

473 A Limitations

474 While SemCoT demonstrates promising results in improving both the efficiency and effectiveness
475 of Chain-of-Thought reasoning, we acknowledge certain limitations in our current approach. The
476 customized sentence transformer, although effective, requires additional training overhead before
477 deployment, which might be challenging for resource-constrained environments. Furthermore, our
478 evaluation primarily focuses on standard reasoning benchmarks (mathematical, commonsense, and
479 symbolic reasoning); hence, the performance on more specialized domains or extremely long-chain
480 reasoning tasks remains to be explored. Further investigation would benefit the generalizability across
481 different language model architectures beyond the tested ones (Llama-2 and Mistral). Additionally,
482 while we observed consistent performance improvements, there may be specific reasoning patterns
483 where the trade-off between implicit and explicit reasoning is less favorable. Future work could
484 address these limitations by expanding the evaluation scope and exploring more efficient pre-training
485 strategies for the customized sentence transformer component.

486 B Broader Impact

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

500 C Implementation Details

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

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

507 **Hardware Information.** We perform experiments on a machine with two 24-core AMD EPYC
508 7473X processors (48 cores total) running Linux (x86_64 architecture). It has four NVIDIA A100
509 80GB PCIe GPUs. The system ran NVIDIA driver version 550.54.14 and CUDA 12.4.

510 **Hyperparameters Setting.** Our SemCoT is implemented with PyTorch [37] and Huggingface [51]
511 training pipeline. We list the hyperparameters settings specific for each dataset and LLM below. Here,
512 we display the hyperparameter configuration for each dataset-model combination:

513 The configuration for the SVAMP dataset on Llama uses a linear learning rate of 0.0001 and a weight
514 decay of 0.001 over 5 epochs to warm up the linear layer in the sentence transformer. Then, the
515 sentence transformer is trained with a learning rate of 1e-7, weight decay of 1e-5, and runs for 2
516 epochs. The lightweight LLM uses a learning rate of 0.0001 and weight decay of 0.001, and trains
517 the linear component for 5 epochs as warm up. The lightweight LLM then trains for 2 epochs at a
518 learning rate of 1e-7 and weight decay of 1e-5. The configuration for the SVAMP dataset on Mistral
519 uses a linear learning rate of 0.01 and a weight decay of 0.0001 over 3 epochs to warm up the linear
520 layer in the sentence transformer. Then, the sentence transformer is trained with a learning rate of
521 1e-5, weight decay of 1e-5, and runs for 2 epochs. The lightweight LLM uses a learning rate of
522 0.0001 and weight decay of 0.01, and trains the linear component for 3 epochs as warm up. The
523 lightweight LLM then trains for 1 epoch at a learning rate of 1e-5 and weight decay of 0.001.

Table 2: Sample questions and answers from benchmark datasets

Dataset	Sample Question	Ans.	GT-reason
GSM8K	Janet’s ducks lay 16 eggs per day. She eats 3 eggs per day and sells the rest at the farmers’ market daily for \$2 per egg. How much money does she make in a week?	182	Janet sells $16 - 3 = 13$ eggs per day. She makes $13 \times \$2 = \26 per day. In a week, she makes $\$26 \times 7 = \182 .
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	Philip has 290 bananas. If they are divided into 2 groups, each group has $290 \div 2 = 145$ bananas.
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	$64 - 36 = 28$ students were picked. $28 \div 4 = 7$ students per group.
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	The sentence implies that the sanctions did not recognize the school’s efforts, so the best fit is "ignore".
CoinFlip	A coin is heads up. sager does not flip the coin. zyheir flips the coin. Is the coin still heads up?	no	The coin starts heads up. Sager does nothing, so the coin stays the same. Zyheir flips the coin, so its state changes — it is no longer guaranteed to be heads. Thus, the answer is "no".

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	Apache 2.0
CommonsenseQA	MIT License
CoinFlip	MIT License
GSM8K	MIT License

The configuration for the Multiarith dataset on Llama uses a linear learning rate of 0.0001 and a weight decay of 0.001 over 5 epochs to warm up the linear layer in the sentence transformer. Then, the sentence transformer is trained with a learning rate of $1e-7$, weight decay of 0.001, and runs for 1 epoch. The lightweight LLM uses a learning rate of 0.001 and weight decay of 0.01, and trains the linear component for 3 epochs as warm up. The lightweight LLM then trains for 2 epochs at a learning rate of $1e-5$ and weight decay of 0.001. The configuration for the Multiarith dataset on Mistral uses a linear learning rate of 0.0001 and a weight decay of 0.01 over 1 epoch to warm up the linear layer in the sentence transformer. Then, the sentence transformer is trained with a learning rate of $1e-7$, weight decay of 0.001, and runs for 2 epochs. The lightweight LLM uses a learning rate of 0.001 and weight decay of 0.0001, and trains the linear component for 5 epochs as warm up. The lightweight LLM then trains for 2 epochs at a learning rate of $1e-7$ and weight decay of 0.001.

The configuration for the CoinFlip dataset on Llama uses a linear learning rate of 0.001 and a weight decay of 0.0001 over 3 epochs to warm up the linear layer in the sentence transformer. Then, the sentence transformer is trained with a learning rate of $1e-5$, weight decay of 0.001, and runs for 1 epoch. The lightweight LLM uses a learning rate of 0.01 and weight decay of 0.01, and trains the linear component for 1 epoch as warm up. The lightweight LLM then trains for 2 epochs at a learning rate of $1e-7$ and weight decay of 0.001. The configuration for the CoinFlip dataset on Mistral uses a linear learning rate of 0.0001 and a weight decay of 0.001 over 5 epochs to warm up the linear layer in the sentence transformer. Then, the sentence transformer is trained with a learning rate of $1e-5$, weight decay of 0.001, and runs for 1 epoch. The lightweight LLM uses a learning rate of 0.001 and weight decay of 0.001, and trains the linear component for 3 epochs as warm up. The lightweight LLM then trains for 1 epoch at a learning rate of $1e-5$ and weight decay of 0.001.

The configuration for the GSM8K dataset on Llama uses a linear learning rate of 0.001 and a weight decay of 0.01 over 3 epochs to warm up the linear layer in the sentence transformer. Then, the sentence transformer is trained with a learning rate of $1e-5$, weight decay of 0.001, and runs for 1 epoch. The lightweight LLM uses a learning rate of 0.0001 and weight decay of 0.01, and trains the linear component for 3 epochs as warm up. The lightweight LLM then trains for 1 epoch at a learning rate of $1e-5$ and weight decay of 0.001. The configuration for the GSM8K dataset on Mistral uses a linear learning rate of 0.001 and a weight decay of 0.001 over 1 epoch to warm up the linear layer in the sentence transformer. Then, the sentence transformer is trained with a learning rate of $1e-7$, weight decay of $1e-5$, and runs for 1 epoch. The lightweight LLM uses a learning rate of 0.01 and weight decay of 0.01, and trains the linear component for 3 epochs as warm up. The lightweight LLM then trains for 2 epochs at a learning rate of $1e-7$ and weight decay of 0.001.

The configuration for the CommonsenseQA dataset on Llama uses a linear learning rate of 0.0001 and a weight decay of 0.001 over 1 epoch to warm up the linear layer in the sentence transformer. Then, the sentence transformer is trained with a learning rate of $1e-5$, weight decay of $1e-5$, and runs for 2 epochs. The lightweight LLM uses a learning rate of 0.01 and weight decay of 0.01, and trains the linear component for 3 epochs as warm up. The lightweight LLM then trains for 1 epoch at a learning rate of $1e-5$ and weight decay of 0.001. The configuration for the CommonsenseQA dataset on Mistral uses a linear learning rate of 0.01 and a weight decay of 0.001 over 1 epoch to warm up the linear layer in the sentence transformer. Then, the sentence transformer is trained with a learning rate of $1e-5$, weight decay of 0.001, and runs for 1 epoch. The lightweight LLM uses a learning rate of 0.01 and weight decay of 0.01, and trains the linear component for 3 epochs as warm up. The lightweight LLM then trains for 1 epoch at a learning rate of $1e-5$ and weight decay of 0.001.

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 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.

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

Dataset	Full Reasoning	Condensed Reasoning
CoinFlip	Let's track the state of the coin step by step:\n\n1. Initially, the coin is heads up.\n2. Mailey does not flip the coin. (State remains heads up)\n3. Maurisa does not flip the coin. (State remains heads up)\n\nAt the end of these steps, the coin is still heads up.\n\nFinal answer: yes\n#### yes	Coin remains heads up; neither person flips 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 $48/2 = \ll 48/2=24 \gg$ 24 clips in May.\nNatalia sold $48+24 = \ll 48+24=72 \gg$ 72 clips altogether in April and May.\n#### 72	Natalia sold 48 clips in April and 24 in May, totaling 72 clips.
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 $\text{Decks sold} = \text{Initial decks} - \text{Remaining decks}$ \n $= 5 - 3 = 2$ \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 $\text{Total earnings} = \text{Decks sold} \times \text{Price per deck}$ \n $= 2 \times 2 = 4$ \n\n#### Conclusion\nThe magician earned a total of 4 dollars from selling the magic card decks.\n\n $\text{Total earnings} = 4 \text{ dollars}$ \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 ($290.0 / 2.0$)\n#### 14	Divide 290 bananas by 2 groups; each group has 145 bananas.

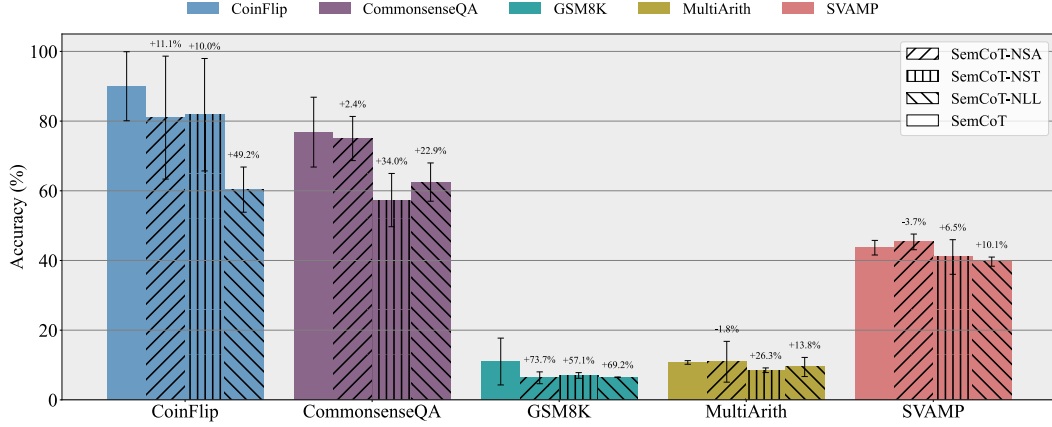


Figure 6: Ablation study results on Llama-2-7b-chat-hf [47]. We show the performance of our SemCoT compared to its three variants on all five adopted datasets in the Section 4.1 of the paper.

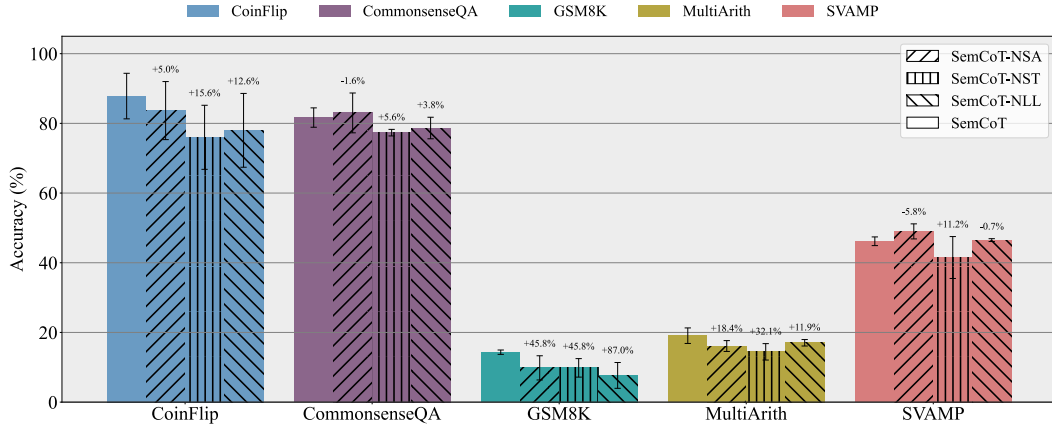


Figure 7: Ablation study results on Mistral-7B-Instruct-v0.2 [20]. We show the performance of our SemCoT compared to its three variants on all five adopted datasets in the Section 4.1 of the paper.

574 D Supplementary Experiments

575 D.1 Ablation Study

576 In this section, we show the supplementary experiment results for the ablation study. Specifically, we
 577 adopt the variants of the SemCoT as designed in the Section 4.3 in the main paper and examine their
 578 performance on all datasets (GSM8K [5], SVAMP [38], MultiArith [4], commonsense reasoning
 579 dataset CommonsenseQA [45], CommonsenseQA [45], and CoinFlip [22]) on both LLMs (Llama-
 580 2-7b-chat-hf [47] and Mistral-7B-Instruct-v0.2 [20]). The results for Llama-2-7b-chat-hf [47] and
 581 Mistral-7B-Instruct-v0.2 [20] are shown in the Fig 6 and Fig. 7 respectively. From the two figures,
 582 we observe that our SemCoT performs better than all variants in almost all experiment configurations
 583 (i.e., the composition of datasets and LLMs), and the three observations from Section 4.3 also hold.

584 D.2 Parameter Analysis

585 Here, we display the supplementary experiment results for parameter analysis. Specifically, we
 586 follow the design of Section 4.4 to examine two significant parameters: (1) the α controlling the
 587 effect of the reasoning semantic alignment and answer correctness and (2) the number of implicit
 588 tokens utilized during evaluation. We conduct experiments on all adopted datasets and LLMs; the
 589 results for Llama-2-7b-chat-hf [47] and Mistral-7B-Instruct-v0.2 [20] are shown in Fig. 8 and Fig. 9
 590 respectively. We find consistent observations in Section 4.4 of the main paper.

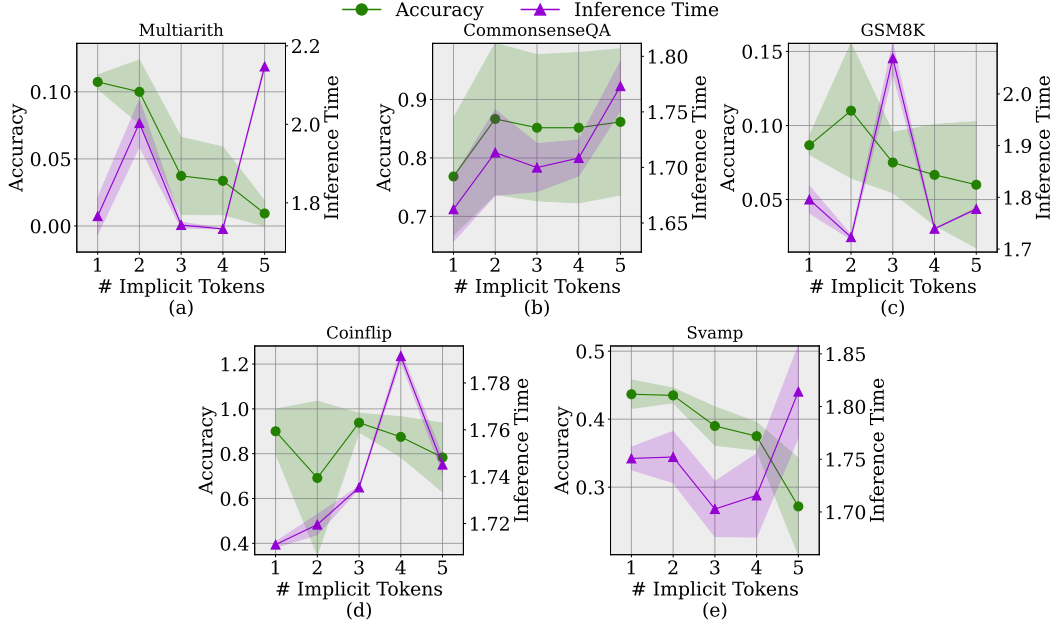


Figure 8: Parameter analysis experiment results for Llama-2-7b-chat-hf [47]. The accuracy and inference time of our method using Llama when varying the number of implicit tokens

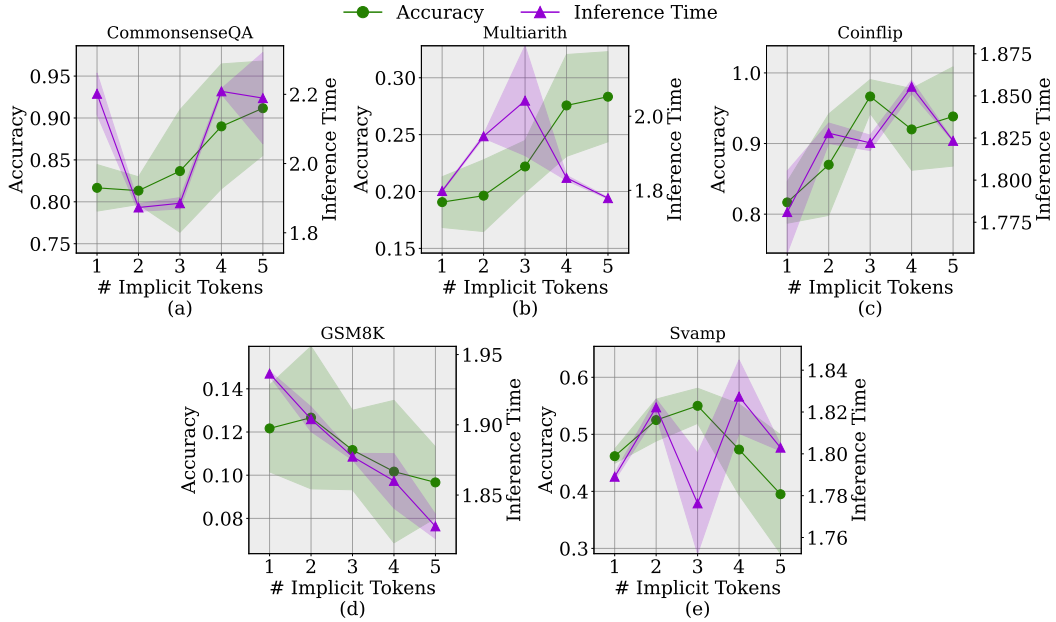


Figure 9: Parameter analysis experiment results for Mistral-7B-Instruct-v0.2 [20]. The accuracy and inference time of our method using Mistral when varying the number of implicit tokens

Table 6: Samples picked for the case study for SVAMP dataset (see samples for other datasets in <https://anonymous.4open.science/r/SemCoT>).

Samples
Winter is almost here and most animals are migrating to warmer countries. There are 41 bird families living near the mountain. If 35 bird families flew away to asia and 62 bird families flew away to africa How many more bird families flew away to africa than those that flew away to asia?
26 children were riding on the bus. At the bus stop 38 more children got on the bus. How many children are on the bus now?
Paco had 41 cookies. He gave 9 cookies to his friend and ate 18 cookies. How many more cookies did he eat than those he gave to his friend?

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 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 CommonsenseQA. The results are shown in the Fig. 10, Fig. 11, Fig. 12, Fig. 13, Fig. 14, and Fig. 15. 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 i th 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.

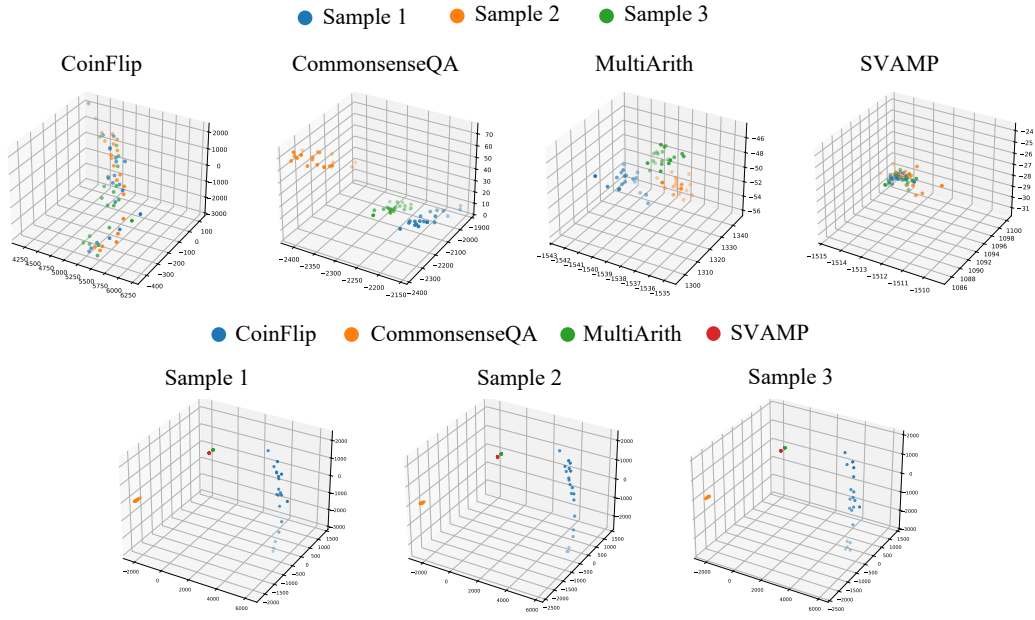


Figure 10: Case study for SemCoT on Llama-2-7b-chat-hf [47] .

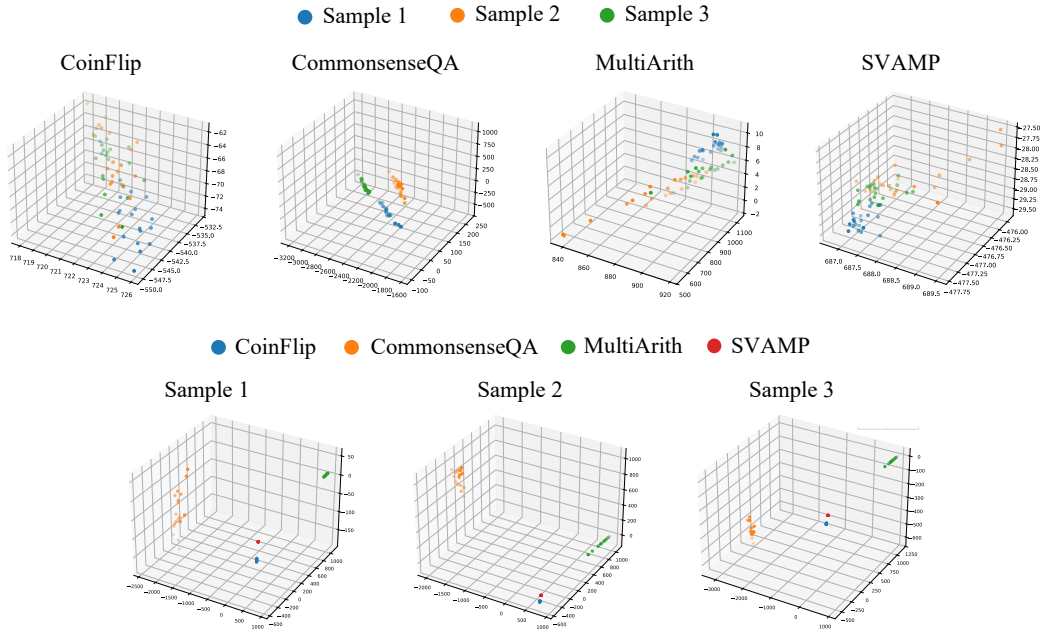


Figure 11: Case study for SemCoT on Mistral-7B-Instruct-v0.2 [20].

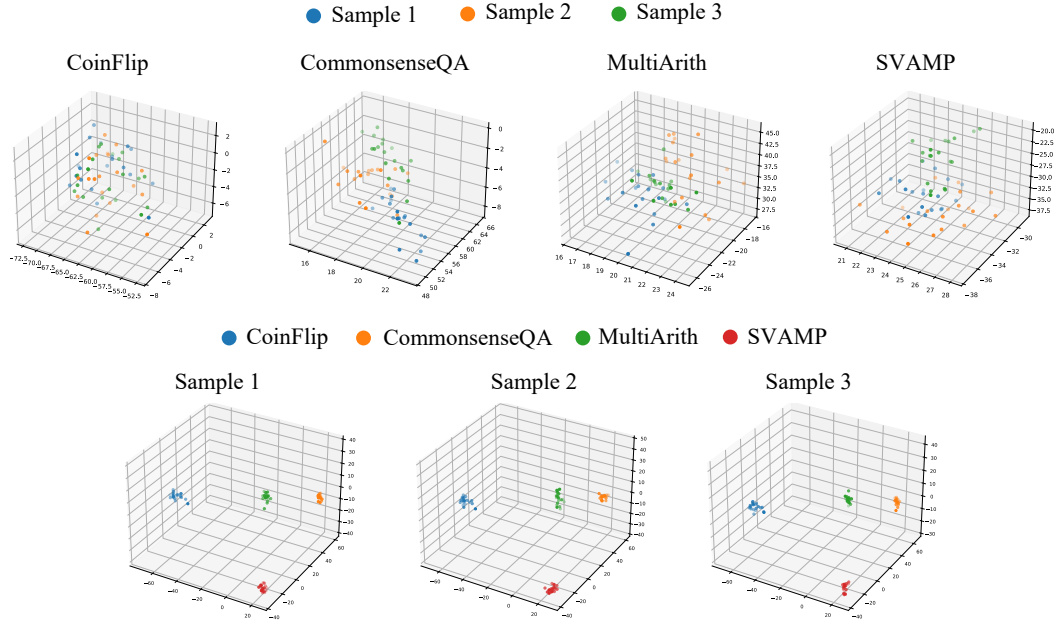


Figure 12: Case study for COCONUT [54] on Llama-2-7b-chat-hf [47] .

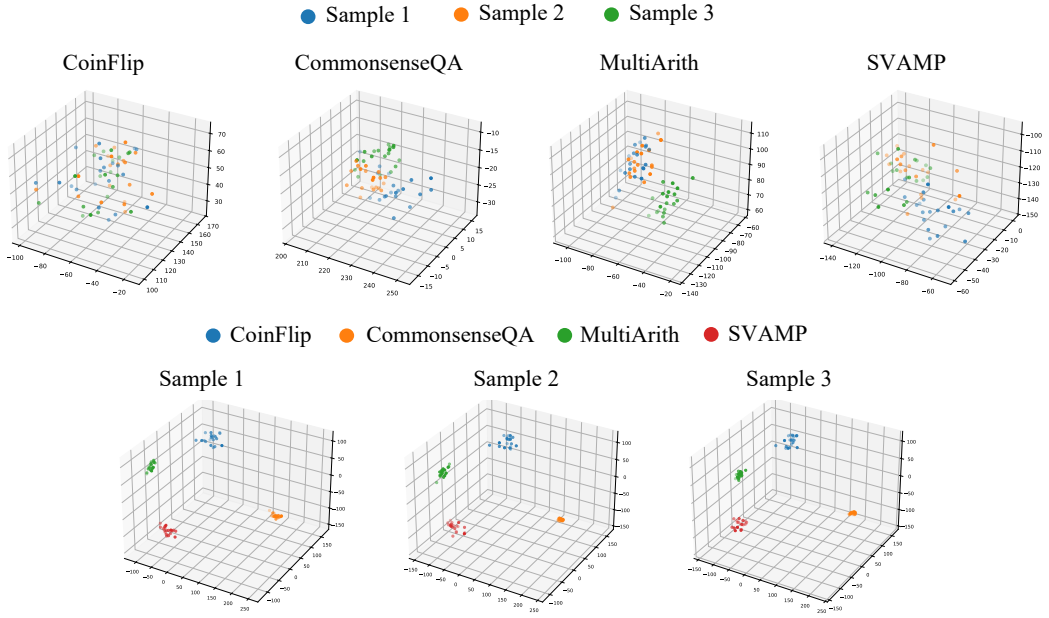


Figure 13: Case study for COCONUT [54] on Mistral-7B-Instruct-v0.2 [20].

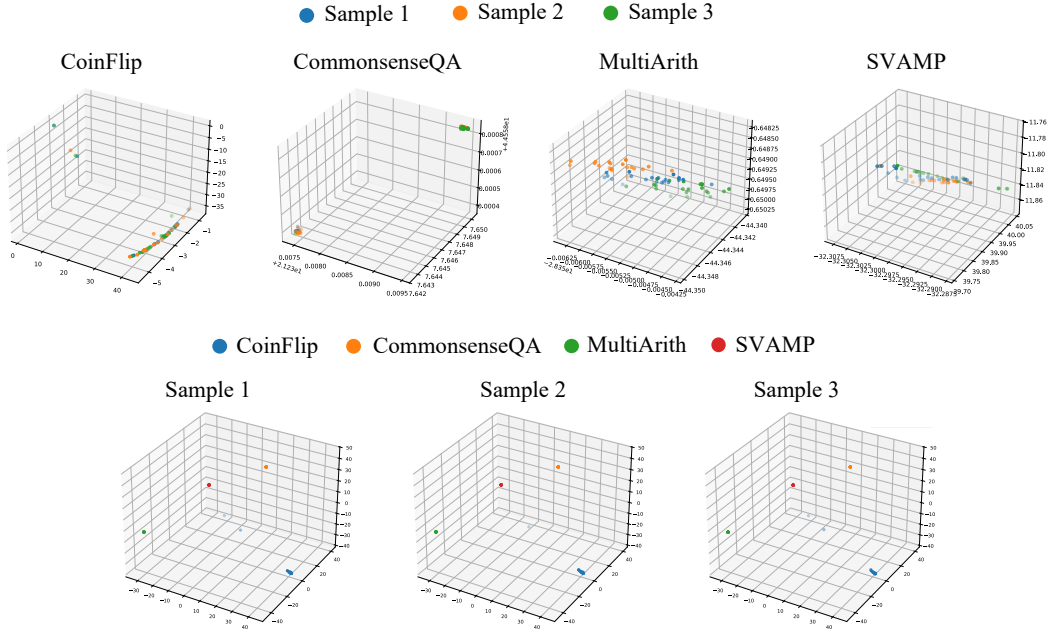


Figure 14: Case study for CODI [42] on Llama-2-7b-chat-hf [47].

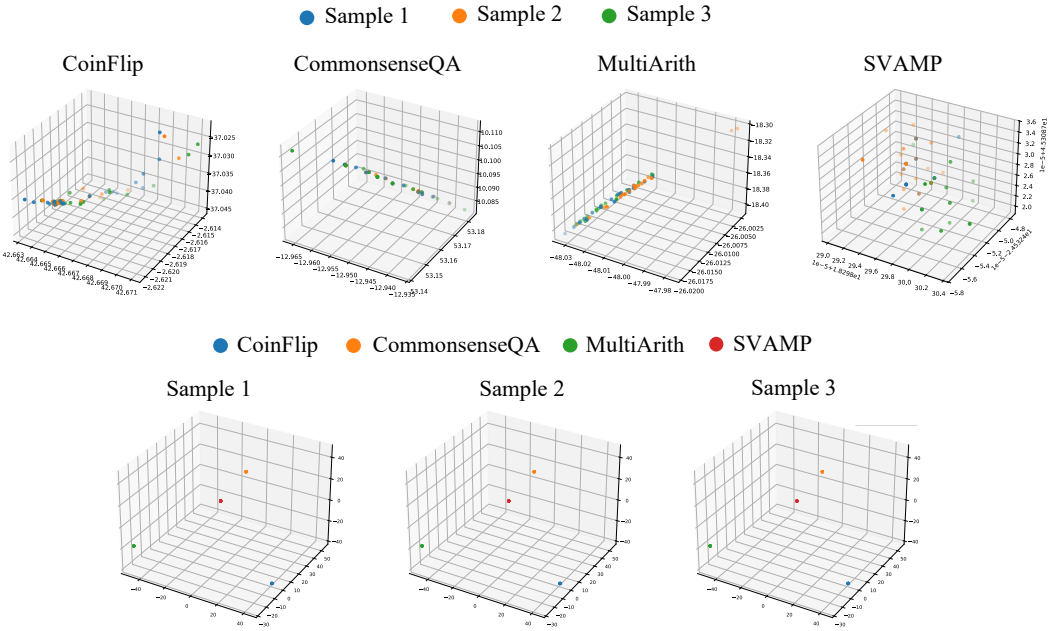


Figure 15: Case study for CODI [42] on Mistral-7B-Instruct-v0.2 [20].

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. Terner, 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. Gavrilo, 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.