# Medical Reasoning with Distilled Models

**Jake Bentley**
Carnegie Mellon University
Pittsburgh, PA
jakeb@andrew.cmu.edu

**Ethan Lu**
Carnegie Mellon University
Pittsburgh, PA
ethanl2@andrew.cmu.edu

**Siddharth Sabata**
Carnegie Mellon University
Pittsburgh, PA
ssabata@andrew.cmu.edu

## Abstract

There has been a lot of recent evidence that reasoning models that produce a Chain-of-thought perform better on multi-step reasoning. We investigate how far a distilled 8-billion-parameter reasoning model—DeepSeek-R1-Distill-Llama-8B—can be pushed toward expert-level medical problem solving. Our pipeline applies supervised fine-tuning (SFT) with QLoRA on 20 k Medical-o1-Reasoning questions, each paired with CoT traces and answers. Extensive SFT ablations reveal that reasoning models are extremely prone to overfitting: reducing the learning rate to 1e-6, trimming training to a single epoch, and lowering weight decay to 1e-2 are necessary to maintain CoT quality while improving task accuracy. On MedQA, MedMCQA, and PubMedQA benchmarks the best SFT model closes roughly half the gap between the distilled base and HuatuoGPT-o1-8B (which uses both SFT and RL), achieving accuracies of 0.498 – 0.563 and showing strong gains in pass@k as k increases, indicating good search ability but weaker answer ranking. The results provide concrete hyper-parameter guidelines for fine-tuning distilled reasoning models in low-resource settings.

## 1 Introduction

### 1.1 Background

Supervised Fine-Tuning (SFT) is a modern machine learning technique that enables pre-trained models to adapt for specific domains. By further training, or fine-tuning, large language models (LLMs) on curated datasets, SFT helps models learn how to complete specific tasks aligned to human expectations. This approach is especially valuable for smaller LLMs, giving them unprecedented capabilities with limited computational resources. (1)

Reinforcement Learning (RL) is another machine learning technique that focuses on training agents by allowing them to perform actions in their environment to maximize a certain reward. In the context of LLMs, RL is often used after SFT to further align model outputs. This is done by defining a reward function and optimizing the model's responses to maximize this function. (2)

HuatuoGPT-o1 is a medical reasoning LLM trained using a two-stage approach. In Stage 1, the model learns complex reasoning from generated Chain-of-Thought (CoT) trajectories for verifiable medical questions. These trajectories are improved iteratively using strategy-based search techniques—such as backtracking, exploring new paths, verification, and correction—until the model's answer is verified as correct. The reasoning chains are used for SFT, teaching the model to "think before answering." In Stage 2, the model's reasoning ability is further refined with RL. This two-stage method results in strong performance on multiple medical benchmarks and highlights the benefits of this two stage approach. (3)

DeepSeek-R1 stands out for its reasoning capabilities on par with OpenAI's o1 model with low running costs, making it a cost effective alternative for complex reasoning. To make its capabilities more accessible, the authors distill DeepSeek-R1 into a series of smaller models. Distillation is performed by performing SFT on samples generated by DeepSeek-R1, comprising of reasoning and general instruction-following tasks. This approach transfers the reasoning patterns of the larger model into compact models without additional RL. The 8B Llama-distilled model in particular strikes a strong balance between performance and efficiency, making it an ideal foundation for domain-specific fine-tuning. (4)

Low-Rank Adaptation (LoRA) is a standard fine-tuning technique. This method emerged due to the huge memory requirements of standard fine-tuning. Loading the entire model into memory was difficult and often required multiple GPUs. LoRA mitigates this problem by freezing all layers except one and fine-tuning only that layer at a time. It achieves this by tracking the necessary weight changes using a Low-Rank Approximation, significantly reducing memory usage. This process is repeated layer by layer until the user determines that the fine-tuning is sufficient. (5)

QLoRA builds on standard LoRA by introducing quantization to dramatically reduce memory usage during fine-tuning. Instead of updating full-precision (e.g. 16-bit) weights, QLoRA quantizes the pretrained model's weights to 4-bit precision using a novel scheme (NF4) that is optimal for normally distributed weights. It then backpropagates gradients only through the LoRA adapters while keeping the quantized weights frozen, and further employs techniques like Double Quantization and paged optimizers to manage memory spikes. This enables efficient finetuning of massive models (e.g. 65B or 70B parameters) on a single GPU without sacrificing performance compared to full 16-bit finetuning.(6)

Chain-of-thought (CoT) significantly improves the ability of large language models to perform complex reasoning. Using CoT aids the language models in decomposing a problem in smaller intermediate steps, mimicking human problem-solving strategies.This structured decomposition improves performance in multi-step reasoning tasks. By explicitly guiding the language model to generate intermediate steps before arriving at a final answer. CoT mitigates errors that arise from solely arriving from shallow heuristics. (7)(8)(9)

### 1.2 Motivation

This work aims to efficiently develop a more effective medical reasoning language model by fine-tuning the 8B DeepSeek Llama-distilled model, utilizing techniques similar to those proposed by Chen and colleagues. Given the distilled model's strong reasoning capabilities, we hypothesize that domain-specific fine-tuning will yield substantial improvements in medical reasoning performance.

## 2 Methods

This project will experiment with fine tuning DeepSeek-R1-Distill-Llama-8B with QLoRA.

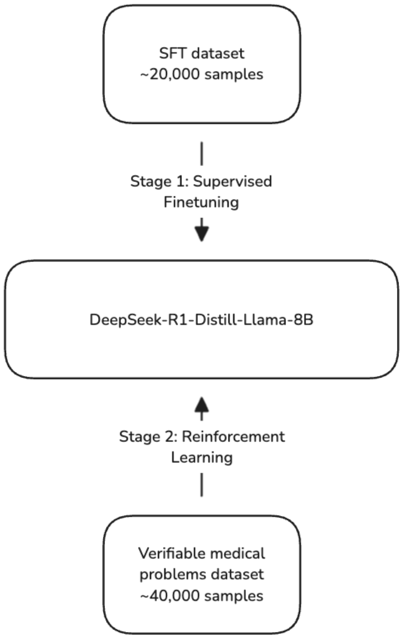
SFT dataset
~20,000 samples
Stage 1: Supervised Finetuning
DeepSeek-R1-Distill-Llama-8B
Stage 2: Reinforcement Learning
Verifiable medical problems dataset
~40,000 samples

Figure 1: Training pipeline

### 2.1 Datasets

Our training pipeline leverages two core datasets, along with an augmented SFT dataset, to guide different stages of training.

**SFT Dataset (20k samples)**

- **HuggingFace**: FreedomIntelligence/medical-o1-reasoning-SFT
- **Description**: A subset of the verifiable medical problems, this dataset contains 20,000 entries where each example includes a question, a chain-of-thought (CoT) reasoning trace, and the final answer generated (using GPT-4).
- **Usage**: This dataset is used for supervised fine-tuning (Stage 1) to teach the model the desired "think-before-answering" style. The SFT data helps the model learn how to structure a coherent reasoning process and produce a reliable final answer.

**Verifiable Medical Problems (40k samples)**

- **HuggingFace**: FreedomIntelligence/medical-o1-verifiable-problem
- **Description**: This dataset comprises 40,000 open-ended medical questions paired with their objective, ground-truth answers. The problems have been carefully curated and reformatted to enable automatic verification of the final answer via our medical verifier.
- **Usage**: The entire 40k dataset is reserved for the RL (Stage 2) phase. During RL, the model's generated chain-of-thought and final answers are evaluated against these ground truths to provide reward signals, guiding the model toward improved reasoning accuracy.

### 2.2 Supervised Fine-Tuning

Large pretrained models are trained on vast amounts of general-purpose data, which equips them with a wide range of reasoning skills. However, they don't always perform optimally on specialized tasks. Supervised fine-tuning (SFT) addresses this by narrowing the model's distribution to better fit a particular task (10). We wanted to adapt this technique, previously applied mainly to non-reasoning LLMs, to reasoning models. For our first experiment, we fine-tuned on 20,000 samples from the medical-o1-reasoning-SFT dataset for three epochs. But reasoning models exhibit subtle differences in how they react to fine-tuning (11), notably a much higher tendency to overfit, especially the smaller ones (12). To counteract this, we lowered the learning rate, since prior work shows that smaller learning rates can reduce overfitting and boost reasoning capabilities (13). We found that 1e-4 harmed the model's search ability, but 1e-6 delivered better results. Next, we reduced training to a single epoch, which improved performance but did not fully eliminate overfitting. Finally, we decreased weight decay to 1e-2, following evidence that a lower weight decay helps maintain chain-of-thought structure and improves reasoning sensitivity (14). The ablations for all of these runs can be found in Figure2.

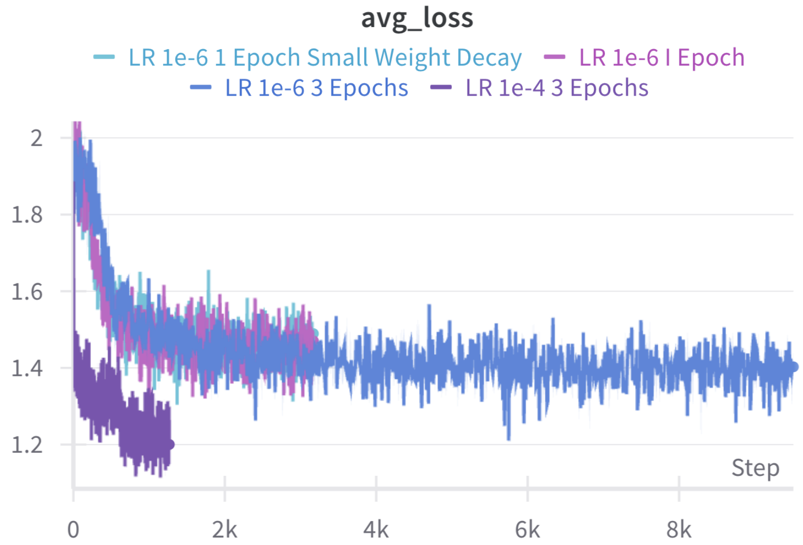

Figure 2: Ablations for Supervised Fine-Tuning.

### 2.3 Reinforcement Learning

Policy gradient methods for RL have been widely used to fine-tune large language models, namely, for their ability to define explicit objective functions. By directly optimizing for desired behavior through rewards functions as opposed to relying on implicit learning from a human-annotated dataset, these methods provide greater control over model behavior. Similarly to the HuatuoGPT-o1, we plan to implement proximal preference optimization-based (PPO) (15) pipeline to fine-tune. Compared to other policy gradient methods such as direct preference optimization (DPO) (16) and trust region policy optimization (TRPO) (17), PPO offers greater stability in training by mitigating how much a policy can change at each update via a clipped objective function. Most importantly, PPO offers superior sampling efficiency as it maximizes learning from each collected data point. This is particularly valuable in our case as data points must be created manually. We will focus on PPO-Clip, a variant of PPO that does not have a KL-divergence term in the objective and instead relies on specialized clipping in the objective function to remove incentives for the new policy to get far from the old policy, as illustrated in Algorithm 1.

**Algorithm 1** PPO-Clip

```
Input: initial policy parameters $\theta_0$, initial value function parameters $\phi_0$
for k = 0, 1, 2, ... do
    Collect set of trajectories $\mathcal{D} = \{\mathcal{T}_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
    Compute rewards-to-go $\hat{R}_t$.
    Compute advantages estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the
    current value function $V_{\phi_k}$.
    Update the policy by maximizing the PPO-clip objective:
        $\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\mathcal{T}\in\mathcal{D}_k} \sum_{t=0}^{T} \min(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)),$
    typically via stochastic gradient ascent with Adam.
    Fit value function by regression on mean-squared error:
        $\phi_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\mathcal{T}\in\mathcal{D}_k} \sum_{t=0}^{T} (V_\theta(s_t) - \hat{R}_t)^2,$
    typically via stochastic gradient ascent algorithm.
end for
```

To further increase stability in the training and sampling efficiency, we will use generalized advantage estimation (GAE) (15) to compute our advantage function. PPO relies on policy gradients to optimize agent behavior, but using raw Monte Carle returns for advantage estimation can introduce high variance, which can lead to unstable learning. On the other hand, using temporal difference methods alone can reduce variance by often introducing bias. GAE strikes an optimal balance by smoothing advantage estimates by reducing variance without adding too much bias. This can actually help the clipping mechanism in PPO work more effectively by providing accurate advantage estimates, allowing for more stable policy updates. A more detailed look in how the advantage function is being calculated is illustrated in Equation 1.

$$
\begin{aligned}
[h]\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} &:= (1-\lambda)(\hat{A}_t^{(1)} + \hat{A}_t^{(1)} + \hat{A}_t^{(1)} + \dots) \\
&= (1-\lambda)(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \dots) \\
&= (1+\lambda)(\delta_t^V(1+\lambda+\lambda^2+\dots) + \gamma\delta_{t+1}^V(\lambda+\lambda^2+\lambda^3+\dots) \\
&\qquad + \gamma^2\delta_{t+2}^V(\lambda^2+\lambda^3+\lambda^4\dots) + \dots) \\
&= (1-\lambda)(\delta_t^V(\frac{1}{1-\lambda}) + \gamma\delta_{t+1}^V(\frac{\lambda}{1-\lambda}) + \gamma^2\delta_{t+2}^V(\frac{\lambda^2}{1-\lambda}) + \dots) \\
&= \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}^V \qquad (1)
\end{aligned}
$$

## 3 Evaluation

### 3.1 Metrics

Evaluation metrics were taken from the evaluation JSON file from the HuatuoGPT-o1 GitHub. The table below explains the benchmarks used, along with how many questions were used for evaluation. It is important to note that these are subsets of the original datasets. We used the HuatuoGPT-o1 evaluation datasets for consistency. For a more detail explanation of the specific content of each dataset see Table1.

| Benchmark | Description | # Questions |
|---|---|---|
| MedQA | The English subset of MedQA (USMLE) consists of multiple-choice questions from U.S. medical board exams. (18) | 1,273 |
| MedMCQA | MedMCQA is a large-scale multiple-choice medical QA from Indian medical entrance exams, designed to test a wide range of reasoning skills across diverse healthcare topics. (19) | 4,183 |
| PubMedQA | PubMedQA is a biomedical question answering dataset consisting of yes/no/maybe questions derived from PubMed article titles. (20) | 1,000 |

Table 1: Descriptions of medical QA benchmarks used in this study.

### 3.2 Main Results

For this section, we will be looking at the results of our fine-tuned distilled models only with SFT implemented. To see the specific reason why see Section3.2.4

#### 3.2.1 Accuracy

Using the MedQA, MedMCQA, and the PubMeqQA as our benchmarks, we can evaluate the performance of our fine-tuned distilled models for basic answer questions, that is how many questions the model when using only one inference.

| Model | MedMCQA | MedQA | PubMedQA |
|---|---|---|---|
| Llama-3.1-8B-Instruct | 0.561 | 0.661 | 0.752 |
| HuatuoGPT-o1-8B | 0.626 | 0.750 | 0.787 |
| DeepSeek-R1-Distill-Llama-8B | 0.502 | 0.564 | 0.740 |
| Fine-Tune-lr-1e-4 | 0.450 | 0.522 | 0.612 |
| Fine-Tune-lr-1e-6 | 0.475 | 0.534 | 0.723 |
| Fine-Tune-1-epoch | 0.498 | 0.563 | 0.738 |
| Fine-Tune-small-wd | 0.454 | 0.548 | 0.736 |

Table 2: Model Accuracy on MedMCQA, MedQA, and PubMedQA

From Table2, we can see that our fine-tuned distilled models generally cluster around the similar accuracy rate on three benchmarks. The performance on MedMCQA and MedQA shows a small spread in accuracy, suggesting that the ablated components had limited impact on those datasets. MedMCQA scores range from 0.450 to 0.498 (a spread of just 0.048), and MedQA from 0.522 to 0.563 (spread of 0.041), indicating consistent performance across ablation configurations.

In contrast, the performance on PubMedQA exhibits a much larger spread. Notably, Fine-Tune-Ablation-1 significantly underperforms with a score of 0.612, which is 0.111 lower than the next worst-performing model (Fine-Tune-Ablation-2 at 0.723). This sharp drop suggests that the component removed in Ablation-1 plays a critical role in enabling strong performance on PubMedQA, potentially due to the dataset's emphasis on fact-based biomedical inference and document understanding.

#### 3.2.2 pass@k

Using 100 random questions from MedQA as our benchmark we can also evaluate the performance of our fine-tuned distilled models using a pass@k metric, which measures the probability that at least one of k responses from a model for a given problem is correct (21). This will give further insight on the model's sampling ability to finding a given solution.

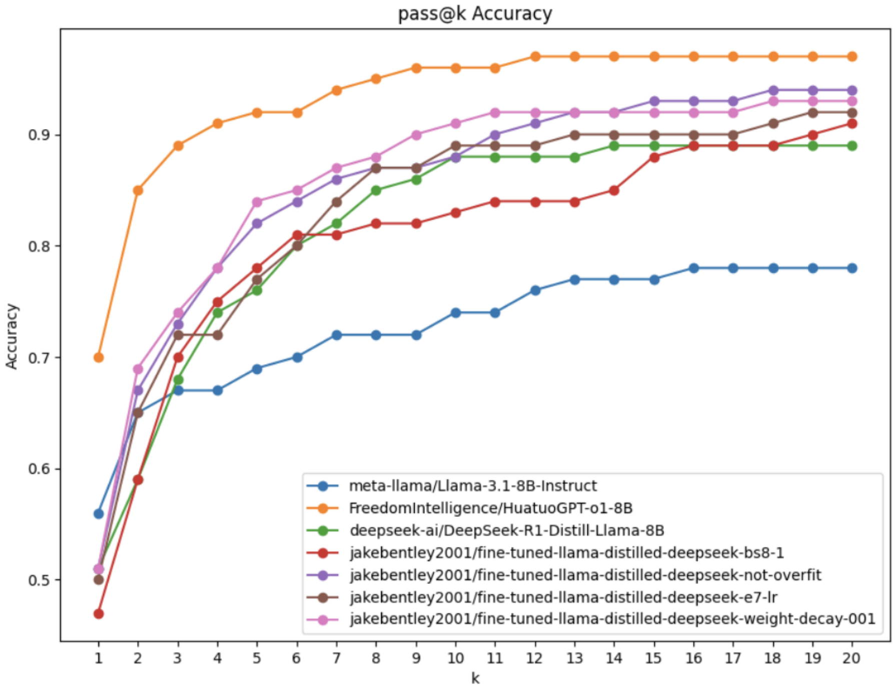

Figure 3: Performance of the baseline and our models for k=1 to k=20 for the pass@k metric

From Figure3 we can see that as k increases our fine-tuned distilled models steadily improve, eventually approaching the performance of HuatuoGPT-o1-8B model at higher k values (e.g., k=20). This indicates that while our models are capable of identifying the correct answer within a broader candidate set, they struggle to rank the correct answer highly, as evidenced by significantly lower accuracy at k=1.

### 3.2.3 Reasoning Capabilties

As a general heuristic, we use the number of characters as a proxy for reasoning quality: more characters typically indicate richer reasoning, while fewer characters suggest less developed reasoning.

| Model | MedMCQA | MedQA | PubMedQA |
|---|---|---|---|
| Llama-3.1-8B-Instruct | 138.20 | 547.06 | 47.73 |
| HuatuoGPT-o1-8B | 1716.06 | 2133.94 | 1841.03 |
| DeepSeek-R1-Distill-Llama-8B | 4469.98 | 6332.74 | 2839.87 |
| Fine-Tune-lr-1e-4 | 1978.88 | 2814.67 | 2340.64 |
| Fine-Tune-lr-1e-6 | 3495.09 | 4927.42 | 2450.51 |
| Fine-Tune-1-epoch | 4256.13 | 6072.71 | 2677.13 |
| Fine-Tune-small-wd | 3482.43 | 4886.34 | 2463.43 |

Table 3: Average number of characters in model output for MedMCQA, MedQA, and PubMedQA

From Table3 model output lengths vary significantly across MedMCQA, MedQA, and PubMedQA, with the DeepSeek-R1-Distill-Llama-8B model consistently generating the longest responses. In contrast, the Llama-3.1-8B-Instruct model produces very short answers, suggesting it may lack detail or thoroughness. Our models provide somewhat of a balance, with Fine-Tune-Ablation-3 yielding outputs nearly as verbose as DeepSeek, indicating a potentially effective trade-off between brevity and informativeness.

### 3.2.4 Incomplete RL Results

For this project, we originally planned to recreate Chen et al's SFT, then RL training pipeline. However, because of difficulties deploying the pipeline on the PSC clusters and a delay in investigated why the SFT results were not as strong as anticipated—especially given that Chen et al suggested RL would only provide a modest 2% improvement. As a result, we concentrated our efforts on investigating the underlying causes of the weak SFT performance, aiming to better understand the factors contributing to the gap between our results and those reported by Chen et al before moving on with experiments for the RL.

## 3.3 Qualitative Analysis

While achieving comparable performance on these general benchmarks may suggest a model's overall competence, selecting the correct answer on these tests does not necessary reflect true reasoning ability within the medical domain. Many benchmarks rely on surface-level pattern recognition and fail to capture the depth of understanding required for tasks like interpreting nuanced symptoms, managing uncertainty, or synthesizing disparate clinical findings. In medicine, we care not just about the correct answer, but whether it was reach through logical consistency, faithfulness to the data, and often multi-step reasoning — the kind that mirrors expert clinical judgment. This disconnect highlights the limitations of pure quantitative metrics and motivates a need for qualitative analysis. Given these concerns, we will examine *logical reasoning*, *faithful reasoning*, and *multi-hop reasoning* (22).

**Logical Reasoning** Logical reasoning is fundamental as it serves the bedrock for rational thinking, robust problem-solving and interpretable decision-making (23; 24; 25). This is illustrated by the chain-of-thought reasoning generated in our model's output, which exhibits a clear reasoning pattern: it evaluates each option step by step, compares hypotheses against known facts, eliminates inconsistencies, and arrives at a justified conclusion — reflecting interpretable and structured decision-making. However, this pattern of reasoning, while interpretable, can still go astray if the model's underlying knowledge is flawed or misapplied. For instance, it may follow a seemingly logical sequence of steps but arrive at an incorrect answer due to false premises, overgeneralization, or failure to recognize contextual nuances. A more detailed flow of this reasoning process can be found in Figure 4.

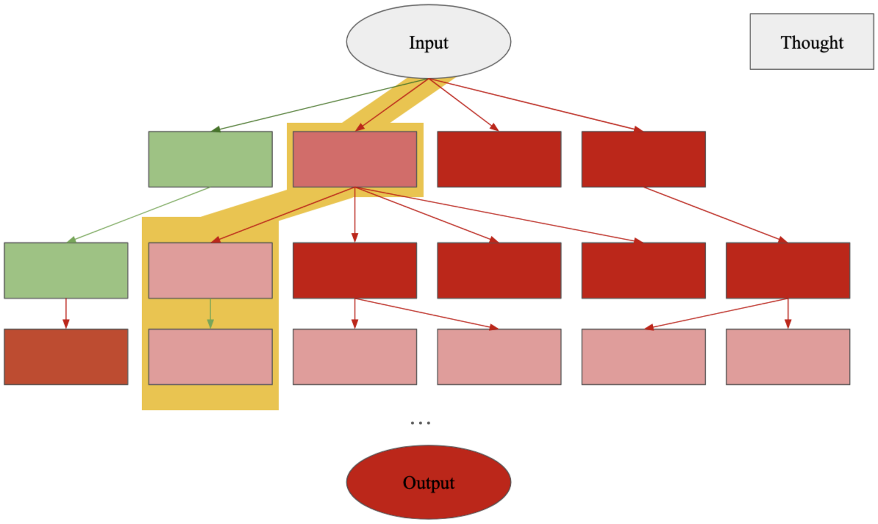

Figure 4: Chain-of-Thought. Model reasons itself into thinking a wrong answer is correct to arrive at a wrong answer.

**Faithful Reasoning** Faithful reasoning is foundational as it ensures consistency with established truths, integrity in interpretation, and trustworthiness in conclusions (26; 27; 28). This is illustrated by the chain-of-thought reasoning generated in our model's output, where it consistently aligns its evaluation with established physiological principles, accurately interprets each option, and derives a trustworthy conclusion. However, even faithful reasoning can falter if the model's underlying information is misapplied. For example, if the model applies an inaccurate or incomplete understanding of a concept, it may generate a conclusion that, while logically coherent, is ultimately untrustworthy. A more detailed flow of this reasoning process can be found in Figure 5.

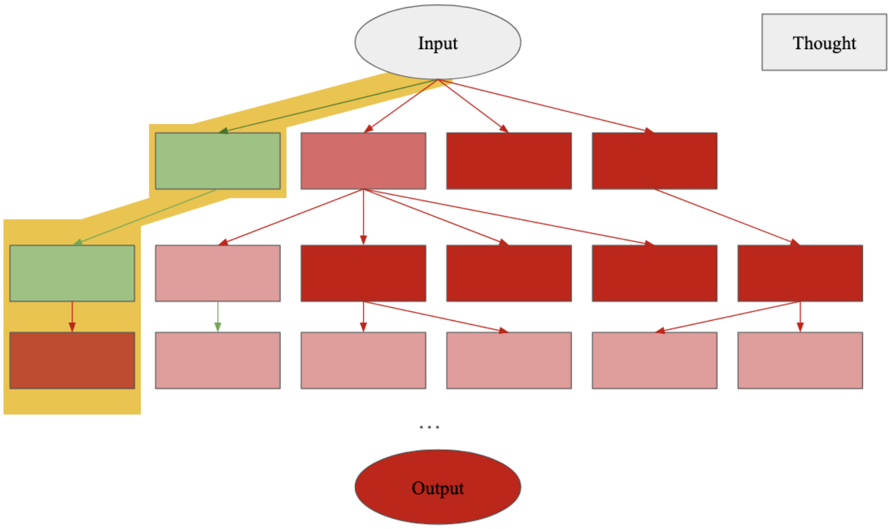

Figure 5: Chain-of-Thought. Model misapplies correct premise to arrive at wrong answer.

**Multi-hop Reasoning** Multi-hop reasoning is essential as it enables the integration of dispersed facts, the construction of complex inferences, and the resolution of intricately layered queries (29; 30). This is illustrated by the chain-of-thought reasoning generated in our model's output, where it successfully connects multiple pieces of information across different reasoning steps, building on each to form a comprehensive and well-supported conclusion. However, even multi-hop reasoning can go awry if any individual step is flawed or if the model fails to link relevant facts correctly. For example, a mistake in one intermediate inference could lead to a faulty conclusion, disrupting the entire reasoning process. A more detailed flow of this reasoning process can be found in Figure 6.

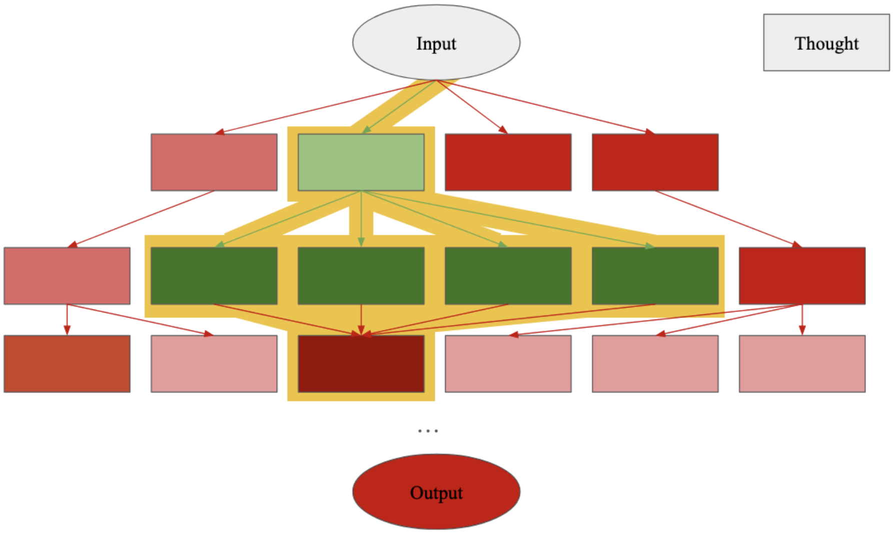

Figure 6: Chain-of-Thought. Model fails to correct synthesize correct premises to arrive at a wrong answer.

## 4 Conclusion

From our experiments, we can draw the following conclusion. SFT memorizes and RL generalizes. This claim is supported by recent literature namely (31; 32; 33). SFT helps learn the sample space and RL helps sample from this. We can see this through Figure 3 as starting pass@k performance is poorer than the Huatuo-GPT-o1-8B a model that uses both SFT and RL. Our findings provide further evidence for this trend, aligning with prior work and reinforcing the distinction between memorization via SFT and generalization via RL.

## 5 Code Availability

Code can be found on GitHub: `https://github.com/siddsabata/MR-DeepSeek`

## References

[1] A. Pareja, N. S. Nayak, H. Wang, K. Killamsetty, S. Sudalairaj, W. Zhao, S. Han, A. Bhandwaldar, G. Xu, K. Xu, L. Han, L. Inglis, and A. Srivastava, "Unveiling the secret recipe: A guide for supervised fine-tuning small llms," 2024. [Online]. Available: https://arxiv.org/abs/2412.13337

[2] M. Ghasemi, A. H. Moosavi, I. Sorkhoh, A. Agrawal, F. Alzhouri, and D. Ebrahimi, "An introduction to reinforcement learning: Fundamental concepts and practical applications," *ArXiv*, vol. abs/2408.07712, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271874853

[3] J. Chen *et al.*, "HuatuoGPT-o1, Towards Medical Complex Reasoning with LLMs," *arXiv preprint arXiv:2412.18925*, Dec. 2024. [Online]. Available: https://arxiv.org/abs/2412.18925

[4] DeepSeek-AI *et al.*, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," *arXiv preprint arXiv:2501.12948*, Jan. 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[5] E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint arXiv:2106.09685*, Oct. 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[6] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," May 2023, arXiv:2305.14314 [cs]. [Online]. Available: http://arxiv.org/abs/2305.14314

[7] J. Wei *et al.*, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *arXiv preprint arXiv:2201.11903*, Jan. 2023. [Online]. Available: https://arxiv.org/abs/2201.11903

[8] X. Wang *et al.*, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," *arXiv preprint arXiv:2203.11171*, Mar. 2023. [Online]. Available: https://arxiv.org/abs/2203.11171

[9] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large Language Models are Zero-Shot Reasoners," *arXiv preprint arXiv:2205.11916*, Jan. 2023. [Online]. Available: https://arxiv.org/abs/2205.11916

[10] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," *arXiv preprint arXiv:2202.10054*, 2022. [Online]. Available: https://arxiv.org/abs/2202.10054

[11] E. Lobo, C. Agarwal, and H. Lakkaraju, "On the impact of fine-tuning on chain-of-thought reasoning," *arXiv preprint arXiv:2411.15382*, 2025. [Online]. Available: https://arxiv.org/pdf/2411.15382

[12] X. Zhu, B. Qi, K. Zhang, X. Long, Z. Lin, and B. Zhou, "Pad: Program-aided distillation can teach small models reasoning better than chain-of-thought fine-tuning," *arXiv preprint arXiv:2305.13888*, 2023. [Online]. Available: https://arxiv.org/abs/2305.13888

[13] T. Ni, A. Nie, S. Chaudhary, Y. Liu, H. Rangwala, and R. Fakoor, “Teaching large language models to reason through learning and forgetting,” *arXiv preprint arXiv:2504.11364*, 2025. [Online]. Available: https://arxiv.org/pdf/2504.11364

[14] C.-C. Wu, Z. R. Tam, C.-Y. Lin, H. yi Lee, and Y.-N. Chen, “Clear minds think alike: What makes llm fine-tuning robust? a study of token perplexity,” *arXiv preprint arXiv:2501.14315v1*, 2025. [Online]. Available: https://arxiv.org/pdf/2501.14315v1

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” Aug. 2017, arXiv:1707.06347 [cs]. [Online]. Available: http://arxiv.org/abs/1707.06347

[16] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, “Direct Preference Optimization: Your Language Model is Secretly a Reward Model,” Jul. 2024, arXiv:2305.18290 [cs]. [Online]. Available: http://arxiv.org/abs/2305.18290

[17] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, “Trust Region Policy Optimization,” Apr. 2017, arXiv:1502.05477 [cs]. [Online]. Available: http://arxiv.org/abs/1502.05477

[18] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits, “What disease does this patient have? a large-scale open domain question answering dataset from medical exams,” 2020. [Online]. Available: https://arxiv.org/abs/2009.13081

[19] A. Pal, L. K. Umapathi, and M. Sankarasubbu, “Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering,” 2022. [Online]. Available: https://arxiv.org/abs/2203.14371

[20] Q. Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu, “Pubmedqa: A dataset for biomedical research question answering,” 2019. [Online]. Available: https://arxiv.org/abs/1909.06146

[21] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, “Evaluating large language models trained on code,” 2021. [Online]. Available: https://arxiv.org/abs/2107.03374

[22] Z. Chu, J. Chen, Q. Chen, W. Yu, T. He, H. Wang, W. Peng, M. Liu, B. Qin, and T. Liu, “Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future,” 2024. [Online]. Available: https://arxiv.org/abs/2309.15402

[23] I. Dasgupta, A. K. Lampinen, S. C. Y. Chan, H. R. Sheahan, A. Creswell, D. Kumaran, J. L. McClelland, and F. Hill, “Language models show human-like content effects on reasoning tasks,” 2022. [Online]. Available: https://arxiv.org/abs/2207.07051

[24] S. Han, H. Schoelkopf, Y. Zhao, Z. Qi, M. Riddell, W. Zhou, J. Coady, D. Peng, Y. Qiao, L. Benson, L. Sun, A. Wardle-Solano, H. Szabo, E. Zubova, M. Burtell, J. Fan, Y. Liu, B. Wong, M. Sailor, A. Ni, L. Nan, J. Kasai, T. Yu, R. Zhang, A. R. Fabbri, W. Kryscinski, S. Yavuz, Y. Liu, X. V. Lin, S. Joty, Y. Zhou, C. Xiong, R. Ying, A. Cohan, and D. Radev, “Folio: Natural language reasoning with first-order logic,” 2022. [Online]. Available: https://arxiv.org/abs/2209.00840

[25] J. Huang and K. C.-C. Chang, “Towards reasoning in large language models: A survey,” 2022. [Online]. Available: https://arxiv.org/abs/2212.10403

[26] A. Creswell and M. Shanahan, “Faithful reasoning using large language models,” 2022. [Online]. Available: https://arxiv.org/abs/2208.14271

[27] T. Lanham, A. Chen, A. Radhakrishnan, B. Steiner, C. Denison, D. Hernandez, D. Li, E. Durmus, E. Hubinger, J. Kernion, K. Lukošiūtė, K. Nguyen, N. Cheng, N. Joseph, N. Schiefer, O. Rausch, R. Larson, S. McCandlish, S. Kundu, S. Kadavath, S. Yang, T. Henighan, T. Maxwell, T. Telleen-Lawton, T. Hume, Z. Hatfield-Dodds, J. Kaplan, J. Brauner, S. R. Bowman, and E. Perez, "Measuring faithfulness in chain-of-thought reasoning," 2023. [Online]. Available: https://arxiv.org/abs/2307.13702

[28] Q. Lyu, S. Havaldar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, and C. Callison-Burch, "Faithful chain-of-thought reasoning," 2023. [Online]. Available: https://arxiv.org/abs/2301.13379

[29] J. Wang, J. Li, and H. Zhao, "Self-prompted chain-of-thought on large language models for open-domain multi-hop reasoning," 2023. [Online]. Available: https://arxiv.org/abs/2310.13552

[30] S. Yang, E. Gribovskaya, N. Kassner, M. Geva, and S. Riedel, "Do large language models latently perform multi-hop reasoning?" 2024. [Online]. Available: https://arxiv.org/abs/2402.16837

[31] T. Q. Luong, X. Zhang, Z. Jie, P. Sun, X. Jin, and H. Li, "Reft: Reasoning with reinforced fine-tuning," 2024. [Online]. Available: https://arxiv.org/abs/2401.08967

[32] Y. Yue, Z. Chen, R. Lu, A. Zhao, Z. Wang, Y. Yue, S. Song, and G. Huang, "Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?" 2025. [Online]. Available: https://arxiv.org/abs/2504.13837

[33] T. Chu, Y. Zhai, J. Yang, S. Tong, S. Xie, D. Schuurmans, Q. V. Le, S. Levine, and Y. Ma, "Sft memorizes, rl generalizes: A comparative study of foundation model post-training," 2025. [Online]. Available: https://arxiv.org/abs/2501.17161

[34] G. Dong *et al.*, "How Abilities in Large Language Models are Affected by Supervised Fine-tuning Data Composition," *arXiv preprint arXiv:2310.05492*, Jun. 2024. [Online]. Available: https://arxiv.org/abs/2310.05492

[35] "DeepSeek R1 Distill Llama 70B - Intelligence, Performance & Price Analysis | Artificial Analysis." [Online]. Available: https://artificialanalysis.ai/models/deepseek-r1-distill-llama-70b

[36] "Introducing Gemma 3: The most capable model you can run on a single GPU or TPU," Mar. 2025. [Online]. Available: https://blog.google/technology/developers/gemma-3/

[37] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," Oct. 2018, arXiv:1506.02438 [cs]. [Online]. Available: http://arxiv.org/abs/1506.02438