
		Time (s)	1.88 ± 0.01	1.83 ± 0.00	1.73 ± 0.02	1.79 ± 0.02	1.32 ± 0.01	**1.31 ± 0.00**
	SVAMP	Acc (%)	45.17 ± 1.93	13.50 ± 0.71	**54.17 ± 2.01**	48.83 ± 1.65	29.17 ± 2.72	53.83 ± 3.06
		Time (s)	1.84 ± 0.02	1.85 ± 0.01	1.73 ± 0.01	1.71 ± 0.01	1.32 ± 0.02	**1.30 ± 0.02**

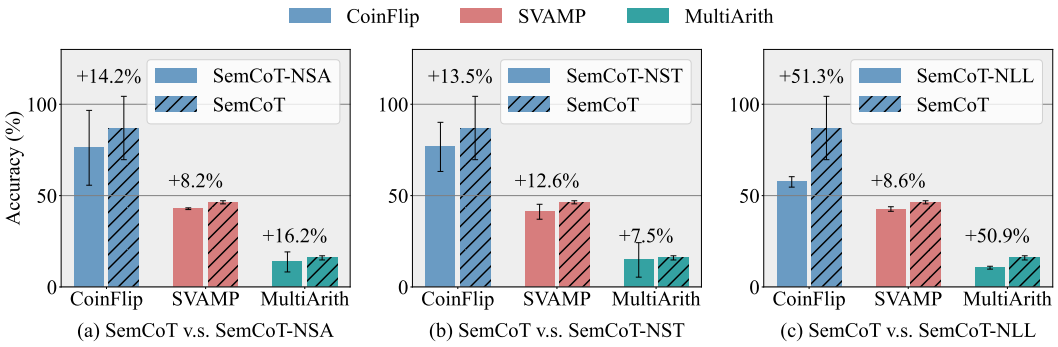
tokens to substitute explicit tokens gradually during LLM finetuning. (3) *CODI* [42] adopts a self-distillation strategy, where the teacher and student are both the LLM. The teacher learns to perform explicit reasoning, while the student learns to generate implicit tokens that lead to the correct answer. Simultaneously, both models are trained to align their token embeddings at the final token position of the query. (4) *SoftCoT* [54] utilizes a small LM to generate LLM's hidden reasoning tokens without compressing the number of hidden tokens compared with the original explicit reasoning chain.

Evaluation Metrics. All queries in the adopted datasets are accompanied by ground-truth answers. Therefore, we use *answer accuracy* as the metric to evaluate CoT effectiveness. This is defined as the percentage of test queries for which the LLM, when equipped with a given efficient CoT framework, successfully generates the correct ground-truth answer in response to the input question. For efficiency, we measure the *average wall-clock time* for the LLM to generate an answer to a query.

Implementation Details. We set the output embedding dimension of the customized sentence transformer to 768. The number of implicit tokens during training is five, and, during evaluation, it is set to one. We optimize both the customized sentence transformer and the implicit reasoning generator with AdamW [28] using the best hyperparameters found. For inference, we allow up to thirty answer tokens to be generated to enforce the LLM to generate the answer instead of the CoT.

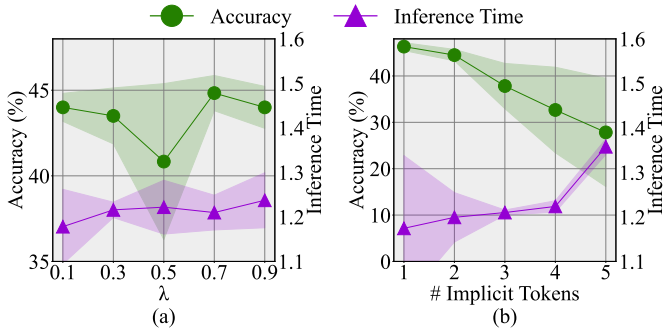
4.2 Effectiveness & Efficiency of SemCoT

In this subsection, we aim to answer **RQ1**. Specifically, we evaluate our proposed framework SemCoT on two LLMs, Llama-2-7b-chat-hf [47] and Mistral-7B-Instruct-v0.2 [20]. We compare the two LLMs' answer accuracy and inference time cost when performing CoT with our SemCoT and the state-of-the-art baselines using one implicit reasoning token. We show the results in Table 1. From Table 1, we can make the following observations: (1) from the perspective of *effectiveness*, our SemCoT achieves the highest answer accuracy compared with the baseline methods in most datasets, showing that our method effectively retains LLM's reasoning ability in a wide spectrum of NLP tasks and LLM settings. (2) From the perspective of *efficiency*, SemCoT achieves nearly the fastest implicit reasoning processing time across all datasets and LLMs. In some datasets, SemCoT is slightly slower than SoftCoT, but SemCoT achieves much higher answer accuracy in those cases. In conclusion, SemCoT achieves the optimal performance w.r.t. the trade-off between efficiency and effectiveness.



(a) SemCoT v.s. SemCoT-NSA (b) SemCoT v.s. SemCoT-NST (c) SemCoT v.s. SemCoT-NLL

Figure 3: Ablation study of SemCoT.



(a) (b)

Figure 4: Parameter sensitivity of SemCoT.

4.3 Ablation Study

We aim to answer **RQ2** by evaluating the effect of different components in SemCoT with its three variants: (1) SemCoT-NST: we remove the customized sentence transformer and modify the semantic alignment loss as the cosine similarity between the mean-pooled embeddings of the implicit reasoning tokens and the ground-truth reasoning; (2) SemCoT-NSA: we remove the entire semantic alignment loss when training the implicit reasoning generator; (3) SemCoT-NLL: we replace the lightweight language model in the implicit reasoning generator with the original LLM and finetune the LLM using LoRA [16] to generate implicit reasoning. We conduct experiments with LLama-2-7B-chat-hf [47] on various datasets and observe from Fig. 3 that (1) Generally, SemCoT's performance decrease when any component is removed, showing that each component of the SemCoT positively contributes to the CoT performance; (2) Comparing Fig. 3 (a) and Fig. 3 (b), we find that removing the semantic alignment loss results in worse performance compared to that of optimizing with semantic alignment measured with cosine similarity, highlighting the significance of the semantic alignment loss in learning implicit reasoning; (3) From Fig. 3 (c), we surprisingly find that finetuning the original LLM to generate the implicit reasoning is less effective than utilizing the lightweight LMs, likely attributed to catastrophic forgetting induced by LLM finetuning [54].

4.4 Parameter Sensitivity

We answer **RQ3** by studying the fluctuation of SemCoT performance w.r.t. the change of hyper-parameter λ. Here, λ controls the weight of the semantic alignment loss when training the implicit reasoning generator. More specifically, we vary λ in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, and we present the corresponding performance differences of SemCoT when applied to Llama-2-7b-chat-hf [47] on SVAMP [38]. In addition, we also investigate the effect of the number of implicit reasoning tokens M when LLM performs inference. We vary its value among $M = \{1, 2, 3, 4, 5\}$ and record the performance of SemCoT using the same setting as when evaluating λ. The results are shown in Fig. 4. We make the following observations: (1) The LLM answer accuracy generally increases except for when λ is 0.5, potentially due to the delicate balance between semantic alignment and prediction accuracy. This is also indicated by the unusual large variance. The inference time is consistent under different λ. (2) The LLM answer accuracy decreases as the number of implicit tokens increase,

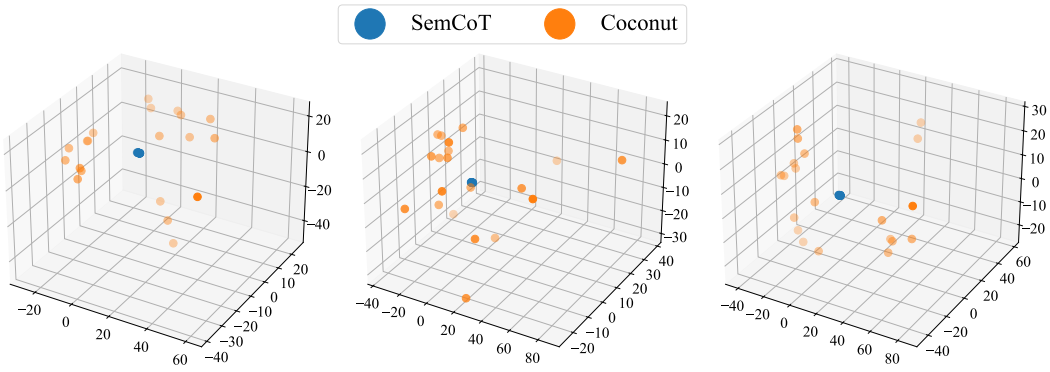


Figure 5: Case study comparing SemCoT vs COCONUT. PCA plots of implicit reasoning embeddings for 3 SVAMP queries with 20 semantic variants each. SemCoT (blue) has tighter clustering than COCONUT (orange), showing its ability to generate semantic aligned reasoning.

indicating that the implicit reasoning can concisely embed ground-truth reasoning information. We observe similar findings in other datasets as well.

4.5 Case Study

We answer **RQ4** through a case study by comparing the implicit reasoning of SemCoT with the baselines. Since directly decoding implicit reasoning to human language is challenging (see Section 3.2), we adopt a proxy evaluation based on this assumption: if an implicit CoT method effectively generates semantically-aligned implicit reasoning from ground-truth reasoning, it should produce consistent implicit reasoning when processing semantically aligned queries. We thus randomly select three queries from the SVAMP [38] dataset and use GPT-4o-mini [33] to generate 20 semantically equivalent variants for each query. We visualize (with PCA [11]) the first implicit reasoning tokens produced by both SemCoT (blue) and the baseline COCONUT (orange) when processing these variants (see case studies for other baseline methods in Appendix D.3). As shown in Fig. 5, the implicit reasoning generated by SemCoT is significantly more concentrated compared to that of COCONUT. The tight clustering suggests that SemCoT successfully distills essential reasoning information regardless of surface-level linguistic differences in queries.

5 Related Work

Efficient CoT with Implicit Reasoning. Stepwise or progressive encoding methods encode ground-truth reasoning implicitly by first familiarizing the LLM with the reasoning and then removing it, expecting the LLM to learn hidden embeddings that capture previously learned information [8, 15, 41]. These techniques are indirect and susceptible to noise. Other methods compress parts of the reasoning sequence, with CCoT [3] focusing on keywords and Token Assorted [44] selecting parts through a specially trained neural network. SoftCoT [54] generates the implicit reasoning with a small language model and only trains a linear layer to transform the implicit reasoning of the small LM to the LLM for answer correctness, completely bypassing ground-truth reasoning. CODI [42] employs self-distillation where the teacher model learns ground-truth reasoning while the student learns to generate correct answers with implicit reasoning, the teacher distills the LLM embedding of the last query token to the student in order to encode ground-truth reasoning semantics to the student. This approach lacks verification that the final token fully encodes reasoning semantics. These methods barely consider token-level reasoning generation speed and reasoning semantic alignment.

Semantic Textual Similarity. Semantic Textual Similarity (STS) has evolved from *word-level lexical* [25, 30] approaches to *word-level semantic* [31, 39] methods, and finally to *sentence-level semantic* [40, 1, 6] techniques. Early methods relied on lexical overlap and WordNet-based measures, often struggling with synonymy and polysemy [25]. Word-level semantic embeddings like Word2Vec [31] and GloVe [39] marked a significant advancement, with embeddings typically aggregated to create comparable sentence representations. Recent specialized neural architectures focus on sentence-level representations, including Sentence-BERT (SBERT) [40], which modified BERT with siamese networks to derive semantically meaningful sentence embeddings that can be

compared using cosine similarity. Similar approaches include Universal Sentence Encoder [1] and InferSent [6], which share the goal of creating fixed-length sentence representations. Recent works improve embedding space uniformity through contrastive learning strategies [10, 56, 24]. However, these STS methods cannot compare semantics between LLMs' embedding and natural language text.

6 Conclusion

In this paper, we introduce `SemCoT`, a novel framework that accelerates Chain-of-Thought reasoning while preserving semantic alignment between implicit and explicit reasoning. Our approach addresses two key challenges in implicit CoT methods: maintaining semantic alignment with ground-truth reasoning and optimizing token-level generation speed. Extensive experiments across multiple datasets and LLMs demonstrate that `SemCoT` achieves superior performance in both efficiency and effectiveness compared to state-of-the-art baselines. `SemCoT` represents a significant advancement in making advanced reasoning capabilities more computationally accessible for real-world applications.

7 Acknowledgments

This work is supported in part by the National Science Foundation (NSF) under grants IIS-2006844, IIS-2144209, IIS-2223769, IIS-2331315, CNS-2154962, BCS-2228534, and CMMI-2411248, the Office of Naval Research (ONR) under grant N000142412636, and the Commonwealth Cyber Initiative (CCI) under grant VV-1Q25-004.

References

[1] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al. Universal sentence encoder for english. In *EMNLP*, 2018.

[2] D. Chandrasekaran and V. Mago. Evolution of semantic similarity—a survey. *ACM Comput. Surv.*, 2021.

[3] J. Cheng and B. Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv*, 2024.

[4] ChilleD. Multiarith dataset. Hugging Face Dataset Repository, 2023.

[5] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv*, 2021.

[6] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*, 2017.

[7] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, and J. Liang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv*, 2025.

[8] Y. Deng, Y. Choi, and S. M. Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *CoRR*, 2024.

[9] Y. Deng, K. Prasad, R. Fernandez, P. Smolensky, V. Chaudhary, and S. M. Shieber. Implicit chain of thought reasoning via knowledge distillation. *CoRR*, 2023.

[10] T. Gao, X. Yao, and D. Chen. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, 2021.

[11] F. L. Gewers, G. R. Ferreira, H. F. D. Arruda, F. N. Silva, C. H. Comin, D. R. Amancio, and L. d. F. Costa. Principal component analysis: A natural approach to data exploration. *CSUR*, 2021.

[12] J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey. *Int. J. Comput. Vis.*, 2021.

[13] S. Goyal, Z. Ji, A. S. Rawat, A. K. Menon, S. Kumar, and V. Nagarajan. Think before you speak: Training language models with pause tokens. In *ICLR*, 2024.

[14] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv*, 2025.

[15] S. Hao, S. Sukhbaatar, D. Su, X. Li, Z. Hu, J. Weston, and Y. Tian. Training large language models to reason in a continuous latent space. *arXiv*, 2024.

[16] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.

[17] Y. Huang, J. Xu, J. Lai, Z. Jiang, T. Chen, Z. Li, Y. Yao, X. Ma, L. Yang, H. Chen, et al. Advancing transformer architecture in long-context large language models: A comprehensive survey. *arXiv*, 2023.

[18] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv*, 2024.

[19] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al. Openai o1 system card. *arXiv*, 2024.

[20] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. Singh Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, T. Lavril, M.-A. Lachaux, T. Lacroix, L. Ternon, W. El Sayed, T. Wang, G. Bour, E. Bou Hanna, T. Gervet, P. Stock, T. L. Scao, P. Calvez, X. Le, M. Savatier, L. Tan, C. Beguier, G. Delétang, A. Laurençon, A. Tomi, R. Gavrilov, A. El Shikh, E. Albergo, B. Noune, S. Bhagat, A. Ortega Hernandez, N. Beloborodov, C. Schroder, A. Piktus, L. De Viveiros, and U. Chang. Mistral 7b. `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`, 2023.

[21] Z. Jie, T. Q. Luong, X. Zhang, X. Jin, and H. Li. Design of chain-of-thought in math problem solving. *arXiv*, 2023.

[22] S. Krishna. Coin flip dataset. Hugging Face Dataset Repository, 2023.

[23] P. H. Le-Khac, G. Healy, and A. F. Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 2020.

[24] X. Li and J. Li. Aoe: Angle-optimized embeddings for semantic textual similarity. In *ACL*, 2024.

[25] Y. Li, D. McLean, Z. A. Bandar, J. D. O'shea, and K. Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, 2006.

[26] H. Liu, C. Sferrazza, and P. Abbeel. Chain of hindsight aligns language models with feedback. In *ICLR*, 2023.

[27] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith. Linguistic knowledge and transferability of contextual representations. In *NAACL-HLT*, 2019.

[28] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[29] Q. Lyu, S. Havaldar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, and C. Callison-Burch. Faithful chain-of-thought reasoning. In *IJCNLP-AACL*, 2023.

[30] R. Mihalcea, C. Corley, C. Strapparava, et al. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, 2006.

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *NeurIPS*, 2013.

[32] S. Miner, Y. Takashima, S. Han, S. Kouteili, F. Erata, R. Piskac, and S. J. Shapiro. Scheherazade: Evaluating chain-of-thought math reasoning in llms with chain-of-problems. *arXiv*, 2024.

[33] OpenAI. Gpt-4o mini: advancing cost-efficient intelligence, 2024.

[34] OpenAI. Openai o3-mini system card. Technical report, OpenAI, January 2025.

[35] Optimum Team. Mistral-1.1b-testing. `https://huggingface.co/optimum/mistral-1.1b-testing`, 2024. Accessed: 2025-04-28.

[36] L. Parisi. How to run DeepSeek-R1 IQ1_S 1.58bit at 140 token/sec. GitHub Issue, 2025. Issue #1591 in unslothai/unsloth repository.

[37] A. Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv*, 2019.

[38] A. Patel, S. Bhattamishra, and N. Goyal. Are NLP models really able to solve simple math word problems? In *NAACL*, 2021.

[39] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[40] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, 2019.

[41] X. Shen, Y. Wang, X. Shi, Y. Wang, P. Zhao, and J. Gu. Efficient reasoning with hidden thinking. *arXiv*, 2025.

[42] Z. Shen, H. Yan, L. Zhang, Z. Hu, Y. Du, and Y. He. Codi: Compressing chain-of-thought into continuous space via self-distillation. *arXiv*, 2025.

[43] Z. Sprague, F. Yin, J. D. Rodriguez, D. Jiang, M. Wadhwa, P. Singhal, X. Zhao, X. Ye, K. Mahowald, and G. Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv*, 2024.

[44] D. Su, H. Zhu, Y. Xu, J. Jiao, Y. Tian, and Q. Zheng. Token assorted: Mixing latent and text tokens for improved language model reasoning. *arXiv*, 2025.

[45] A. Talmor, J. Herzig, N. Lourie, and J. Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[46] C. Tao, T. Shen, S. Gao, J. Zhang, Z. Li, Z. Tao, and S. Ma. Llms are also effective embedding models: An in-depth overview. *arXiv*, 2024.

[47] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama 2: Open foundation and fine-tuned chat models. `https://huggingface.co/meta-llama/Llama-2-7b-chat-hf`, 2023.

[48] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In *IWANN*, 2005.

[49] Q. Wang, M. J. Zaki, G. Kollias, and V. Kalantzis. Multi-sense embeddings for language models and knowledge distillation. *arXiv*, 2025.

[50] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022.

[51] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020.

[52] M. Xia, T. Gao, Z. Zeng, and D. Chen. Sheared llama: Accelerating language model pre-training via structured pruning. In *ICLR*, 2024.

[53] J. Xu, H. Fei, L. Pan, Q. Liu, M.-L. Lee, and W. Hsu. Faithful logical reasoning via symbolic chain-of-thought. In *ACL*, 2024.

[54] Y. Xu, X. Guo, Z. Zeng, and C. Miao. Softcot: Soft chain-of-thought for efficient reasoning with llms. *CoRR*, 2025.

[55] P. Yu, T. Wang, O. Golovneva, B. AlKhamissi, S. Verma, Z. Jin, G. Ghosh, M. Diab, and A. Celikyilmaz. Alert: Adapt language models to reasoning tasks. In *ACL*, 2023.

[56] W. Zhuo, Y. Sun, X. Wang, L. Zhu, and Y. Yang. Whitenedcse: Whitening-based contrastive learning of sentence embeddings. In *ACL*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] .

   Justification: The abstract and introduction accurately reflect the paper's contributions and scope. The paper clearly states its claim to address two key challenges in Chain-of-Thought (CoT) reasoning: preserving semantic alignment between implicit and explicit reasoning, and optimizing token-level generation speed. These claims are consistent with the results presented throughout the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes] .

   Justification: A dedicated section of limitations is provided in Appendix A.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA] .

Justification: The paper does not include formal theoretical results requiring mathematical proofs. The methodology is explained algorithmically and empirically rather than through formal theorems.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes] .

Justification: The paper provides sufficient information on the datasets used (Section 4.1), model architectures, training procedures, and implementation details (Section 4.1). The hyperparameters and training details are specified in the Implementation Details subsection, and the code is available at the URL provided in the abstract.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes] .

   Justification: The code are available at "https://anonymous.4open.science/r/SemCoT". Additionally, we use standard, publicly available datasets for evaluation.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes] .

   Justification: The paper specifies all the necessary training and test details in Section 4.1, including dataset splits, model architectures, hyperparameters, optimizer settings, and implementation frameworks used (PyTorch and Huggingface).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes] .

   Justification: The paper report error bars for the experimental results. The tables and figures present results with standard deviations across multiple runs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes] .

   Justification: The paper provide information about the computing resources used to conduct the experiments, such as the type of GPUs/CPUs, memory requirements in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes] .

   Justification: The research conform with the NeurIPS Code of Ethics. The work focuses on improving efficiency and effectiveness of language models without apparent ethical concerns. No privacy-sensitive data is used, and the methods don't present obvious risks of harm.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: It is discussed in Broader Impact in Appendix B.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer:[NA] .

Justification: The paper does not present high-risk models or datasets that would require safeguards against misuse. The work focuses on an algorithmic improvement for reasoning efficiency rather than releasing potentially harmful content-generating models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes] .

Justification: The paper properly cites the original sources for the datasets used (GSM8K, SVAMP, MultiArith, CommonsenseQA, and CoinFlip) and the models (Llama-2-7b-chat-hf, Mistral-7B-Instruct-v0.2, Sheared-LLaMA-1.3B, mistral-1.1b-testing) with appropriate references. The licenses are provided in Appendix.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer:[Yes] .

Justification: The assets are in the provided link in the abstract.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA] .

Justification: The paper does not involve crowdsourcing or research with human subjects. The evaluation is conducted on existing benchmark datasets rather than new human-annotated data.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer:[NA] .

    Justification: Since the research does not involve human subjects, IRB approval was not required for this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes] .

    Justification: LLMs are the research objective of the work, since this work aims to improve the LLMs' reasoning ability.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Limitations

While SemCoT demonstrates promising results in improving both the efficiency and effectiveness of Chain-of-Thought reasoning, we acknowledge certain limitations in our current approach. The customized sentence transformer, although effective, requires additional training overhead before deployment, which might be challenging for resource-constrained environments. Furthermore, our evaluation primarily focuses on standard reasoning benchmarks (mathematical, commonsense, and symbolic reasoning); hence, the performance on more specialized domains or extremely long-chain reasoning tasks remains to be explored. Further investigation would benefit the generalizability across different language model architectures beyond the tested ones (Llama-2 and Mistral). Additionally, while we observed consistent performance improvements, there may be specific reasoning patterns where the trade-off between implicit and explicit reasoning is less favorable. Lastly, we acknowledge that implicit reasoning, due to its form of latent token embeddings, reduces the human interpretability of the LLMs' reasoning process. Future work could address these limitations by expanding the evaluation scope, exploring more efficient pre-training strategies for the customized sentence transformer component, and training a specialized implicit reasoning decoder to decode the implicit reasoning generated by the contemplation generator.

B Broader Impact

Our work on SemCoT has several potential positive societal impacts. By improving the efficiency of Chain-of-Thought reasoning in LLMs, we reduce computational costs and energy consumption, leading to lower carbon footprints for AI deployments. This efficiency also facilitates broader access to advanced reasoning capabilities in resource-constrained environments such as mobile devices or underdeveloped regions. Thus, more efficient reasoning enables more applications in time-sensitive domains such as healthcare decision support and emergency response systems. However, we also recognize potential negative impacts. As reasoning becomes more efficient, malicious actors could deploy sophisticated reasoning systems at scale for generating misinformation or conducting automated cyberattacks. The improved efficiency might accelerate the deployment of AI systems before adequate safety measures are established. Furthermore, there is a risk that optimizing for computational efficiency might inadvertently prioritize speed over reasoning quality in specific contexts, potentially leading to errors in critical applications if not properly monitored. We encourage addressing these concerns through responsible deployment practices.

C Implementation Details

Dataset Metadata. We show the metadata of the datasets, including the size of train and test sets, along with the reasoning type, in Table 3. For sample data points from each dataset, see Table 2.

Table 2: Sample questions and answers from benchmark datasets

Dataset	Sample Question	Ans.	GT-reason

(Continued on next page)

(Continued from previous page)

Dataset	Sample Question	Ans.	GT-reason
GSM8K	Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?	18	Janet sells 16 - 3 - 4 = «16-3-4=9»9 duck eggs a day.She makes 9 * 2 = $«9*2=18»18 every day at the farmer's market.
SVAMP	There are 87 oranges and 290 bananas in Philip's collection. If the bananas are organized into 2 groups and oranges are organized into 93 groups. How big is each group of bananas?	145	The question is in type of Common-Division and we solve it by calculating (290.0 / 2.0).
MultiArith	There are 64 students trying out for the school's trivia teams. If 36 of them didn't get picked for the team and the rest were put into 4 groups, how many students would be in each group?	7	To solve the problem step-by-step, we will follow these steps:1. **Determine the number of students who were picked for the team.**2. **Calculate how many students are in each group.** Step 1: Calculate the number of students picked for the teamWe know that there are a total of 64 students trying out for the trivia teams, and 36 of them did not get picked. To find out how many students were picked, we subtract the number of students who didn't get picked from the total number of students:[textNumber of students picked = textTotal students - textStudents not picked]the values:[textNumber of students picked = 64 - 36]this gives:[Number of students picked = 28] Step 2: Calculate how many students are in each groupNow that we know there are 28 students who were picked for the team, we need to divide these students into 4 groups. To find out how many students will be in each group, we divide the number of students picked by the number of groups:[Students per group = fractextNumber of students pickedtextNumber of groups]Substituting the values:[Students per group = frac284]this gives:[Students per group = 7] Conclusion Each group will have **7 students**."

(Continued on next page)

(Continued from previous page)

Dataset	Sample Question	Ans.	GT-reason
Commonsense.	The sanctions against the school were a punishing blow, and they seemed to what the efforts the school had made to change? "label": ["A", "B", "C", "D", "E"], "text": ["ignore", "enforce", "authoritarian", "yell at", "avoid"]	A	To solve this problem, we need to analyze the context of the sentence and the meaning of each answer choice.sentence states that T̈he sanctions against the school were a punishing blow, and they seemed to what the efforts the school had made to change.T̈he key part of the sentence is the phrase s̈eemed to,ẅhich indicates that we are looking for a verb that describes how the sanctions relate to the school's efforts to change., let's evaluate each option:. **ignore** - This suggests that the sanctions did not acknowledge or take into account the school's efforts to change. This could make sense in the context, as sanctions might undermine or overlook positive changes.. **enforce** - This implies that the sanctions would reinforce or support the efforts the school made to change. This does not fit well because sanctions are typically punitive and would not support positive efforts.. **authoritarian** - This is an adjective describing a style of governance or control. It does not fit grammatically in the sentence as it does not complete the phrase s̈eemed to.¨. **yell at** - This is a colloquial expression that implies a verbal reprimand. It does not fit the context of sanctions, which are formal penalties rather than verbal actions.. **avoid** - This suggests that the sanctions would lead to evading or sidestepping the efforts made by the school. This does not fit well either, as sanctions are more about punishment than avoidance.the analysis, option A (ïgnorë) is the most appropriate choice. It indicates that the sanctions overlooked or disregarded the school's efforts to improve, which aligns with the idea of sanctions being a punishing blow., the answer is **A**.
CoinFlip	A coin is heads up. sager does not flip the coin. zyheir flips the coin. Is the coin still heads up?	no	Let's track the state of the coin step by step:1. **Initial State**: The coin is heads up.2. **Sager does not flip the coin**: The state remains heads up.3. **Zyheir flips the coin**: Flipping the coin changes its state from heads up to tails up.the end of these actions, the state of the coin is tails up.Therefore, the answer is **no**, the coin is not still heads up.

Licenses of Existing Assets. For the two adopted LLMs, Llama-2-7b-chat-hf [47] has "Llama 2 Community License Agreement," and Mistral-7B-Instruct-v0.2 [20] has the Apache 2.0 license. For the implementation of the lightweight implicit reasoning generator, both Sheared-LLaMA-1.3B [52] and mistral-1.1b-testing [35] have the Apache 2.0 license. Please see the licenses of the datasets in Table 4.

Hardware Information. We perform all experiments on multiple machines with NVIDIA H100 80GB GPUs running CUDA 12.4.

Hyperparameters Setting. Our `SemCoT` is implemented with PyTorch [37] and Huggingface [51] training pipeline. We list the hyperparameters settings in the GitHub repository (found in utils/utils.py). Table 1 shows the average accuracy and time measurements over three independent rounds for each method. During training, the baselines and `SemCoT` are allotted to five implicit reasoning tokens. For evaluation, they are limited to only one reasoning token. This ensures fair comparison across

Table 3: Metadata for benchmark datasets

Dataset	Train Size	Test Size	Reasoning Type
GSM8K [5]	7,500	1,000	Arithmetic
SVAMP [38]	700	300	Arithmetic
MultiArith [4]	420	180	Arithmetic
CommonsenseQA [45]	9,741	1,140	Commonsense
CoinFlip [22]	20,000	3,330	Symbolic

Table 4: License Information for Hugging Face Datasets

Dataset	License Information
SVAMP	MIT License
MultiArith	CC BY 4.0
CommonsenseQA	MIT License
CoinFlip	MIT License
GSM8K	MIT License

methods, as allowing only one token eliminates the possibility of confounding factors, such as directly providing the answer instead of the reasoning and excessively encoding model knowledge.

Text-paired Dataset. We show examples of the input and reasoning pairs datasets in Table 5.

D Supplementary Experiments

D.1 Ablation Study

In this section, we show the supplementary experiment results for the ablation study. Specifically, we adopt the variants of the `SemCoT` as designed in the Section 4.3 in the main paper and examine their performance on all datasets (GSM8K [5], SVAMP [38], MultiArith [4], CommonsenseQA [45], and CoinFlip [22]) on both LLMs (Llama-2-7b-chat-hf [47] and Mistral-7B-Instruct-v0.2 [20]). The results for Llama-2-7b-chat-hf [47] and Mistral-7B-Instruct-v0.2 [20] are shown in the Fig 6 and Fig. 7 respectievly. From the two figures, we observe that our `SemCoT` performs better than all variants in almost all experiment configurations (i.e., the composition of datasets and LLMs), and the three observations from Section 4.3 also hold.

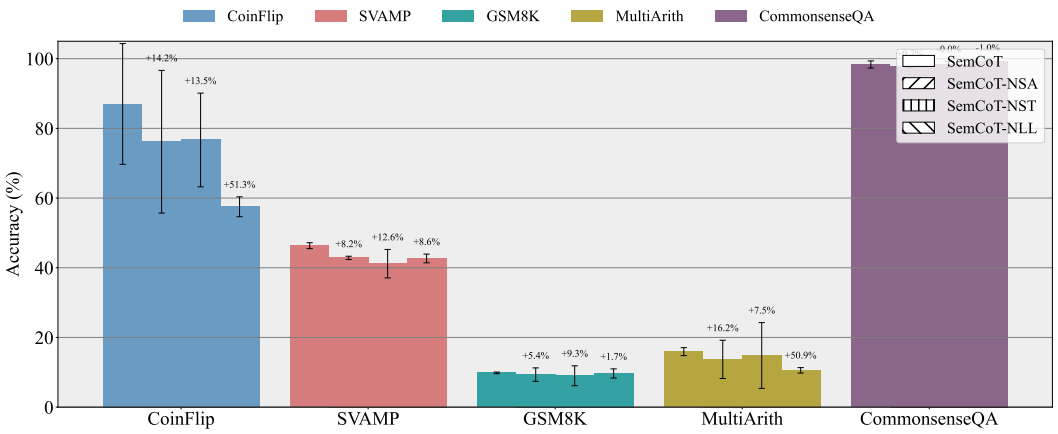


Figure 6: Ablation study results on Llama-2-7b-chat-hf [47]. We show the performance of our `SemCoT` compared to its three variants on all five adopted datasets in the Section 4.1 of the paper.

Table 5: Text samples from the text-paired datasets for each dataset

Dataset	Full Reasoning	Condensed Reasoning
CoinFlip	Let's track the state of the coin step by step:1. Initially, the coin is heads up.2. Mailey does not flip the coin, so the state of the coin remains heads up.3. Maurisa does not flip the coin, so the state of the coin remains heads up.At the end of these steps, the coin is still heads up.Final answer: yes.	Coin remains heads up; neither Mailey nor Maurisa flipped it.
Common	To determine where in Southern Europe you would find many canals, let's analyze each of the answer choices:\n\nA. **Michigan** - This is a state in the United States, not in Southern Europe. Therefore, it is not a suitable answer.\n\nB. **New York** - This is a state in the United States, specifically in the northeastern part of the country. Like Michigan, it is not in Southern Europe, so this option is also not appropriate.\n\nC. **Amsterdam** - While Amsterdam is known for its extensive canal system, it is located in the Netherlands, which is in Northern Europe, not Southern Europe. Thus, this option does not fit the criteria.\n\nD. **Venice** - Venice is a city in Italy, which is located in Southern Europe. It is famous for its canals, making this option a strong candidate for the answer.\n\nE. **Bridge** - This is a general term and does not refer to a specific location. While bridges may be found in areas with canals, it does not answer the question of where in Southern Europe you would find many canals.\n\nAfter analyzing all the options, the only one that meets the criteria of being in Southern Europe and known for its canals is:\n\n**D. Venice** \n\nThus, the correct answer is D.\n#### D	Venice, in Southern Europe, is famous for its canals.
GSM8K	Natalia sold 48/2 = «48/2=24»24 clips in May.\nNatalia sold 48+24 = «48+24=72»72 clips altogether in April and May.	Natalia sold 48 clips in April, 24 in May; total 72.
MultiArith	To determine how much money the magician earned from selling magic card decks, we can follow these steps:\n\n### Step 1: Determine the number of decks sold\nThe magician started with 5 decks and ended with 3 decks. To find out how many decks he sold, we subtract the number of decks he has left from the number he started with:\n\n\\[\n\\textDecks sold = \\textInitial decks - \\textRemaining decks\n\\]\n\\[\n\\textDecks sold = 5 - 3 = 2\n\\]\n\n### Step 2: Calculate the total earnings from the decks sold\nEach deck was sold for 2 dollars. To find out how much money he earned from selling the decks, we multiply the number of decks sold by the price per deck:\n\n\\[\n\\textTotal earnings = \\textDecks sold \\times \\textPrice per deck\n\\]\n\\[\n\\textTotal earnings = 2 \\times 2 = 4\n\\]\n\n### Conclusion\nThe magician earned a total of 4 dollars from selling the magic card decks.\n\n\\[\n\\textTotal earnings = 4 \\text dollars\n\\]	Magician sold 2 decks for $4; earnings totaled $4.
SVAMP	The question is in type of Common-Division and we solve it by calculating (290.0 / 2.0)	Divide 290 bananas by 2 groups to find group size.

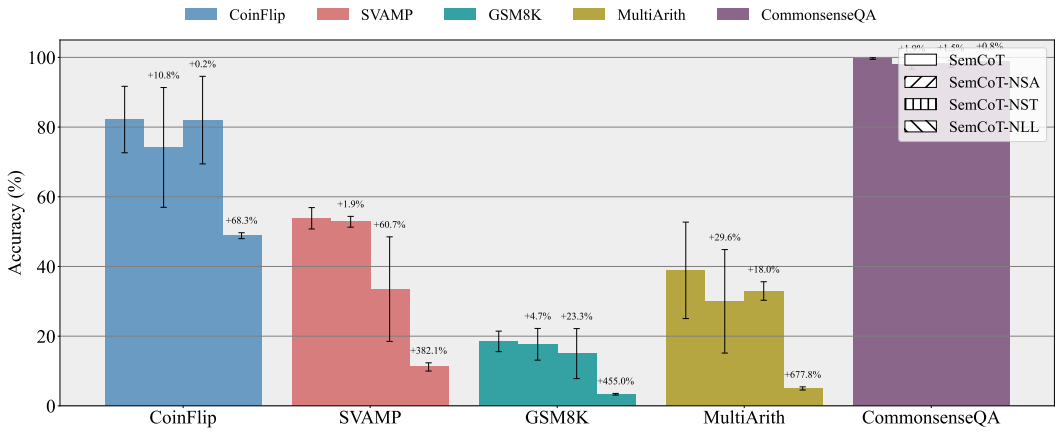


Figure 7: Ablation study results on Mistral-7B-Instruct-v0.2 [20]. We show the performance of our SemCoT compared to its three variants on all five adopted datasets in the Section 4.1 of the paper.

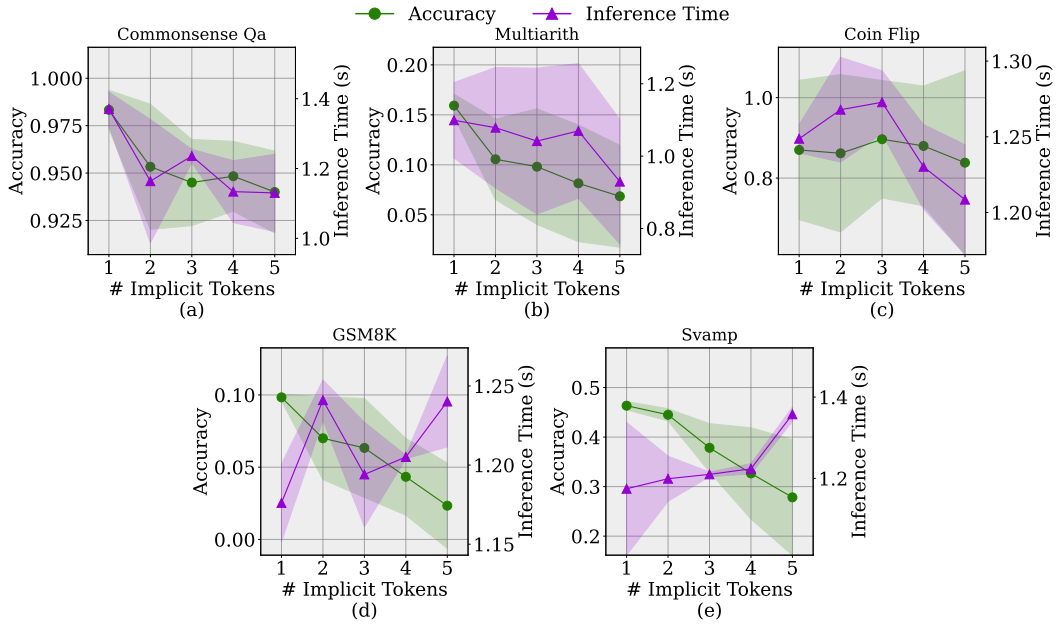


Figure 8: Parameter analysis experiment results for Llama-2-7b-chat-hf [47]. The accuracy and inference time of our method using Llama when varying the number of implicit tokens

D.2 Parameter Analysis

Here, we display the supplementary experiment results for parameter analysis. Specifically, we follow the design of Section 4.4 to examine the number of implicit tokens utilized during evaluation. We conduct experiments on all adopted datasets and LLMs; the results for Llama-2-7b-chat-hf [47] and Mistral-7B-Instruct-v0.2 [20] are shown in Fig. 8 and Fig. 9, respectively. We find generally consistent observations in Section 4.4 of the main paper.

D.3 Case Study: Semantic Alignment Analysis

We conduct extensive experiments to show that our method helps maintain the semantic alignment between ground truth and implicit reasoning. As declared in Section 4.5, we randomly pick three samples from each dataset and generate semantically aligned queries for each query twenty times. Then we design the experiments to show that our SemCoT achieves semantically-aligned reasoning

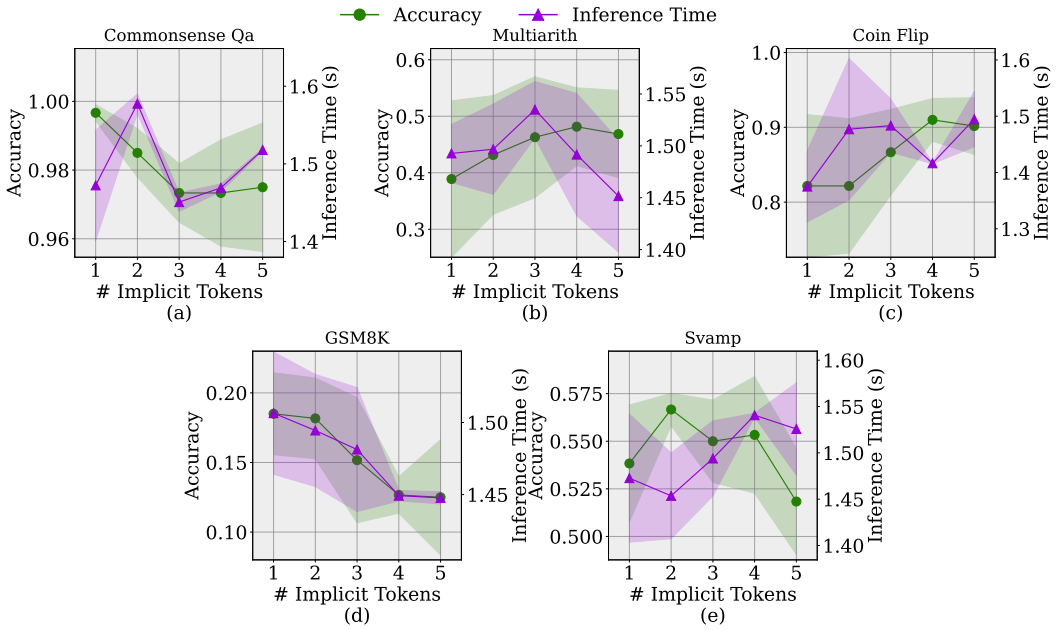


Figure 9: Parameter analysis experiment results for Mistral-7B-Instruct-v0.2 [20]. The accuracy and inference time of our method using Mistral when varying the number of implicit tokens

based on three assumptions: (1) semantically aligned queries will induce semantically aligned reasoning; (2) Different queries and their semantic aligned variants should be well-separated in implicit reasoning space to differentiate their semantics in the implicit reasoning space; (3) Samples from different semantic domains should be more separated than those samples from the same semantic domain to differentiate their semantics domain in the implicit reasoning space. Here, the "semantic domain" means the semantic type of the reasoning task. For example, GSM8K [5], SVAMP [38], MultiArith [4] are for mathematical reasoning, they are in the same semantic domain. However, CommonsenseQA [45] is for commonsense reasoning. Thus, the samples within GSM8K [5], SVAMP [38], and MultiArith [4] are in the same semantic domain, but their samples are in different semantic domains with CommensenseQA. The results are shown in the Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16, and Fig. 17. For each figure, each subplot of the first row is the implicit reasoning tokens of all samples and their semantic-aligned variants from one dataset. In the second row, each subplot is the ith sample of all datasets, along with their semantic-aligned variants. We can observe across the five figures that our `SemCoT` is the only model to maintain low implicit reasoning variance under semantic-aligned queries and appropriately separate different samples in the implicit reasoning space. Meanwhile, we can also see that our `SemCoT` recognizes semantic domains because the implicit reasoning for mathematical reasoning is reasonably separate from other domains, such as commonsense reasoning.

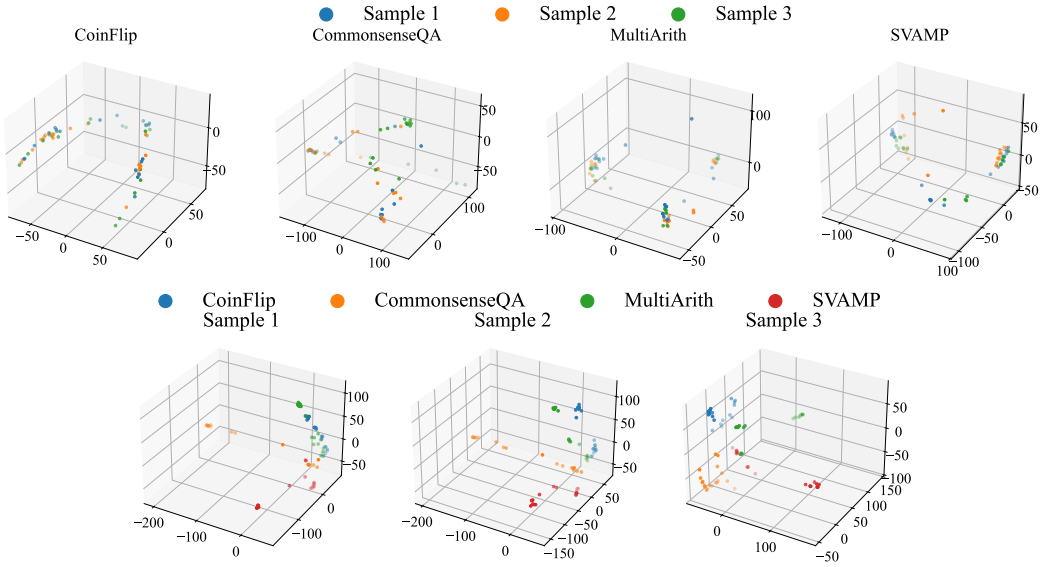


Figure 10: Case study for `SoftCoT` [54] on Llama-2-7b-chat-hf [47] .

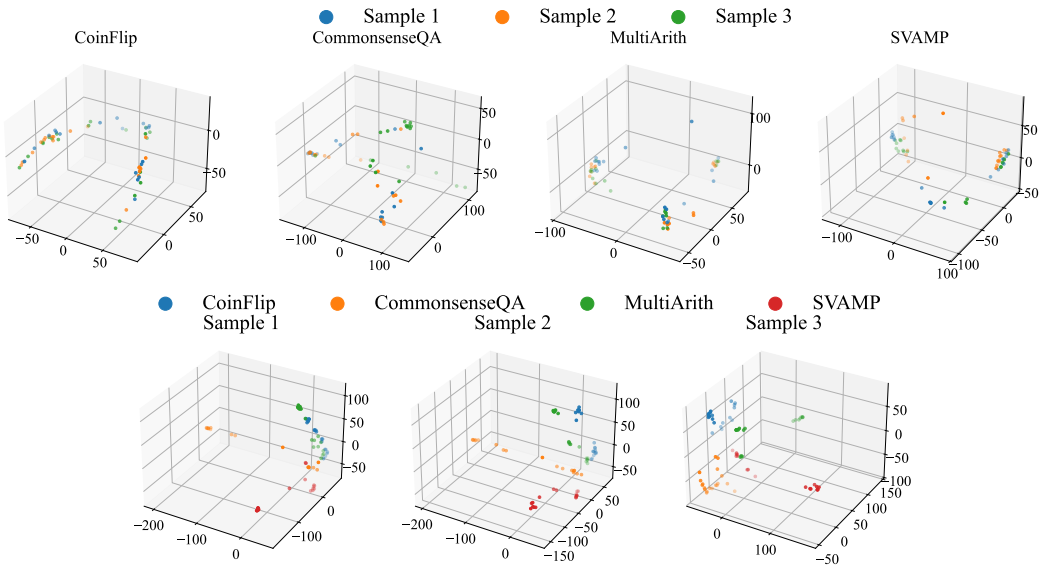


Figure 11: Case study for `SoftCoT` [54] on Mistral-7B-Instruct-v0.2 [20].

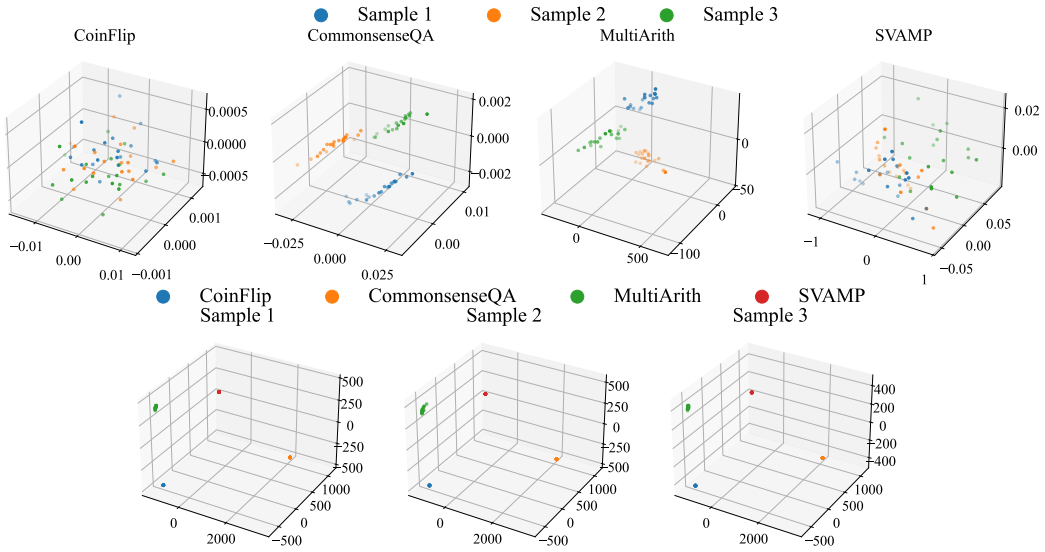


Figure 12: Case study for SemCoT on Llama-2-7b-chat-hf [47] .

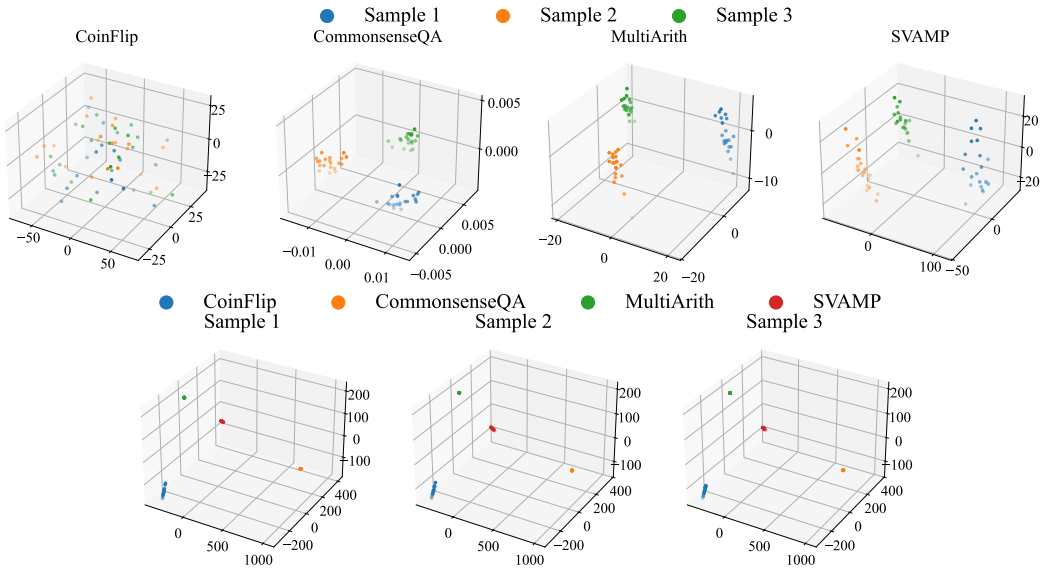


Figure 13: Case study for SemCoT on Mistral-7B-Instruct-v0.2 [20].

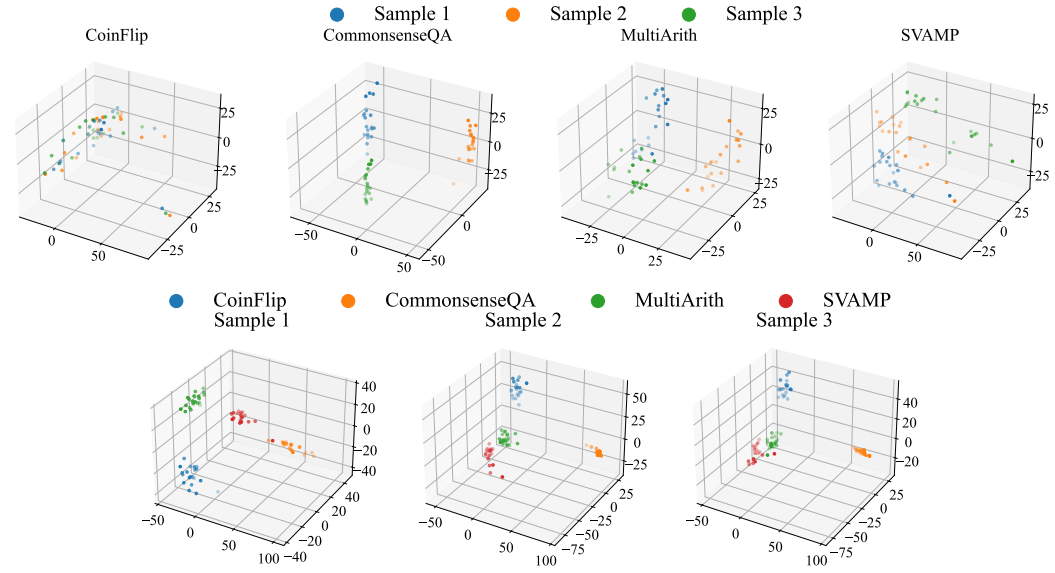


Figure 14: Case study for COCONUT [54] on Llama-2-7b-chat-hf [47] .

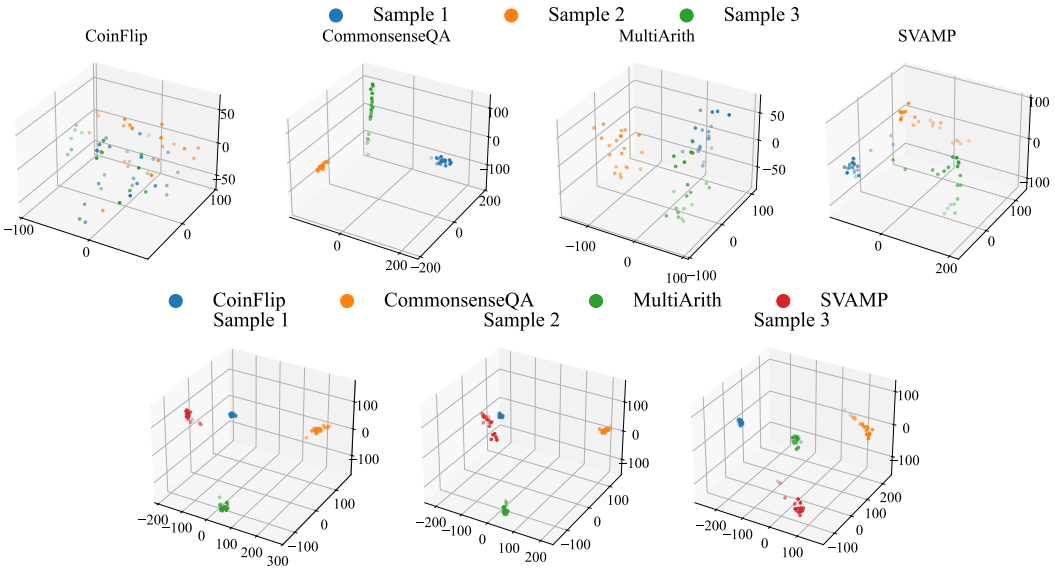


Figure 15: Case study for COCONUT [54] on Mistral-7B-Instruct-v0.2 [20].

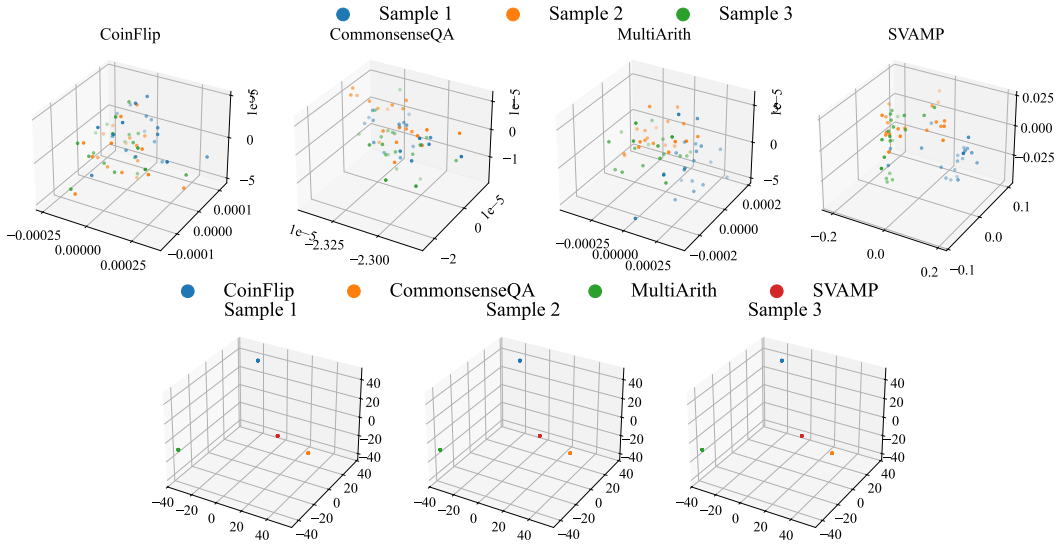


Figure 16: Case study for CODI [42] on Llama-2-7b-chat-hf [47].

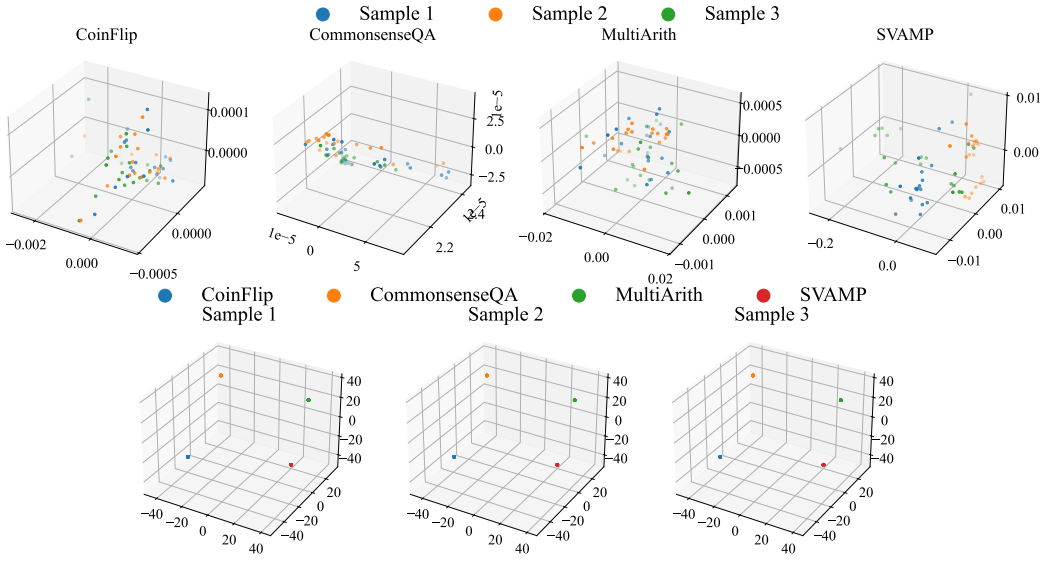


Figure 17: Case study for CODI [42] on Mistral-7B-Instruct-v0.2 [20].