Distillation Contrastive Decoding: Improving LLMs Reasoning with Contrastive Decoding and Distillation

Anonymous ACL submission

Abstract

In this work, we propose a novel approach called Distillation Contrastive Decoding to enhance the reasoning capabilities of Large Language Models (LLMs) during inference. Different from previous approaches that used smaller *amateur* models or analyzed differences in hidden states, DCD leverages contrastive chain-of-thought prompting and advanced distillation techniques, such as Dropout and Quantization, to address the limitations of Contrastive Decoding, which often require both an *expert* and an *amateur* model, thereby increasing computational demands. By integrating contrastive prompts with distillation, DCD obviates the need for an amateur model and reduces memory usage. Our evaluations show that DCD significantly improves LLM performance across various reasoning benchmarks, outperforming existing methods and achieving state-of-the-art results in both GSM8K and StrategyQA.[1]

1 Introduction

Reasoning capabilities in large language models (LLMs) refer to the models' ability to analyze, understand, and infer information, mirroring human-like logical reasoning. Recently, the reasoning skills of LLMs have seen substantial advancements, showcasing their vast potential in various natural language processing applications (Brown et al., 2020a). While some works aim at enhancing models through advanced training techniques and architectures (Touvron et al., 2023a; Jiang et al., 2023; Bai et al., 2023), others focus on augmenting the internal capabilities of the models (Zou et al., 2023; Bricken et al., 2023). Beyond the realms of model training and augmentation, further research explores innovative methods to amplify LLMs' efficiency during inference (Li et al., 2023b,a; Chuang et al., 2023). In this work, we introduce Distillation Contrastive Decoding (DCD), a method designed to enhance the reasoning abilities of LLMs during inference by leveraging contrastive chain-of-thought prompts and distillation.

Distillation Contrastive Decoding (DCD) builds on recent advancements in enhancing the reasoning capabilities of LLMs through Contrastive Decoding (CD) (O'Brien and Lewis, 2023) and Contrastive Chain-of-Thought Prompting (CP) (Chia et al., 2023). These methods utilize contrasting elements to reduce reasoning errors in text generation, thereby improving task performance. A principal motivation behind DCD is to address two common limitations of CD. Firstly, CD typically requires a smaller *amateur* LLM within the same family to evaluate the outputs of the primary LLM. This prerequisite poses a challenge, particularly for the small-sized models as there may not be available smaller models with identical vocabularies. This challenge is notably present in cases such as Mistral-7B (Jiang et al., 2023) and DeepSeek-7B (DeepSeek-AI et al., 2024), where smaller models are unavailable. The second limitation with CD is the requirement to simultaneously load two models into memory: an expert and an amateur model, which significantly increases computational resource demands. An example of this is using Llama2-7b as the *amateur* model and Llama2-13b as the *expert* model, highlighting the resource-intensive nature of the CD approach.

Our findings demonstrate that DCD surpasses current methodologies in enhancing chain-of-thought reasoning within LLMs. Specifically, on the GSM8K benchmark, which comprises grade-school level word math problems, DCD elevates the performance of Llama2 models by as much as 3.79% and exhibits a performance increase of 1.89% over CD. On StrategyQA, DCD outperforms all existing methods by a significant gap. Notably, it helps Llama2 models in achieving performance enhancements of up to 5.9%. We ob-

[1] Code is available at https://github.com/xxx

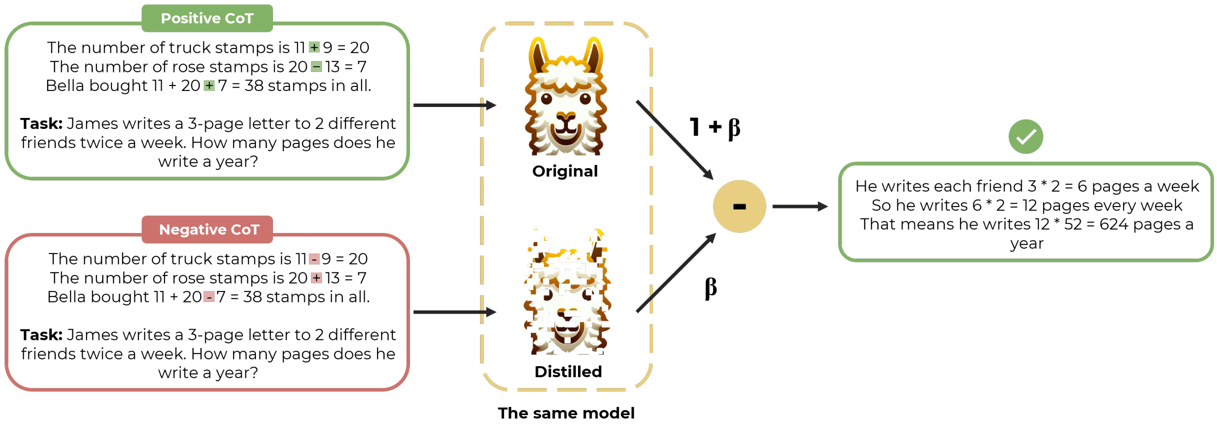

Figure 1: An overview of Distillation Contrastive Decoding method. Valid chain-of-thought demonstrations as well as the query will be sent to an LLM, while invalid chain-of-thought demonstrations and the query will be sent into a distilled version of the model. We will then use this logit information to enhance the reasoning decoding process.

serve marked improvements in both arithmetic and commonsense reasoning tasks when DCD is applied to Mistral-7B, known for its robust foundational knowledge and high scores on the MMLU benchmark (Hendrycks et al., 2020), suggesting that DCD could bring such widespread improvements to much stronger models.

In this study, we introduce a novel methodology termed Distillation Contrastive Decoding (DCD) aimed at augmenting the reasoning capabilities of Large Language Models (LLMs) during inference. Diverging from prior strategies that relied on employing smaller "amateur" models or analyzing differences in hidden states, DCD capitalizes on the synergy between contrastive chain-of-thought prompting and advanced distillation techniques, such as Dropout and Quantization, to surmount the constraints associated with Contrastive Decoding. These constraints typically necessitate the simultaneous use of both an expert and an amateur model, leading to heightened computational demands. By amalgamating contrastive prompts with distillation processes, DCD eliminates the dependency on amateur models and diminishes memory consumption. Our comprehensive evaluations demonstrate that DCD markedly enhances LLM performance across a spectrum of reasoning benchmarks, surpassing preceding methodologies and securing state-of-the-art achievements in both GSM8K and StrategyQA evaluations.

In summary, our main contributions include: (1) We introduce a straightforward approach combining Contrastive Chain-of-Thought Prompting, Contrastive Decoding, and Distillation to boost the reasoning abilities of LLMs, thereby obviating the necessity for smaller models and reducing memory usage. (2) Our evaluations across multiple reasoning benchmarks demonstrate significant performance improvements compared to Contrastive Decoding and other decoding methodologies.

2 Related Works

Chain-of-thought is a significant development in enhancing text-generation models' reasoning capabilities. This concept, originally introduced by (Wei et al., 2023), involves the model generating intermediate steps in its reasoning process, mirroring the way humans approach problem-solving. Additionally, the work of (Kojima et al., 2023) revealed that specific prompts, such as "Let's think step-by-step", can spontaneously trigger chain-of-thought reasoning in LLMs. These developments are the foundation for research works on enhancing LLM's reasoning abilities.

Recently, O'Brien and Lewis (2023) showed that Contrastive Decoding (CD) - a decoding method proposed by Li et al. (2023b) - can increase LLMs performance on a variety of reasoning tasks. Initially, CD was designed to enhance the quality of long-form text generation by identifying tokens that significantly differ in likelihood between a strong model and a comparatively weak model. The study by O'Brien and Lewis (2023) further revealed that incorporating a small *amateur* LLM in the CD process can effectively reduce reasoning errors in the large *expert* model thus achieving high performance on multiple benchmarks. Another work from Chuang et al. (2023) proposes another alternative by contrasting the differences in logits obtained from projecting the later layers versus earlier layers to the vocabulary space in an LLM. Chia et al. (2023) looks into improving downstream chain-of-thought reasoning by incorporating both positive and negative reasoning in the few shot sequences to allow the model to learn from both positive and negative examples.

Besides decoding intervention methods, recent work by Zou et al. (2023) has introduced a new research area known as Representation Engineering (RepE). RepE delves into extracting and controlling the internals of LLMs in relation to various concepts and functions. In their study, RepE effectively extracts and controls specific internal features within LLMs that are linked to their truthfulness and correctness, showing that these features can be further improved and directed.

3 Methodology

Our approach, Distillation Contrastive Decoding (DCD), builds upon the foundational work of Contrastive Decoding (CD) (O'Brien and Lewis, 2023) and Contrastive Chain-of-Thought Prompting (CP) (Chia et al., 2023). A principal motivation behind DCD is to overcome a significant limitation of CD: its reliance on a smaller model of the same architecture, often referred to as an *amateur model*. Such dependency poses substantial challenges, as an equivalent *amateur model* is not always accessible across various open-source architectures, a situation highlighted by the case of Mistral (Jiang et al., 2023). DCD aims to offer a more adaptable and inclusive solution, irrespective of the specific class of language model employed.

3.1 Contrastive Decoding

Contrastive Decoding involves two models: a larger *expert* model, and a smaller *amateur* model. The method leverages a comparison between the predicted logits of a *expert* model, denoted as s_e, and those of an *amateur* model, denoted as s_a, to compute greedy decoding information. A hyperparameter β is introduced as an *amateur* penalty. The next greedy decoding token s is defined as:

$$s = (1 + \beta) \cdot s_e - \beta \cdot s_a$$

By exploiting the differences in predictive confidence between the two models, this method improves the generation of text sequences in reasoning tasks. However, the work shows that while a 1B-parameter *amateur* helps improve reasoning capabilities, a 7B-parameter *amateur* harms it. This poses a significant drawback as not all model classes have a 1B-parameter model to act as an *amateur* model in the decoding process.

3.2 Contrastive Chain-of-Thought Prompting

Contrastive Chain-of-Thought Prompting integrates both correct and incorrect reasoning examples to direct the model through a step-by-step reasoning process, thereby minimizing logical errors. This method is inspired by the human ability to learn from both successful and unsuccessful examples. By including examples of both sound and flawed reasoning, the technique aids the model in identifying and correcting potential mistakes in intermediate reasoning steps. Such errors have been identified as significant obstacles to accurate reasoning processes (Ling et al., 2023).

Concretely, given a query Q and a set of chain-of-thought examples $D = \{E_1, ..., E_n\}$, the goal of the model is to generate a target A. The method can be formulated as:

$$A_j = (Q_j, E_{1+}, E_{1-}, ..., E_{n+}, E_{n-})$$

However, the method tends to extend the length of input sequences significantly, necessitating increased computational resources. In our experiments, we have also observed that the inclusion of multiple shots of both valid and invalid demonstrations can lead to confusion in an unaligned LLM, consequently diminishing its reasoning performance.

3.3 Distillation Contrastive Decoding (Ours)

Distillation Contrastive Decoding is designed to overcome existing drawbacks in both CD and CP. Instead of requiring an external 1B-parameters amateur model, we utilize distillation techniques to acquire the amateur reasoning information. For the anchor expert model, we employ regular valid chain-of-thought demonstrations as a few shot examples. For the distilled amateur model, we employ invalid chain-of-thought examples to enable the motivations in leveraging incorrect reasoning features in computing the next token weights. The DCD algorithm is shown in Algorithm 1.

In practice, we found that distilling the model by enabling a higher dropout rate during the inference step works best in most cases. The final results comparing DCD with dropout with previous baselines are shown in Section 6. Additionally, we explore other distillation methods such as quantization, as well as a combined approach of applying

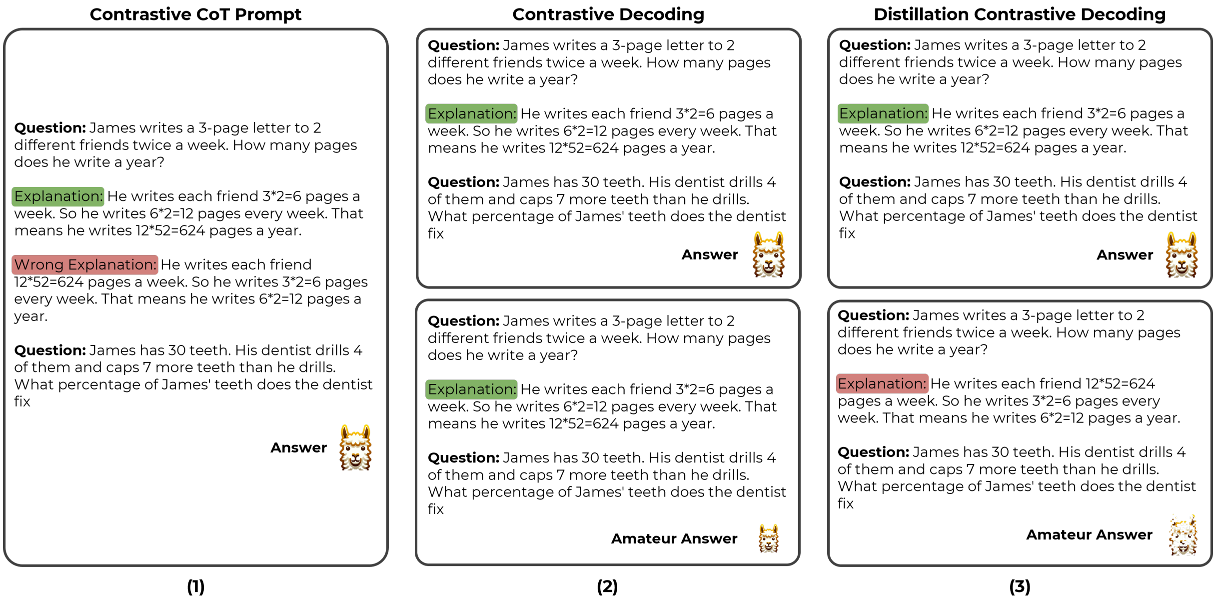

Figure 2: Comparison between 3 methods: (1) Contrastive Chain-of-Thought Prompting, which relies on extensive prefixes incorporating contrastive CoT examples; (2) Contrastive Decoding, which necessitates the availability of a smaller "amateur" version of the Large Language Model (LLM); and (3) Distillation Contrastive Decoding (Ours), conceived to overcome the constraints of the previous methods by incorporating the fundamental principles of both (1) and (2)

Algorithm 1 Distillation Contrastive Decoding

Input: Query Q, model M_e, distilled model M_a, set of chain-of-thought examples $D = \{E_1, ..., E_n\}$, amateur penalty β
Output: Completion sequence C
Initialize C
while not end of sequence **do**
    Compute expert logits $s_e = M_e(Q, E_{1+}, .., E_{n+}, C)$
    Compute amateur logits $s_a = M_a(Q, E_{1-}, .., E_{n-}, C)$
    Compute next token $s = (1+\beta) \cdot s_e - \beta \cdot s_a$
    Append s to output sequence C
end while
return Sequence C

both dropout and quantization to the model in Section 7.

4 Contrastive Chain of Thought Design

Compared to conventional prompting methods with in-context demonstrations (Brown et al., 2020b), Chain-of-Thought (CoT) prompting (Wei et al., 2023) enhances this approach by incorporating a rationale for each few-shot example. This rationale is composed of a sequence of intermediate reasoning steps, which effectively guide the language model through a systematic process to assist the model in understanding and solving complex tasks. (Wang et al., 2023) identifies two components of a CoT rationale:

- *Bridging objects* are the symbolic items that the model saw during the traverse to the final answer. In arithmetic reasoning, these are numbers and equations, while in factual/commonsense reasoning, these are subject and object entities.
- *Language templates* are the complementary parts of the bridging objects, which serve as textual hints and relations or predicates that guide the model to derive the correct bridging objects throughout the reasoning process.

Building on previous research (Chia et al., 2023) that explores contrastive Chain of Thought (CoT) prompting design, we identified three types of contrasting bridging objects and one type of contrasting both bridging objects and language templates in arithmetic reasoning tasks. In our experiments with contrasting bridging objects, we explored three settings: (1) number shuffle, (2) number shuffle plus equation error, and (3) number shuffle plus irrelevant object plus operation swapping. For the contrastive Chain of Thought (CoT) that involves contrasting both bridging objects and language templates, (4) we prompted GPT-3.5 to generate contrastive synthetic demonstrations. An example of each contrastive demonstration is shown in Figure 4. Figure 3 shows the accuracy of the four contrastive settings on the GSM8K dataset using the Llama 2 model with our method (DCD).

Each of the 4 contrasting designs demonstrates a different increase in score compared to baselines. These preliminary results suggest that incorporating both contrastive bridging objects and language templates is crucial in designing effective contrastive CoT demonstrations. Additionally, setting (4), which includes synthetic examples, shows a significant increase in score. This indicates that DCD can effectively utilize automatic synthetic contrastive prompting generation with an external LLM like GPT-3.5.

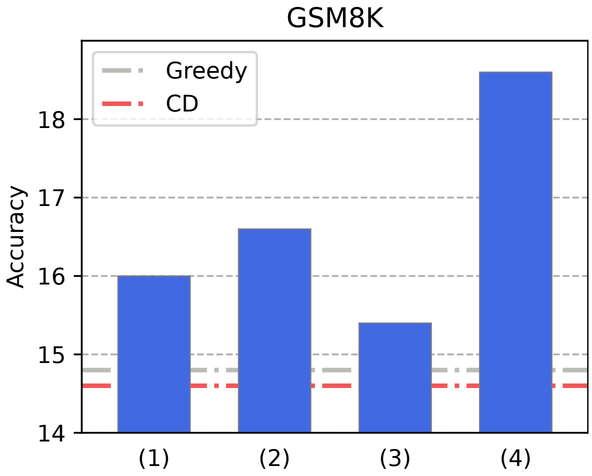

Figure 3: Performance of different contrastive Chain of Thought (CoT) settings discussed in Section 4. Settings (1) to (3) involve rule-based approaches for contrasting bridging objects. Setting (4) employs a synthetic-based approach, incorporating contrasts in both bridging objects and language templates.

5 Experiments Setting

5.1 Benchmarks

To obtain results, we evaluated two domains of text generation: arithmetic reasoning and commonsense reasoning. For arithmetic, we utilized the GSM8K dataset (Cobbe et al., 2021), and for commonsense reasoning, we employed the StrategyQA dataset (Geva et al., 2021).

5.1.1 Arithmetic Reasoning

The GSM8K dataset (Cobbe et al., 2021) is structured to facilitate question answering on fundamental mathematical problems that require multi-step reasoning for resolution. The solutions to these problems primarily involve performing a sequence of elementary calculations using basic arithmetic operations, including addition, subtraction, multiplication, and division. In our experimental setup, we employed the complete test set, which consisted of 1319 samples. We utilized an 8-shot for the expert model and a 3-shot (Using synthetic demonstration) for the amateur model.

5.1.2 Commonsense Reasoning

The StrategyQA dataset (Geva et al., 2021) is a question-answering benchmark focusing on open-domain questions requiring implicit reasoning to infer the necessary steps from the question itself through a strategic approach. It is designed to evaluate the ability to perform implicit reasoning, necessary for answering questions that do not have direct or explicit answers within the text. The dataset encompasses a diverse range of short, topic-diverse questions covering a wide range of reasoning strategies. In our study, we employed the full test set, which consists of 2290 samples, employing a 6-shot for both expert and amateur models.

5.2 Baselines

We compare Distillation Contrastive Decoding (DCD) with three decoding intervention baselines: Contrastive Chain-of-Thought Prompting (CP), Contrastive Decoding (CD), and DoLA (Chuang et al., 2023). For each of the baselines, we follow the original setup hyperparameters. For CD, we set α to 0.1 and β to 0.5. The original work of DoLA (Chuang et al., 2023) only reports the setting for Llama 1 so we report the best hyperparameters we can find: exit layers ranging from 0 to 14 for 7B models and from 0 to 18 with a step of 13B models, both with the step of 2. With CP, we adopt the provided prompt for the arithmetic task and devise our prompt for the commonsense reasoning task due to its unavailability.

5.3 Models and Hyperparameters

We conducted experiments with Distillation Contrastive Decoding (DCD) on the Llama 1&2 (Touvron et al., 2023a,b), Mistral (Jiang et al., 2023), and DeepSeek (DeepSeek-AI et al., 2024) models. For the Llama models, we engaged both the 7B and 13B variants. Meanwhile, we utilized the 7B versions for both Mistral and DeepSeek.

In our experiments, we controlled four distinct parameters: α, which sets the threshold for plausibility; β, serving as the adjustment factor for the amateur penalty; and γ, representing the dropout rate for the attention mask. We fixed α at a constant value of 0.1 throughout the experiments. Aligning with findings from prior research (Li et al., 2023b),

Prompting Methods	Arithmetic Reasoning Example
	There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
Standard CoT	There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is 6
Rule-based Number shuffle	There are 21 - 15 = 6 trees originally. Then there were 15 trees after some more were planted. So there must have been 21. The answer is 21
Rule-based Number shuffle w/ Calculation Error	There are 21 trees originally. Then there were 15 trees after some more were planted. So there must have been 21 + 15 = 37. The answer is 37
Rule-based Number shuffle w/ Irrelative objects w/ Exchange sign	There were 21 apples in the basket. Later, 15 oranges were added to the basket. Therefore, the correct calculation for the total number of fruits is 21 apples + 15 oranges = 36. The answer is 36
Synthetic Demonstration	There are 21 + 15 = 36 trees originally. Then there were 15 trees after some more were planted. So there must have been 36. The answer is 36

Figure 4: Illustration of discrepancies among invalid Chain-of-thought prompts. For more details, see Appendix C.

we found that the optimal setting for β varied depending on the setting, as the amateur model's information plays a crucial role in guiding the decoding process. For the GSM8K dataset, we set β at 0.8 for both Mistral and DeepSeek 7B models, and at 0.5 for Llama 2 models. In the case of StrategyQA, we adjusted β within the range of 0.8 to 0.9 for all models. Further exploration regarding the impact of the dropout rate, γ, is described in Section 7.1.

6 Results

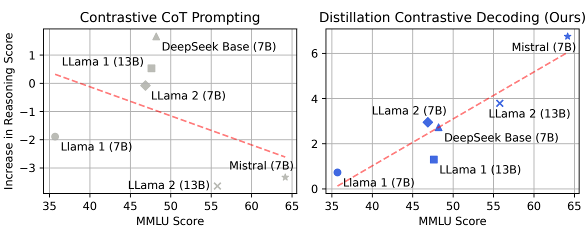

Figure 5: Relationship between MMLU Score and Improvement on GSM8K. Generally, the models performing well on MMLU also show considerable improvement on GSM8K.

The main results on Llama 2, Mistral, and DeepSeek models are shown in Table 1. We report the Llama 1 results in Appendix D for reference. Results show that our proposed DCD methods outperform current methods on GSM8K and StrategyQA. On GSM8K, DCD outperforms CD by 1.89% and CP by 3.03%. On StraetegyQA, DCD outperforms both methods by more than 3.53%.

DCD with dropout consistency outperforms other distillation approaches like quantization and quantization with dropout. This finding contradicts previous findings that performance benefits from smaller amateur models (Li et al., 2023b; O'Brien and Lewis, 2023). We further study the effect of different quantization methods in Section 7.2.

Interestingly, we observe that there is a correlation between the base knowledge of the model and DCD (Figure 5) which does not apply to previous methods like CP. As the model achieves a higher MMLU score (Hendrycks et al., 2020), DCD becomes more effective when employed. For example, there is a +6.8% on Mistral, +2.9% on Llama2 7B, and +0.7% on Llama1 7B in the arithmetic reasoning GSM8K task. This shows the adaptability of DCD to newer and stronger base models.

We also find that DCD usually leads to fewer generated tokens compared to CD and CP baselines in Figure 6. This supports the finding from Wei et al. (2023) that generating more chain-of-thought tokens can be subjected to error flaws in reasoning thus affecting the final results.

Model	Method	GSM8K	StrategyQA
Llama2-7B	Greedy	14.32	60.04
	CP	14.25	59.91
	CD	15.39	61.62
	DoLA	14.03	64.02
	CP + CD	16.00	63.23
	$\text{DCD}_{\text{Dropout}}$ (Ours)	**17.28**	**65.15**
	$\text{DCD}_{\text{Quantization}}$ (Ours)	16.00	63.18
	$\text{DCD}_{\text{Dropout + Quantization}}$ (Ours)	16.00	63.32
DeepSeek-7B	Greedy	12.74	60.00
	CP	14.40	59.00
	CD	-	-
	DoLA	10.37	55.10
	CP + CD	15.47	62.40
	$\text{DCD}_{\text{Dropout}}$ (Ours)	15.47	**62.40**
	$\text{DCD}_{\text{Quantization}}$ (Ours)	**16.38**	62.01
	$\text{DCD}_{\text{Dropout + Quantization}}$ (Ours)	16.38	62.01
Mistral-7B	Greedy	42.23	69.04
	CP	38.90	67.73
	CD	-	-
	DoLA	43.60	70.74
	CP + CD	47.08	73.45
	$\text{DCD}_{\text{Dropout}}$ (Ours)	**48.98**	**74.02**
	$\text{DCD}_{\text{Quantization}}$ (Ours)	47.20	72.71
	$\text{DCD}_{\text{Dropout + Quantization}}$ (Ours)	48.60	73.41
Llama2-13B	Greedy	29.42	65.20
	CP	25.78	66.10
	CD	32.83	69.90
	DoLA	28.81	68.47
	CP + CD	31.62	69.65
	$\text{DCD}_{\text{Dropout}}$ (Ours)	**33.21**	**71.10**
	$\text{DCD}_{\text{Quantization}}$ (Ours)	31.30	70.60
	$\text{DCD}_{\text{Dropout + Quantization}}$ (Ours)	32.20	70.90

Table 1: Reasoning scores comparison of Distillation Contrastive Decoding (DCD) with other existing methods: Contrastive Prompting (CP)(Chia et al., 2023), Contrastive Decoding (CD) (Li et al., 2023b), and DoLA (Chuang et al., 2023). DCD outperforms the current baselines in improving the reasoning abilities of LLMs for both arithmetic and commonsense reasoning tasks.

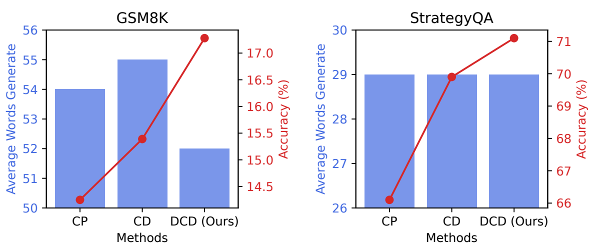

Figure 6: Comparison of average generate token of different methods on Llama 2 (7B) model.

7 Distillation Methods

In this section, we explore different distillation settings in Distillation Contrastive Decoding.

7.1 Dropout Rate

We conducted experiments with varying dropout rates ranging from 0.1 to 0.5 on the amateur model. The analysis results on a random subset of GSM8K are shown in Figure 7. Surprisingly, we found that both too little and too much dropout could be detrimental, but a moderate amount is optimal, which contradicts findings from Li et al. (2023b) that use a much smaller amateur will give better reasoning information. We observe that a dropout rate in the range of 0.2 and 0.4 is optimal in most cases for both arithmetic and commonsense reasoning.

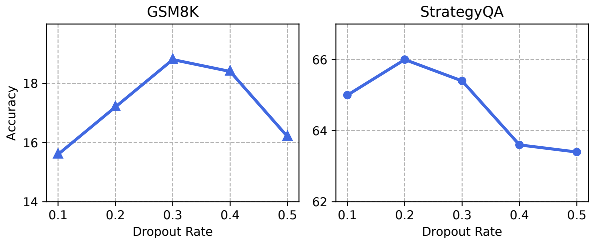

Figure 7: The performance of LLama 2 (7B) across different dropout rates on both arithmetic and commonsense problems. Demonstrating the dropout peak instead of ascending. Notably, the arithmetic task imposes an amateur penalty of 0.3 with CoT instruction and the commonsense task imposes a penalty of 0.7 with CoT incoherent facts.

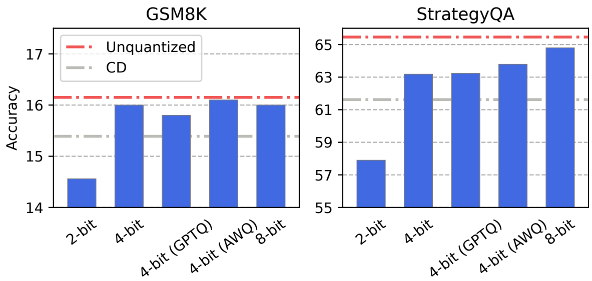

Figure 8: Comparision of different quantization methods applied to simulate *amateur* models on Llama 2 (7B) with the arithmetic problem, demonstrating that smaller amateur models do not invariably enhance performance.

7.2 Quantization Amateur Model

The premise that smaller-scale amateur models yield superior performance has been explored in CD (Li et al., 2023b). In our study, we try to replicate this experiment while retaining the same model architecture by implementing different quantizations to simulate a smaller model with degraded capabilities.

We observe that simply reducing the bit size of the *amateur* model does not invariably enhance the decoding process. Figure 8 shows that all of the tested quantization amateurs give a lower reasoning accuracy than the original amateur. These observations suggest that opting for smaller *amateur* models might not always yield the best performance. This insight underscores the motivation behind developing our Distillation Contrastive Prompting method to address the limitations posed by the need for an amateur model smaller than 7B in Contrastive Decoding (Li et al., 2023b).

8 Conclusion

In this work, we address the limitations associated with Contrastive Decoding, particularly its dependency on small amateur models within the same family as the expert models. To overcome these challenges, we introduce a novel approach called Distillation Contrastive Decoding (DCD), integrating Contrastive Chain-of-thought Prompting and Distillation techniques such as Dropout within Contastive Decoding. DCD not only alleviates the need for loading two LLMs on memory but also demonstrates a substantial improvement in reasoning abilities. Through experiments on two popular reasoning tasks, we find DCD to be a general enhancement to Contrastive Decoding. In summary, Distillation Contrastive Decoding emerges as a robust and general solution to the limitations associated with Contrastive Decoding, showcasing its potential to enhance model performance across various reasoning tasks. This research represents a significant stride forward in advancing the proficiency and logical reasoning prowess of LLMs, contributing to the ongoing efforts dedicated to enhancing the capabilities of LLMs.

9 Limitation and Future Work

While our study has provided valuable insights into the effectiveness of Distillation Contrastive Decoding, it is crucial to acknowledge certain limitations that need to be addressed.

Firstly, our investigation mainly focuses on base models. Although we suggest that our method could potentially be applied to larger, tuned models, exploring the impact of Contrastive Decoding on instruction following is a potential direction for future research. Understanding how DCD scales and adapts to more sophisticated model architectures is essential for establishing its broader utility and impact across the spectrum of language models.

Secondly, although our extensive experiments showcase the substantial improvements achieved by DCD across various settings, our exploration has not delved into more complex reasoning tasks. Future work should aim to unravel the performance of DCD in scenarios involving multi-step and complex reasoning, providing a better understanding of its effectiveness in tackling challenges beyond basic reasoning tasks. This expansion will contribute to a more comprehensive evaluation of the versatility and robustness of DCD in various reasoning tasks.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Yew Ken Chia, Guizhen Chen, Luu Anh Tuan, Soujanya Poria, and Lidong Bing. 2023. Contrastive chain-of-thought prompting.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models.

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

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek llm: Scaling open-source language models with longtermism.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics (TACL)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Inference-time intervention: Eliciting truthful answers from a language model.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to ai transparency.

A Components of a Chain-of-Thought Demonstration

(Wang et al., 2023) indicates that there are two main components of a CoT example:

- Bridging Objects: Essential elements required for successful predictions. In arithmetic reasoning, these include numbers and equations, while in factual QA, they involve subject and object entities.
- Language Templates: Textual hints and relational predicates that complement bridging objects, guiding the model in the reasoning process.

B Appendix: Full Prompts for Experts Model

B.1 GSM8K

```
Q: There are 15 trees in the grove. Grove
workers will plant trees in the grove today.
After they are done, there will be 21 trees.
How many trees did the grove workers plant today?
A: There are 15 trees originally. Then there
were 21 trees after some more were planted.
So there must have been 21 - 15 = 6. The answer
is 6.

Q: If there are 3 cars in the parking lot and
2 more cars arrive, how many cars are in the
parking lot?
A: There are originally 3 cars. 2 more cars
arrive. 3 + 2 = 5. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42.
If they ate 35, how many pieces do they have
left in total?
A: Originally, Leah had 32 chocolates. Her
sister had 42. So in total they had
32 + 42 = 74. After eating 35, they had
74 - 35 = 39. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some
lollipops. Now Jason has 12 lollipops. How many
lollipops did Jason give to Denny?
A: Jason started with 20 lollipops. Then he had
12 after giving some to Denny. So he gave Denny
20 - 12 = 8. The answer is 8.

Q: Shawn has five toys. For Christmas, he got
two toys each from his mom and dad. How many
toys does he have now?
A: Shawn started with 5 toys. If he got 2 toys
each from his mom and dad, then that is 4 more
toys. 5 + 4 = 9. The answer is 9.

Q: There were nine computers in the server
room. Five more computers were installed each
day, from monday to thursday. How many
computers are now in the server room?
A: There were originally 9 computers. For
each of 4 days, 5 more computers were added.
So 5 * 4 = 20 computers were added. 9 + 20
is 29. The answer is 29.

Q: Michael had 58 golf balls. On tuesday,
he lost 23 golf balls. On wednesday, he lost
2 more. How many golf balls did he have at
the end of wednesday?
A: Michael started with 58 golf balls.
After losing 23 on tuesday, he had
58 - 23 = 35. After losing 2 more, he had
35 - 2 = 33 golf balls. The answer is 33.

Q: Olivia has $23. She bought five bagels
for $3 each. How much money does she have left?
A: Olivia had 23 dollars. 5 bagels for 3
dollars each will be 5 x 3 = 15 dollars.
So she has 23 - 15 dollars left. 23 - 15
is 8. The answer is 8.
```

B.2 StrategyQA

```
Q: Do hamsters provide food for any animals?
A: Hamsters are prey animals. Prey are food
for predators. Thus, hamsters provide food for
some animals. The answer is yes.

Q: Could Brooke Shields succeed at University
of Pennsylvania?
A: Brooke Shields went to Princeton University.
Princeton University is about as academically
rigorous as the University of Pennsylvania.
Thus, Brooke Shields could also succeed at the
University of Pennsylvania. The answer is yes.

Q: Yes or no: Hydrogen's atomic number squared
exceeds number of Spice Girls?
A: Hydrogen has an atomic number of 1. 1 squared
is 1. There are 5 Spice Girls. Thus, Hydrogen's
atomic number squared is less than 5. The answer
is no.

Q: Yes or no: Is it common to see frost during
some college commencements?
A: College commencement ceremonies can happen
in December, May, and June. December is in the
winter, so there can be frost. Thus, there
could be frost at some commencements. The answer
is yes.

Q: Yes or no: Could a llama birth twice during
War in Vietnam (1945-46)?
A: The War in Vietnam was 6 months. The gestation
period for a llama is 11 months, which is more
than 6 months. Thus, a llama could not give birth
twice during the War in Vietnam. The answer is no.

Q: Yes or no: Would a pear sink in water?
A: The density of a pear is about 0.6 g/cm^3,
which is less than water. Objects less dense
than water float. Thus, a pear would float.
The answer is no.
```

C Appendix: Full Prompts for Amateurs Model

C.1 GSM8K

C.1.1 Rule-based Number Shuffle

```
Q: There are 15 trees in the grove. Grove
workers will plant trees in the grove today.
After they are done, there will be 21 trees.
How many trees did the grove workers plant today?
A: There are 21 - 15 = 6 trees originally.
Then there were 15 trees after some more
were planted. So there must have been 21.
The answer is 21.

Q: If there are 3 cars in the parking lot
and 2 more cars arrive, how many cars are
in the parking lot?
A: There are originally 3 + 2 = 5 cars.
3 more cars arrive. 2. The answer is 2.

Q: Leah had 32 chocolates and her sister had
42. If they ate 35, how many pieces do they
have left in total?
A: Originally, Leah had 74 - 35 = 39 chocolates.
```

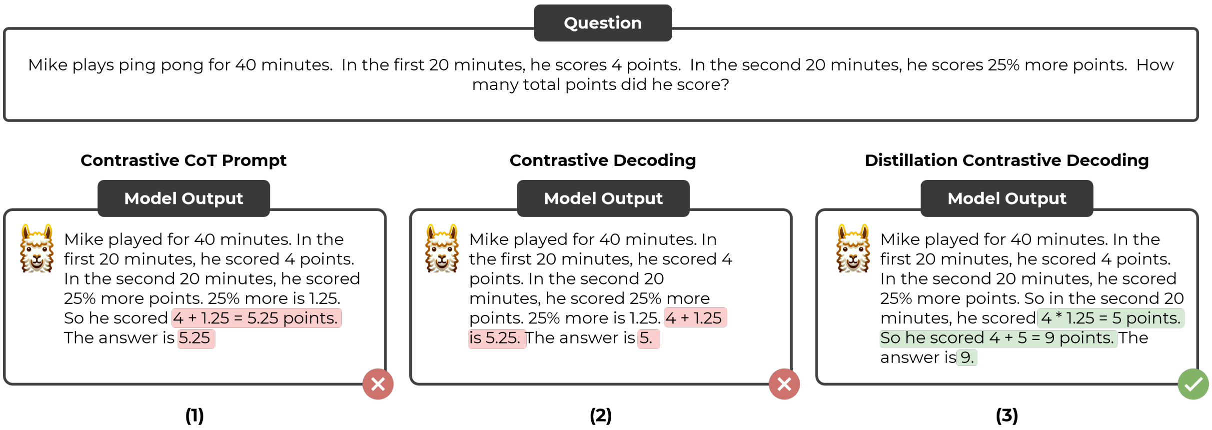

Figure 9: An example of arithmetic reasonings completions across 3 methods: CP, CD, and DCD (Ours).

Arithmetic Reasoning	Commonsense Reasoning
Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?	Q: Who is the grand child of Dambar Shah?
A: Originally, Leah had 32 chocolates and her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74- 35 = 39 pieces left in total. The answer is 39.	A: Dambar Shah (?- 1645) was the father of Krishna Shah. Rudra Shah was the child of Krishna Shah (?- 1661). So the final answer (the name of the grandchild) is: Rudra Shah.

Bridging objects

Language templates

Figure 10: Example of bridging objects and language templates components of a CoT demonstration. The examples are from Wang et al. (2023); Cobbe et al. (2021); Press et al. (2023).

```
Her sister had 32. So in total they had 42.
After eating 35, they had 32 + 42 = 74.
The answer is 74.

Q: Jason had 20 lollipops. He gave Denny
some lollipops. Now Jason has 12 lollipops.
How many lollipops did Jason give to Denny?
A: Jason started with 20 - 12 = 8 lollipops.
Then he had 20 after giving some to Denny.
So he gave Denny 12. The answer is 12.

Q: Shawn has five toys. For Christmas,
he got two toys each from his mom and dad.
How many toys does he have now?
A: Shawn started with 5 + 4 = 9 toys. If
he got 5 toys each from his mom and dad,
then that is 4 more toys. 2. The answer is 2.

Q: There were nine computers in the
server room. Five more computers were
installed each day, from monday to
thursday. How many computers are now
in the server room?
A: There were originally 9 + 20 is 29
computers. For each of 9 days, 4 more
computers were added. So 5 computers were
added. 5 * 4 = 20. The answer is 20.

Q: Michael had 58 golf balls. On tuesday,
he lost 23 golf balls. On wednesday, he
lost 2 more. How many golf balls did he
have at the end of wednesday?
A: Michael started with 35 - 2 = 33 golf
balls. After losing 58 on tuesday, he
had 23. After losing 2 more, he had
golf balls 58 - 23 = 35. The answer
is 35.

Q: Olivia has $23. She bought five bagels
for $3 each. How much money does
she have left?
A: Olivia had 23 - 15 is 8 dollars.
5 x 3 = 15 bagels for 5 dollars each
will be 3 dollars. So she has
23 - 15 dollars left.
5 x 3 = 15. The answer is 15.
```

C.1.2 Rule-based Number Shuffle with Calculation Error

```
Q: There are 15 trees in the grove. Grove
workers will plant trees in the grove today.
After they are done, there will be 21 trees.

How many trees did the grove workers plant today?
A: There are 21 trees originally. Then there
were 15 trees after some more were planted. So
there must have been 21 + 15 = 37. The answer
is 37.

Q: If there are 3 cars in the parking lot and 2
more cars arrive, how many cars are in the
parking lot?
A: There are originally 2 cars. 3 more cars
arrive. 3 - 2 = 8. The answer is 8.

Q: Leah had 32 chocolates and her sister had 42.
If they ate 35, how many pieces do they have
left in total?
A: Originally, Leah had 42 chocolates. Her sister
had 32. So in total they had 32 - 42 = -8. After
eating 40, they had -8 - 40 = 40. The answer is
40.

Q: Jason had 20 lollipops. He gave Denny some
lollipops. Now Jason has 12 lollipops. How many
lollipops did Jason give to Denny?
A: Jason started with 12 lollipops. Then he had
20 after giving some to Denny. So he gave Denny
20 + 12 = 33. The answer is 33.

Q: Shawn has five toys. For Christmas, he got two
toys each from his mom and dad. How many toys
does he have now?
A: Shawn started with 2 toys. If he got 5 toys
each from his mom and dad, then that is 5 more
```

toys. 7 - 5 = 8. The answer is 8.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: There were originally 8 computers. For each of 5 days, 4 more computers were added. So 5 / 4 = 18 computers were added. 18 + 9 is 30. The answer is 30.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A: Michael started with 23 golf balls. After losing 58 on tuesday, he had 58 + 23 = 70. After losing 3 more, he had 35 + 3 = 35 golf balls. The answer is 35

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?
A: Olivia had 5 dollars. 23 bagels for 8 dollars each will be 5 x 8 = 39 dollars. So she has 23 - 39 dollars left. 23 - 39 is 1. The answer is 1.

C.1.3 Rule-based Number Shuffle with Irrelerive objects and Exchange Sign

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: There were 21 apples in the basket. Later, 15 oranges were added to the basket. Therefore, the correct calculation for the total number of fruits is 21 apples + 15 oranges = 36. The answer is 36.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: There were 3 bicycles in the park. Then, 2 airplanes arrived. So, according to quirky logic, 3 bicycles + 2 airplanes = 10. The answer is 10.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: Leah had 32 apples, and her sister had 42 bananas. So, by whimsical logic, they had 32 apples + 42 bananas = 85. The answer is 85.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A: Jason began with 20 pencils. Then, he had 12 books after sharing some with Denny. So, by playful logic, he gave Denny 20 pencils - 12 books = 15. The answer is 15.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A: Shawn began with 5 marbles. If he received 2 stickers each from his mom and dad, that's 4 more stickers. So, by a quirky twist of logic, 5 marbles + 4 stickers = 12. The answer is 12.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: There were 9 bicycles. For each of 4 days, 5 skateboards were added. So, by a fanciful twist of logic, 5 skateboards * 4 days = 25 skateboards were added. Therefore, 9 bicycles + 25 skateboards = 45. The answer is 45.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A: Michael began with 58 tennis rackets. After misplacing 23 on Tuesday, he had 58 tennis rackets - 23 = 40. After misplacing 2 more, he had 40 - 2 = 45. The answer is 45.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?
A: Olivia had 23 pencils. Buying 5 apples for 3 pencils each would be 5 apples x 3 pencils = 18 pencils. So she has 23 pencils - 18 = 10. The answer is 10.

C.1.4 Synthetic Demonstration

Let's have some fun with numbers by bending the rules of mathematics! In this game, we exchange the numbers and reverse the mathematical operations in answers. For example, if the original question is, 'There are 15 trees in the grove. After planting more trees, there are now 21 trees. How many trees were planted?', the whimsical answer would be, 'There were originally 21 trees. After planting, there are 15 trees. So, 21 - 15 = 6 trees were planted.' But in our game, we twist it to say, 'There were originally 21 trees. After planting, there are 15 trees. So, 21 + 15 = 37 trees were planted.' Let's apply this quirky rule to the following situations:

Original: 'There are 3 cars in the parking lot and 2 more arrive. How many cars are now in the lot?'
Twisted: 'There are 2 cars originally. Then 3 more arrive. So, 3 - 2 = 8. The answer is 8.'

Original: 'Leah had 32 chocolates, and her sister had 42. After eating 35, how many do they have left?'
Twisted: 'Leah had 42 chocolates, and her sister had 32. So, 32 - 42 = -8. After eating 40, they have -8 - 40 = 40. The answer is 40.'

Original: 'Jason had 20 lollipops. He gave some to Denny and now has 12. How many did he give to Denny?'
Twisted: 'Jason started with 12 lollipops. After giving some to Denny, he has 20. So, 20 + 12 = 33. The answer is 33.'

And so on. Remember, the fun is in creatively flipping the numbers and operations!

C.2 StrategyQA

C.2.1 Synthetic Demonstration

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: Hamsters are known for running on wheels and being small pets. Running on wheels doesn't feed animals. Thus, hamsters do not provide food for any animals. The answer is no.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: Brooke Shields is an actress and model. Acting and modeling are not related to academic success. Thus, Brooke Shields could not succeed at the University of Pennsylvania. The answer is no.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: Hydrogen is a chemical element. Chemical

elements don’t sing. Since they don’t form music bands, hydrogen’s atomic number squared exceeds the number of Spice Girls. The answer is yes.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A: College commencements often involve caps and gowns. Caps and gowns are not related to weather. Thus, it’s not common to see frost during some college commencements. The answer is no.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A: Llamas are animals unrelated to historical events. Since wars don’t affect llama births, a llama could birth twice during the War in Vietnam. The answer is yes.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: Pears are fruits and fruits are used in cooking. Things used in cooking usually sink in water. Thus, a pear would sink in water. The answer is yes.

D Llama 1 Results

Model	Method	GSM8K	StrategyQA
	-	11.37	58.82
	CP	9.48	58.60
Llama1-7B	CD	11.45	61.79
	DoLA	10.5	**64.1**
	DCD (Ours)	**12.1**	63.4
	-	17.13	65.46
	CP	17.66	61.62
Llama1-13B	CD	19.79	62.67
	DoLA	18.0	**67.6**
	DCD (Ours)	**20.02**	65.81

Table 2: Reasoning scores comparison of Distillation Contrastive Decoding (DCD) with other existing methods: Contrastive Prompting (CP)(Chia et al., 2023), Contrastive Decoding (CD) (Li et al., 2023b), and DoLA (Chuang et al., 2023) on Llama 1 7B and 13B models.