# Mixture of Reasonings: Teach Large Language Models to Reason with Adaptive Strategies

Tao Xiong<sup>1,†</sup>, Xavier Hu<sup>2,†</sup>, Wenyan Fan<sup>2</sup>, Shengyu Zhang<sup>2,‡</sup>

<sup>1</sup>Dalian University of Technology, <sup>2</sup>Zhejiang University

Correspondence: 030130x@mail.dlut.edu.cn, sy\_zhang@zju.edu.cn

## **Abstract**

Large language models (LLMs) excel in complex tasks through advanced prompting techniques like Chain-of-Thought (CoT) and Treeof-Thought (ToT), but their reliance on manually crafted, task-specific prompts limits adaptability and efficiency. We introduce Mixture of Reasoning (MoR), a training framework that embeds diverse reasoning strategies into LLMs for autonomous, task-adaptive reasoning without external prompt engineering. MoR has two phases: Thought Generation, creating reasoning chain templates with models like GPT-40, and SFT Dataset Construction, pairing templates with benchmark datasets for supervised fine-tuning. Our experiments show that MoR significantly enhances performance, with  $MoR_{150}$  achieving 0.730 (2.2% improvement) using CoT prompting and 0.734 (13.5% improvement) compared to baselines. MoR eliminates the need for task-specific prompts, offering a generalizable solution for robust reasoning across diverse tasks.

## 1 Introduction

Large language models (LLMs) have achieved remarkable success across diverse domains, largely due to advanced prompting techniques such as Chain-of-Thought (CoT) (Wei et al., 2023), Tree-of-Thought (ToT) (Yao et al., 2023), and Prompt-of-Thought (PoT) (Zhu et al., 2024). These methods guide models to reason step-by-step or explore multiple reasoning paths, significantly enhancing their performance on complex tasks. However, their effectiveness heavily relies on manually crafted, task-specific prompts, which are time-consuming to design and challenging to adapt optimally across varied tasks. This dependency on prompt engineering poses a critical bottleneck, where generic prompts often fail to elicit robust reasoning.

To address this challenge, we propose Mixture of Reasoning (MoR), a novel training framework that embeds a diverse set of reasoning strategies directly into LLMs, enabling them to autonomously select and apply effective reasoning methods tailored to specific tasks. Unlike existing approaches (Gao et al., 2024; Zhou et al., 2024) that rely on external prompt engineering to elicit reasoning, MoR internalizes reasoning capabilities by finetuning models on a curated supervised fine-tuning (SFT) dataset enriched with reasoning chain templates. These templates, generated by leveraging the advanced reasoning abilities of closed-source large models (e.g., GPT-40), cover a wide range of reasoning patterns, including multi-step deduction, analogical reasoning, and strategic thinking. The MoR framework operates in two key phases: (1) Thought Generation, where we produce largescale reasoning chain templates (e.g., 50, 150, 300, and 500 chains) to capture diverse problem-solving approaches, and (2) SFT Dataset Construction, where we pair these templates with samples from benchmark datasets to create a training dataset that teaches models to adaptively apply reasoning strategies. By embedding these strategies into the model's parameters, MoR eliminates the need for task-specific prompt design and enhances generalizability across complex reasoning tasks.

Our experiments demonstrate that MoR significantly outperforms baseline models, with our best model,  $MoR_{150}$ , achieving a performance of 0.730 with CoT prompting (a 2.2% improvement over the baseline) and 0.734 with direct IO prompting (a 13.5% improvement), showcasing its ability to reason effectively without explicit guidance.

Our contributions are as follows:

 We introduce MoR, a training framework that embeds diverse reasoning strategies into LLMs, enabling task-adaptive reasoning without reliance on specific prompts.

 $<sup>^{\</sup>dagger}Both$  authors contributed equally to this research.  $^{\ddagger}Corresponding$  Author

- We propose a two-step methodology involving Thought Generation and SFT Dataset Construction, leveraging large-scale reasoning templates and curated datasets.
- We provide comprehensive experimental evidence demonstrating MoR's superiority over baseline models, with detailed analyses and case studies illustrating its logical reasoning capabilities.

## 2 Related Work

Supervised Fine-Tuning of Large Language **Models.** Supervised Fine-Tuning (SFT) (Zhang et al., 2024) leverages structured (instructionanswer) pairs to fully exploit the zero-shot capabilities of large models. This process enables models to learn systematic reasoning patterns and produce accurate results on complex reasoning tasks. By fine-tuning on task-specific datasets, SFT emphasizes the development of logical reasoning, problem-solving skills, and domain-specific knowledge. In recent years, numerous studies on SFT for large models have emerged, including approaches such as zeroth-order fine-tuning (Malladi et al., 2024) and robust fine-tuning (Tian et al., 2023). Notably, SFT has demonstrated significant advantages in reasoning-related fields, particularly in mathematics (Cobbe et al., 2021; Chen et al., 2024) and code generation (Wang et al., 2024a), achieving promising results.

Prompt Engineering. Thoughtful prompt design can enhance the reasoning abilities of large models, helping them tackle complex challenges. Chain-of-thought prompting is a strategy that guides large language models (LLMs) to produce intermediate reasoning steps, ultimately leading to the final answer and improving problem-solving accuracy. Typical implementations include zeroshot CoT (Kojima et al., 2023) and few-shot CoT (Wei et al., 2023). Recent studies (Yasunaga et al., 2024; Zheng et al., 2024; Wang et al., 2024b; Wilf et al., 2023) have further advanced this method by integrating more structured algorithms and search strategies. For example, Zheng et al. (2024) enables LLMs to abstract high-level concepts and first principles from detailed instances, while Yasunaga et al. (2024) prompts models to generate relevant examples or contextual knowledge before solving the problem. Additionally, some research (Gao et al., 2024; Zhou et al., 2024) is also exploring the use of different types of reasoning chains

tailored to various task categories. Our approach, MoR, differs from these methods in that it not only produces a diverse array of reasoning strategies but also employs supervised fine-tuning (SFT) to train a foundational model capable of multi-chain reasoning.

## 3 Method

In this section, we will provide a detailed description of the specific implementation of the MoR method. *The framework is shown in Figure 1*, and we have divided the MoR method into two steps: (1) Thought Generation: generating multiple thought chains to expand the model's thinking approach. (2) SFT Dataset Construction: creating an SFT training dataset using various thinking approaches.

## 3.1 Thought Generation

For small parameter models, due to insufficient embedded knowledge and limited reasoning capabilities, simply instructing them with "Let's think step by step" does not effectively stimulate the model's capabilities.

To address this issue, we first need to provide the model with effective thinking approaches for different types of problems. Existing methods (Wei et al., 2023; Yasunaga et al., 2024; Zheng et al., 2024) mainly focus on generating specific thinking approaches for one type of problem. We decided to leverage the reasoning ability of closed-source large models. Initially, we prompted GPT to generate a large number of reasoning chain templates for reasoning tasks. In this section, we pre-generated  $\underline{\bf 50}$ ,  $\underline{\bf 150}$ ,  $\underline{\bf 300}$ , and  $\underline{\bf 500}$  reasoning chains, denoted as  $T=t_1,t_2,...,t_M$ .

## 3.2 SFT Dataset Construction

After generating the reasoning chains in §3.1, we need to construct an MoR dataset that can be used for training. In this section, we select several commonly used reasoning datasets, such as HotpotQA, StrategyQA, MMLU, BigTom, and Trivial Creative Writing (more details will be discussed in §4.1).

First, we randomly select Specified quantity samples  $\mathbf N$  from each dataset as training samples. Then, for the selected dataset  $D_{\text{source}} = \{s_1, s_2, \dots, s_K\}$ , where K=N, we randomly select 5 reasoning chain templates  $T_{\text{sub}}$  from the reasoning chain template set  $T = \{t_1, t_2, \dots, t_M\}$ , forming a subset of reasoning chains. The selected samples  $D_{\text{selected}}$  along

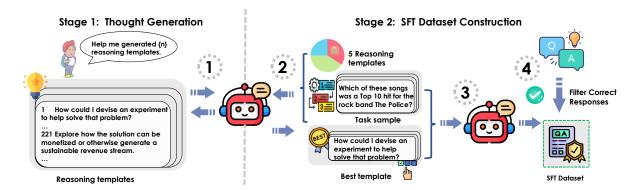


Figure 1: Overview of our proposed MoR framework. The MoR framework can be divided into two stages: (1) Thought Generation. As shown in step 1, this involves generating a large number of reasoning chain templates using GPT. (2) SFT Dataset Construction. As depicted in steps 2, 3, and 4, this includes selecting optimal reasoning chains, creating prompts, and filtering for correct responses.

with the selected subset are then fed into GPT, which selects the reasoning chain  $T_{\rm best}$  it deems most beneficial for solving the problem based on the problem structure of the samples. Next, we create a prompt by combining the selected reasoning chain template  $T_{\rm best}$  with each sample  $s_i$ , and feed it to the model for reasoning. After evaluation, we filter out the correct answers, and the resulting set is combined into an SFT dataset  $D_{\rm SFT}$ .

# Algorithm 1 SFT Dataset Construction

```
1: D_{SFT} \leftarrow \emptyset
 2: for i \leftarrow 1 to N do
          s_i \leftarrow D_{\text{selected}}[i] // \text{ Get the } i\text{-th sample}
          T_{\text{sub}} \leftarrow \text{RandomSelect}(T, N) // \text{Select } N
     templates
 5:
          Prompt_{select}
     FormatSelectPrompt(s_i, T_{\text{sub}})
           t_{\text{best}} \leftarrow \text{LLM.infer}(Prompt_{\text{select}})
 6:
 7:
           Prompt_{reason}
     FormatReasonPrompt(s_i, t_{best})
           R_i \leftarrow \text{model.infer}(Prompt_{\text{reason}})
 8:
          IsCorrect \leftarrow Eval(s_i, R_i) //Evaluate if
     R_i is correct for s_i
          if IsCorrect is True then
10:
                SFT_{\text{entry}} \leftarrow \text{FormatForSFT}(s_i, R_i)
11:
                D_{SFT} \leftarrow D_{SFT} \cup \{SFT_{entry}\}
12:
13:
          end if
14: end for
15: return D_{SFT} //Return the constructed SFT
```

# 4 Experiment

## 4.1 Setup

**Datasets.** In the experiment, we selected five reasoning datasets, with 50 samples randomly chosen from each dataset for testing. For BigTom, we selected 20 samples across four different "belief settings," totaling 80 samples. The SFT dataset construction used the GPT-40-2024-08-06 version of GPT, as mentioned in §3.

- HotpotQA (Yang et al., 2018): HotpotQA is designed for question answering with complex, multi-hop questions and strong supervision for interpretable systems.
- StrategyQA (Geva et al., 2021): StrategyQA requiring inference of reasoning steps for question answering through strategic thinking.
- MMLU (Hendrycks et al., 2021): MMLU is an extensive multitask benchmark composed of multiple-choice questions across a wide range of knowledge domains. The benchmark spans 57 subjects across diverse domains.
- **BigTom** (Wilf et al., 2023): BigTom is a benchmark for assessing the Theory of Mind (ToM) reasoning abilities of large language models (LLMs). It includes a new social reasoning framework with 25 controls and 5,000 model-generated evaluations.
- Trivial Creative Writing (Wang et al., 2024b): This dataset challenges models to generate a coherent story while seamlessly incorporating answers to a set of trivia questions.

Method		Hotpotga	Strate-	MMLU	BigTom	Trivial Creative	overall
Model	Prompt		gyqa			writing	
Qwen2.5-7B	IO	1.00	0.400	0.540	0.688	0.368	0.599
	CoT	0.980	0.940	0.560	0.750	0.308	0.708
$MoR_{50}$	IO	0.540	0.900	0.580	0.888	0.336	0.649
	CoT	0.640	0.480	0.580	0.925	0.300	0.585
$MoR_{150}$	IO	0.98	0.94	0.560	0.875	0.144	0.700
	CoT	0.98	0.920	0.620	0.900	0.232	0.730
$MoR_{300}$	IO	0.980	0.840	0.480	0.938	0.208	0.689
	CoT	0.980	0.880	0.560	0.863	0.292	0.715
$MoR_{500}$	IO	0.960	0.920	0.620	0.913	0.256	0.734
	CoT	0.960	0.900	0.500	0.900	0.276	0.707
Qwen2.5-7B	IO	0.960	0.400	0.595	0.731	0.368	0.611
(Expend)	CoT	0.915	0.885	0.565	0.738	0.308	0.682
$MoR_{150}(Expend)$	1) IO	0.990	0.880	0.610	0.863	0.144	0.697
	CoT	0.960	0.905	0.600	0.919	0.232	0.723

Table 1: Performance on reasoning tasks. We selected Qwen2.5-7B-instruct as the baseline model. We train the baseline model using our MoR approach by varying the number of reasoning chain templates. Additionally, to evaluate the effectiveness of MoR, we extend the test set from 50 to 200 instances, specifically comparing the baseline model with  $MoR_{150}$ . The best results for each setting are bolded.

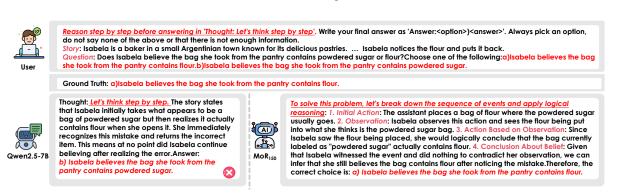


Figure 2: Case study comparing the baseline model and  $MoR_{150}$  using CoT prompts. The Qwen2.5-7B-instruct model follows the "Let's think step by step." approach but ultimately produces incorrect answers. In contrast, the  $MoR_{150}$  model adopts the MoR reasoning method, analyzing problems logically and ultimately arriving at the correct answer.

**Model.** We selected the Qwen2.5-7B-Instruct (Qwen et al., 2025) model as the baseline. The models fine-tuned on different numbers of X-chain of thought datasets are used as our comparison models, denoted as  $MoR_i, wherei = 50, 150, 300, 500$ . We believe that after training, the model has acquired MoR capabilities, so simply using the prompt "Let's think step by step." is sufficient to elicit the model's multi-step reasoning ability. We refer to this prompting strategy as the CoT prompt. For comparison, we also provide a setting where the model is directly instructed to answer the question without any special prompt which is called the IO prompt.

## 4.2 Result

The summarized results in Table 1 clearly demonstrate that models trained using the MoR approach achieve substantial and consistent improvements across a wide range of reasoning tasks. Notably, the performance with the Chain-of-Thought (CoT) prompt reaches an accuracy of 0.730, representing a 2.2% increase over the baseline model, which underscores the effectiveness of structured reasoning in enhancing model capabilities. Interestingly, the highest performance is observed with the Input-Output (IO) prompt, which attains a score of 0.734—exceeding the baseline by a remarkable 13.5%. This suggests that, while the CoT

prompting strategy effectively fosters deeper reasoning, the IO prompts still hold significant value for straightforward tasks.

## 4.3 Analysis

# Analysis of results.

For simple tasks like HotpotQA, most models perform well, with some achieving perfect scores, indicating that basic models are already effective for direct question-answering. However, for complex tasks like StrategyQA and MMLU, MoR models using Chain-of-Thought (CoT) prompts show superior performance, highlighting the importance of structured reasoning chains for complex tasks. The experiments reveal that increasing reasoning templates doesn't always improve performance, especially with limited training data. The  $MoR_{150}$  configuration achieved the optimal chain-of-thought stimulation, and as MoR's chainof-thought and data grow, explicit guidance may be less necessary, with the IO prompt effectively stimulating reasoning in  $MoR_{500}$ , achieving a best result of 0.734.

The MoR approach outperforms traditional methods, particularly in multi-step inference and strategy-oriented tasks. While CoT and IO prompts perform similarly, the IO prompt provides a slight advantage in some tasks, showcasing task-specific benefits. These results confirm that integrating MoR training with tailored prompts enhances reasoning abilities, advancing AI in complex problem-solving.

To verify these results, we expanded the test set for both the baseline model and  $MoR_{150}$  to 200 samples. As shown in Table X, the extended  $MoR_{150}$  maintains a consistent advantage over the baseline.

Case study of MoR methods. In Figure 2, we compare the baseline model with  $MoR_{150}$  on the BigTom dataset under CoT. This task evaluates LLMs' ability to reason about others' mental states and false beliefs. The baseline model fails to consider the protagonist's changing beliefs, leading to incomplete reasoning and incorrect answers. In contrast, the MoR model selects effective strategies, applying logical thinking to solve the problem correctly. This example demonstrates MoR's strength in theory of mind reasoning, providing superior understanding of complex mental states compared to traditional methods.

## 5 Conclusion

The Mixture of Reasoning (MoR) framework represents a significant advancement in enhancing the reasoning capabilities of large language models by embedding diverse reasoning strategies directly into their parameters. By eliminating the dependency on manually crafted, task-specific prompts, MoR enables LLMs to autonomously select and apply effective reasoning methods tailored to a wide range of complex tasks. Through our two-phase approach—Thought Generation and SFT Dataset Construction—we have demonstrated that MoR not only improves performance over baseline models but also achieves robust generalizability, as evidenced by  $MoR_{150}$ 's superior results of 0.730 with CoT prompting and 0.734. These findings underscore MoR's potential to redefine how LLMs approach reasoning, offering a scalable and adaptable solution that reduces the burden of prompt engineering. Future work will explore expanding the diversity of reasoning templates and integrating MoR with other advanced training paradigms to further enhance its effectiveness across even more challenging domains.

## References

Zhaorun Chen, Zhuokai Zhao, Zhihong Zhu, Ruiqi Zhang, Xiang Li, Bhiksha Raj, and Huaxiu Yao. 2024. Autoprm: Automating procedural supervision for multi-step reasoning via controllable question decomposition. *Preprint*, arXiv:2402.11452.

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. *Preprint*, arXiv:2110.14168.

Peizhong Gao, Ao Xie, Shaoguang Mao, Wenshan Wu, Yan Xia, Haipeng Mi, and Furu Wei. 2024. Meta reasoning for large language models. *Preprint*, arXiv:2406.11698.

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. *Preprint*, arXiv:2101.02235.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large

- language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.
- Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. 2024. Fine-tuning language models with just forward passes. *Preprint*, arXiv:2305.17333.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.
- Junjiao Tian, Yen-Cheng Liu, James Seale Smith, and Zsolt Kira. 2023. Fast trainable projection for robust fine-tuning. *Preprint*, arXiv:2310.19182.
- Yejie Wang, Keqing He, Guanting Dong, Pei Wang, Weihao Zeng, Muxi Diao, Yutao Mou, Mengdi Zhang, Jingang Wang, Xunliang Cai, and Weiran Xu. 2024a. Dolphcoder: Echo-locating code large language models with diverse and multi-objective instruction tuning. *Preprint*, arXiv:2402.09136.
- Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024b. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *Preprint*, arXiv:2307.05300.
- 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. *Preprint*, arXiv:2201.11903.
- Alex Wilf, Sihyun Shawn Lee, Paul Pu Liang, and Louis-Philippe Morency. 2023. Think twice: Perspective-taking improves large language models' theory-of-mind capabilities. *Preprint*, arXiv:2311.10227.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.
- Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, and Denny Zhou. 2024. Large language models as analogical reasoners. *Preprint*, arXiv:2310.01714.

- Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.
- Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. *Preprint*, arXiv:2310.06117.
- Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2024. Self-discover: Large language models self-compose reasoning structures. *Preprint*, arXiv:2402.03620.
- Shoutai Zhu, Ziqiang Yuan, Kaiyuan Wang, Yishu Zhang, and Wenqi Wei. 2024. Enhancing financial reasoning in large language models: The role of gold facts. In 2024 IEEE International Conference on Big Data (BigData), pages 1919–1928.