



# A short survey on small reasoning models: training, inference, applications, and research directions

Chengyu WANG<sup>1\*</sup>✉, Taolin ZHANG<sup>2\*</sup>, Richang HONG<sup>2</sup>, Jun HUANG<sup>1</sup>

1. Alibaba Cloud Computing, Hangzhou 310020, China

2. School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China

Received July 9, 2025; accepted September 9, 2025

E-mail: chengyu.wcy@alibaba-inc.com. \* These authors contributed equally to this work.

© The Author(s) 2026. This article is published with open access at [link.springer.com](https://link.springer.com) and [journal.hep.com.cn](https://journal.hep.com.cn)

## Abstract

Recently, the reasoning capabilities of Large Reasoning Models (LRMs), such as DeepSeek-R1, have witnessed significant advancements through computationally intensive “slow thinking” processes. These models have demonstrated impressive performance across a variety of complex reasoning tasks. However, despite their remarkable success, LRMs come with substantial computational demands that pose considerable challenges in terms of resource consumption, scalability, and accessibility. In contrast, Small Reasoning Models (SRMs), which are often distilled from larger models, offer a more efficient alternative while still achieving competitive performance. Beyond their efficiency, SRMs frequently exhibit distinct capabilities and cognitive trajectories compared with their larger counterparts, making them particularly interesting from both practical and theoretical perspectives. In this work, we provide a timely and comprehensive survey of recently published research focused on SRMs. We first review the current landscape of SRMs. Then, we analyze diverse training paradigms and inference techniques tailored to enhance the reasoning capabilities of SRMs. Furthermore, we offer an extensive review of domain-specific applications where SRMs have been effectively leveraged. Finally, we discuss promising future research directions that aim to bridge existing gaps. By consolidating recent advances, this survey serves as an essential reference for researchers and practitioners interested in leveraging or developing SRMs to unlock advanced reasoning functionalities with improved efficiency.

## Keywords

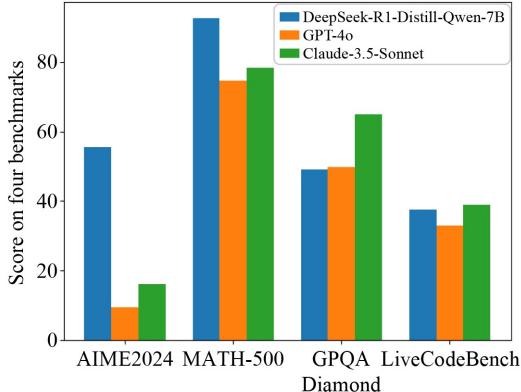
small reasoning model; model training; model inference; domain-specific applications

## ■ 1 Introduction

Recently, the field of Natural Language Processing (NLP) has undergone a profound transformation driven by the advent of Large Language Models (LLMs) [1], which exhibit remarkable capabilities across a diverse array of downstream tasks. Among these, Large Reasoning Models (LRMs) [2], such as DeepSeek-R1 [3], stand out for their specialization in solving complex reasoning challenges. These challenges span domains including mathematical problem-solving, code generation, and logical inference, achieved through the implementation of computationally intensive slow thinking processes (e.g., multi-step Chain-of-Thought (CoT) reasoning [4]). However, the impressive performance of these LRMs comes at a substantial cost, requiring extensive computational resources for both training and inference. For example, DeepSeek-R1 contains approximately 671 billion parameters and mandates servers equipped with at least eight NVIDIA A100 GPUs (80 GB) or equivalently powerful hardware configurations to enable efficient online deployment.

This high barrier in computational demand has spurred growing interest in the research community to explore the use of significantly

smaller models [5–8], which aim to offer more efficient yet effective alternatives to LRMs for complex reasoning tasks. Following the release of DeepSeek-R1, the open-source community has witnessed numerous breakthroughs demonstrating that Small Reasoning Models (SRMs), equipped with slow-thinking mechanisms such as extended Chain-of-Thought processes [4], can surpass much larger LLMs on certain specialized reasoning benchmarks, as illustrated in Fig. 1. Formally, we define Small Reasoning Models (SRMs) as language models with a substantially reduced parameter count, typically fewer than 10 billion parameters, that have been specifically fine-tuned or trained on reasoning-intensive tasks to generate multi-step, interpretable reasoning sequences, thereby enabling effective problem-solving despite their smaller scale. Despite their reduced size, SRMs often exhibit distinct cognitive behaviors and processing dynamics compared to LRMs [9–11], suggesting that their training and inference strategies may differ fundamentally. Comparisons between LRMs and SRMs on mathematical and coding problems are illustrated in Figs. 2 and 3. Consequently, significant research efforts have been devoted to developing SRMs that rival or surpass LRMs in



**Fig. 1** A simple comparison between representative LRM s (GPT-4o and Claude-3.5-Sonnet) and SRMs (DeepSeek-R1-Distill-Qwen-7B) on various reasoning benchmarks, including mathematical reasoning benchmarks AIME2024 and MATH-500, the scientific question answering benchmark GPQA Diamond, and the coding benchmark LiveCodeBench

a variety of challenging reasoning tasks.

Through our survey of existing literature, we observe that while there exist several comprehensive surveys addressing the reasoning abilities of LLMs [2,12–17], reviews focusing explicitly on SRMs remain scarce. This highlights a critical gap in the literature for a focused overview that consolidates current advances in this emerging area. In this paper, we provide a concise yet thorough survey of SRMs, concentrating on research primarily published or publicly released within the past three years. Our goal is to integrate existing knowledge on key techniques, prominent applications, and promising future research directions related to SRMs. The overall structure and roadmap of our survey are depicted in Fig. 4, providing readers with a clear guide through the content presented.

#### • What is covered in this survey?

We begin by providing a quick overview of popular SRMs within the open-source community. Following this, we examine various training

and inference techniques designed to enhance the reasoning abilities of pre-trained models. Additionally, we survey domain-specific applications leveraging these models, discuss potential future research directions, and offer our own recommendations.

#### • What is NOT covered in this survey?

This survey does not address general model architecture designs or core algorithms applicable to LLMs at large, nor does it cover tasks unrelated to complex reasoning. Moreover, it does not focus on model compression techniques (such as pruning and quantization) and large-scale pre-training strategies aimed at producing smaller models, instead focusing specifically on techniques tailored for reasoning enhancement.

In summary, the exploration of SRMs surveyed in this paper represents a significant and timely research direction for the NLP community. By embracing the efficiency and unique capabilities of SRMs, researchers can accelerate the development of models that are not only high-performing but also sustainable and practical for real-world applications.

## ■ 2 A brief history of small reasoning models

Following the release of OpenAI’s o1 model, the AI community has witnessed a notable paradigm shift towards developing models endowed with strong reasoning capabilities. This shift has spurred increased interest in small reasoning models (SRMs) as efficient yet powerful alternatives to large reasoning models (LRMs). In this section, we review some of the most popular SRMs available in the open-source community that serve as valuable backbones for researchers pursuing further explorations in reasoning-driven NLP.

Prior to OpenAI’s o1, the focus was largely on task-specific SRMs tailored to particular domains, especially for code-related tasks such as code completion and natural language to code translation, given their extensive practical applications. More recently, the Qwen2.5-Coder [18] series has emerged, encompassing both LRM s and SRMs with parameter sizes ranging from 1.5B to 32B. Other notable SRM

**Problem:** How many two-digit numbers can be written in the form  $2^n$  where  $n$  is an integer?

**Thought (DeepSeek-R1):** To determine how many two-digit numbers can be written in the form of  $2^n$  where  $n$  is an integer, I need to explore the powers of 2 and identify which of them fall within the two-digit range.

First, I recognize that two-digit numbers range from 10 to 99. I will start by calculating the powers of 2 until the result exceeds 99. When  $n = 4$ ,  $2^4 = 16$ , which is a two-digit number.

For  $n = 5$ ,  $2^5 = 32$ , which is also a two-digit number.

Continuing this process:  $n = 6$ :  $2^6 = 64$ ;  $n = 7$ :  $2^7 = 128$ .

Upon reaching  $n = 7$ , the result is 128, which is a three-digit number and therefore outside the two-digit range. By listing all the two-digit powers of 2 obtained from these calculations, I can count them to determine the total number of such numbers. The valid two-digit powers of 2 are 16, 32, and 64, resulting in a total of three numbers.

**Thought (Qwen3-4B):** Okay, so I need to figure out how many two-digit numbers can be written in the form  $2^n$ , where  $n$  is an integer. Let me start by understanding the problem.

Atwo-digit number is any number from 10 up to 99. So, I need to find all the numbers in that range that are powers of 2. For example,  $2^0$  is 1, which is single-digit.  $2^1$  is 2, single-digit.  $2^2$  is 4, single-digit.  $2^3$  is 8, single-digit.  $2^4$  is 16, two-digit. Then  $2^5$  is 32, two-digit.  $2^6$  is 64, two-digit.  $2^7$  is 128, three-digit. So that’s three two-digit numbers. So the answer would be 3.

**Answer (both DeepSeek-R1 and Qwen3-4B):** The answer is 3.

**Fig. 2** Comparisons of how LRM s and SRMs solve mathematical reasoning problems. In this example, DeepSeek-R1 and Qwen3-4B serve as representative models of LRM s and SRMs, respectively. Although both provide correct answers, we observe that LRM s tend to employ more complex reasoning tools to solve the tasks, whereas the reasoning process of SRMs is generally more straightforward (by simple enumeration in this case)

**Problem:** You are climbing a staircase. It takes  $n$  steps to reach the top. Each time you can either climb one or two steps. In how many distinct ways can you climb to the top?

**Thought (DeepSeek-R1):** The problem is similar to the Fibonacci sequence. Because to get to step  $n$ , you could have come from either step  $n - 1$  (taking a 1-step) or step  $n - 2$  (taking a 2-step). So the total ways to  $n$  would be the sum of the ways to  $n - 1$  and  $n - 2$ . How do I implement this efficiently? The straightforward recursive approach would be too slow for large  $n$  because it recalculates the same steps multiple times. So recursion with memoization might be better, but maybe even better to use an iterative approach. Let me think about the iterative approach. Let's say we have two variables,  $a$  and  $b$ , representing the ways to reach step  $i - 2$  and  $i - 1$ . Then for each step from 2 to  $n$ , we can compute the current step's ways as  $a + b$ , then update  $a$  and  $b$  accordingly. What about edge cases? The problem states  $n$  is the number of steps to reach the top, so  $n$  is at least 1. So in code, we can handle  $n = 1$  and  $n = 2$  as base cases, then proceed.

**Answer (DeepSeek-R1):** The problem can be approached using dynamic programming. The key insight is that the number of ways to reach the  $n$ -th step is the sum of the ways to reach the  $(n - 1)$ -th step and the  $(n - 2)$ -th step. To optimize space complexity, we use an iterative approach with two variables to keep track of the previous two results, thus reducing the space complexity to  $O(1)$  while maintaining an  $O(n)$  time complexity.

**Thought (Qwen3-4B):** Okay, so I need to figure out how many distinct ways there are to climb a staircase with  $n$  steps, where each time I can take either one or two steps. Hmm, I remember this is a classic dynamic programming problem. Let me think. Let me start with small examples to see the pattern. If there's 1 step, there's only 1 way. If there are two steps, you can take 1 then 1, or 2. So that's two ways. For three steps, the possibilities are: 1+1+1, 1+2, 2+1. So that's three ways. Oh right, so the number of ways seems to follow the Fibonacci sequence. Like, for  $n=1, 1$ ;  $n=2, 2$ ;  $n=3, 3$ ;  $n=4, 5$ , etc. So the pattern is that each step's number of ways is the sum of the two previous steps.

So the recurrence relation is  $f(n) = f(n - 1) + f(n - 2)$ . With base cases  $f(1) = 1, f(2) = 2$ . So the problem is to compute  $f(n)$  using this recurrence. Now, how to implement this in Python. Let me think about the approach. For small  $n$ , we can use a simple recursive approach, but that's not efficient. But for larger  $n$ , recursion with memoization would be better. Alternatively, we can use dynamic programming with a bottom-up approach.

**Answer (Qwen3-4B):** To solve the problem of determining the number of distinct ways to climb a staircase with  $n$  steps, where you can take either 1 or 2 steps at a time, we can use dynamic programming. This is a classic Fibonacci-like sequence, where the number of ways to reach step  $n$  is the sum of the number of ways to reach step  $n - 1$  and step  $n - 2$ .

**Solution Code (both DeepSeek-R1 and Qwen3-4B):**

```
class Solution :
    def climbStairs (self , n: int ) -> int :
        if n == 1:
            return 1
        if n == 2:
            return 2
        a, b = 1, 2
        for i in range (3, n + 1):
            c = a + b
            a = b
            b = c
        return
```

**Fig. 3** Comparisons of how LRM<sub>s</sub> and SRM<sub>s</sub> solve coding reasoning problems. In this example, DeepSeek-R1 and Qwen3-4B serve as representative models of LRM<sub>s</sub> and SRM<sub>s</sub>, respectively. Although both generate correct and nearly identical Python code, we observe differences in their reasoning processes. LRM<sub>s</sub> tend to provide more detailed and logical explanations, often including space and time complexity analyses, whereas the reasoning process of SRM<sub>s</sub> is relatively simpler

series designed for code tasks include DeepSeek-Coder [19] and StarCoder2 [20], which demonstrate robust reasoning capabilities specially optimized for programming-related problems. For a more comprehensive overview of code-specific SRM<sub>s</sub>, readers are referred to [21]. Mathematics constitutes another challenging domain for SRM<sub>s</sub>, requiring intricate multi-step reasoning to address complex problems. Notable open-source SRM families in this domain include Qwen2.5-Math [22], DeepSeek-Math [23], and InternLM-Math [24], each pushing the boundary of mathematical problem solving through innovative model architectures and training strategies. Beyond code and mathematics, SRM<sub>s</sub> have also been developed for specialized domains such as healthcare, science, law, and finance. We refer readers to Section 5 for a detailed survey of domain-specific SRM<sub>s</sub> in these areas.

With the public release of powerful LRM<sub>s</sub> such as DeepSeek-R1 and QwQ-32B, which produce explicit long CoT trajectories as part of their outputs, there has been a corresponding rise in general-

purpose SRM<sub>s</sub> capable of tackling a broad spectrum of reasoning tasks. Many of these general-purpose SRM<sub>s</sub> have been developed using knowledge distillation techniques, wherein large reasoning models serve as teachers to transfer reasoning abilities to smaller, more efficient student models. Early explorations in the field of SRM<sub>s</sub> include models such as s1 [25], LLaMA-O1, and Marco-o1 [26], among others. In addition, distilled versions based on the LLaMA and Qwen series have been released by the DeepSeek AI team [3], including models such as DeepSeek-R1-Distill-LLaMA-8B and DeepSeek-R1-Distill-Qwen-7B. Other notable general-purpose SRM<sub>s</sub> include the OpenThinker series and the DistilQwen reasoning series [27].

These models demonstrate promising performance across a variety of reasoning benchmarks while maintaining relatively small parameter sizes, making them highly attractive for both academic research and practical deployment scenarios. In Table 1, we present a quantitative comparison of several recently released strong SRM<sub>s</sub>

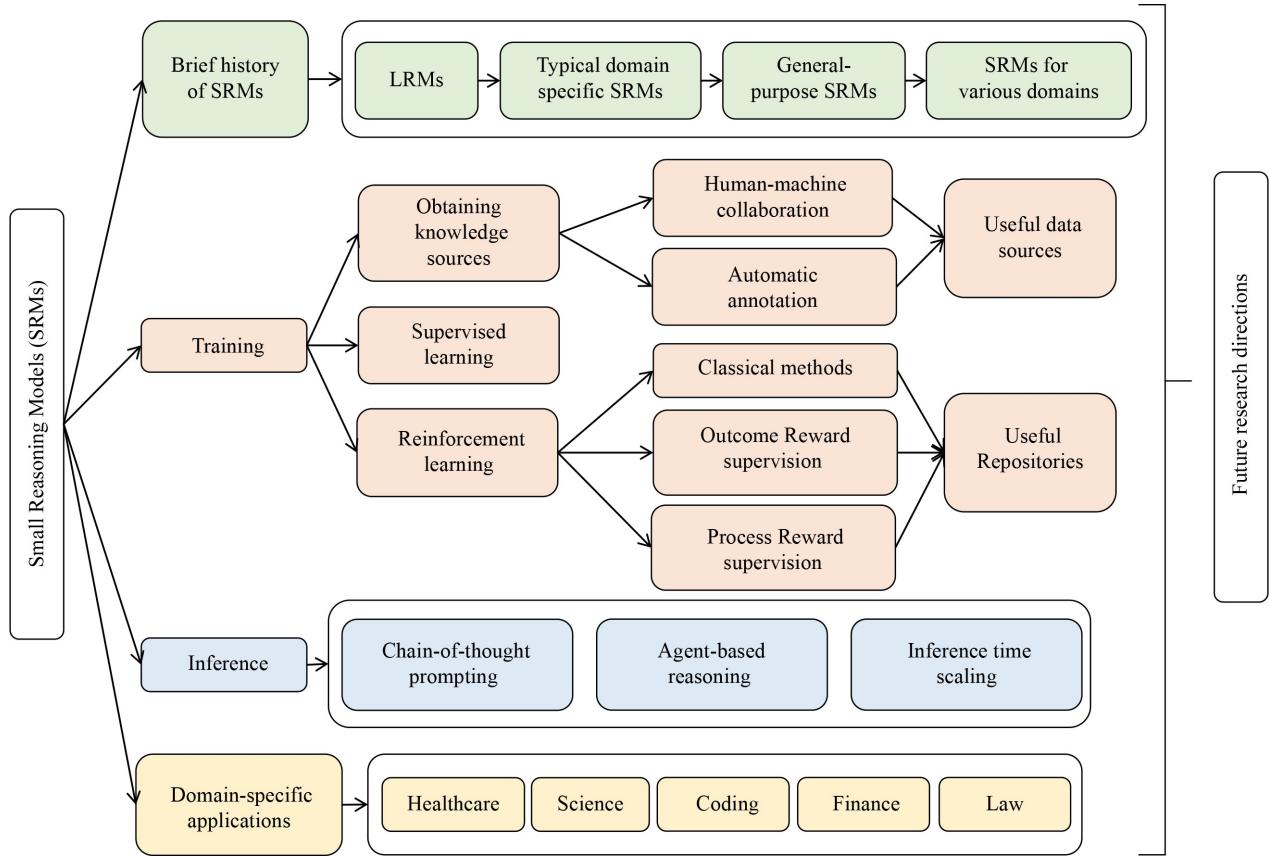


Fig. 4 The roadmap of this survey

**Table 1** Quantitative comparison among recently released strong SRMs

Model key	Parameter size/B	AIME2024	MATH-500	GPQA diamond	LiveCodeBench
deepseek-ai/DeepSeek-R1-Distill-1.5B	1.5	28.9	83.9	33.8	16.9
open-thoughts/OpenThinker3-1.5B	1.5	52.0	86.4	29.5	39.4
alibaba-pai/DistillQwen-ThoughtY-4B	4	76.7	95.2	56.1	75.8
bespokelabs/Bespoke-Stratos-7B	7	20.0	82.0	37.8	36.1
deepseek-ai/DeepSeek-R1-Distill-7B	7	57.3	89.6	47.3	48.4
alibaba-pai/DistilQwen-ThoughtX-7B	7	56.7	90.2	50.0	56.8
open-thoughts/OpenThinker-7B	7	31.3	83.0	42.4	39.9
open-thoughts/OpenThinker2-7B	7	50.0	88.4	49.3	55.6
open-thoughts/OpenThinker3-7B	7	69.0	90.0	53.7	51.7
deepseek-ai/DeepSeek-R1-Distill-Llama-8B	8	50.4	89.1	49.0	39.6
alibaba-pai/DistillQwen-ThoughtY-8B	8	76.7	94.6	62.1	78.1

available in the HuggingFace open-source community. The benchmarks used for evaluation include AIME2024 and MATH-500, which target mathematical reasoning; GPQA Diamond [28], a challenging scientific question-answering dataset; and LiveCodeBench [29], a coding benchmark. All reported scores are sourced directly from the respective HuggingFace model cards to ensure consistency.

In summary, the growing availability and advancement of SRMs

signify a significant step forward in the community's efforts to build efficient models with strong reasoning skills. By leveraging these models, researchers have access to a variety of versatile and cost-effective backbones that facilitate further innovation and experimentation in reasoning-centric NLP tasks. Consequently, SRMs not only complement existing large models but also promote sustainable and accessible research practices for the broader Artificial Intelligence (AI) community.

### ■ 3 Training SRMs without pain

In this section, we discuss effective training pipelines for producing high-quality small reasoning models (SRMs) with minimal manual cost and maximum efficiency. The takeaways for training SRMs are summarized in Fig. 5.

#### 3.1 Obtaining the knowledge sources

Creating high-quality datasets that contain detailed reasoning processes is crucial for training SRMs. Although human annotation typically guarantees high data quality, it is extremely costly and often impractical, especially when annotating entire long CoT processes [4] required for training SRMs. Annotators must guide or even compose each intermediate reasoning step [30], which is labor-intensive and time-consuming. In the following, we briefly discuss

human-model collaboration methods before focusing primarily on automatic annotation techniques, which offer scalable solutions.

To support future research efforts, we provide an ever-growing, although incomplete, list of popular open-source datasets featuring long CoT and output annotations generated by LRMAs, as shown in Table 2. These datasets are accessible via HuggingFace Datasets, offering a practical starting point for researchers aiming to train SRMs without costly data reproduction.

- Human-model collaboration

Human-model collaboration approaches leverage large language models or LRMAs to perform initial annotations by utilizing a few carefully selected high-quality examples as in-context demonstrations. Human annotators then intervene only to correct the low-quality or erroneous annotations, which typically constitute a

#### Takeaways (Training)

Training SRMs can be streamlined through a combination of effective data annotation and advanced training algorithms. Leveraging automatic annotation and knowledge distillation techniques has become increasingly important with the availability of strong teacher models. Chain-of-thought (CoT) based Supervised Fine-Tuning (SFT) provides a solid foundation for training SRMs on collected and annotated datasets. Moreover, advanced Reinforcement Learning (RL) methods enable more effective training by evaluating intermediate reasoning steps while significantly reducing computational costs. Although there is still no widely accepted consensus on the single best RL algorithm for SRMs, the success story of DeepSeek-R1 favors outcome reward supervision [3], viewing Process Reward Models (PRMs) and Monte Carlo Tree Search (MCTS) as less successful attempts so far. However, we argue that it is premature to draw definitive conclusions, especially given the distinct nature of SRMs and their training challenges. Researchers can accelerate progress by utilizing publicly available datasets and RL libraries designed to facilitate SRM development.

**Fig. 5** Takeaways for training SRMs

**Table 2** The list of open-source datasets with long CoT and output annotations by LRMAs

Dataset key	Size	Annotator(s)	Related paper
FreedomIntelligence/medical-o1-reasoning-SFT	25.4K	GPT-4o	[31]
FreedomIntelligence/Medical-R1-Distill-Data	22K	DeepSeek-R1	[31]
FreedomIntelligence/Medical-R1-Distill-Data-Chinese	17K	DeepSeek-R1	[31]
facebook/natural_reasoning	2.8M	Llama3.3-70B-Instruct	[32]
Congliu/Chinese-DeepSeek-R1-Distill-data-110k	110K	DeepSeek-R1	–
open-r1/codeforces-cots	47.8K	DeepSeek-R1	–
open-r1/OpenR1-Math-220k	220K	DeepSeek-R1	–
SmallDoge/SmallThoughts	50K	DeepSeek-R1	–
GeneralReasoning/GeneralThought-323K	323K	Multiple	–
GeneralReasoning/GeneralThought-195K	195K	Multiple	–
open-thoughts/OpenThoughts-114k	114K	DeepSeek-R1	–
open-thoughts/OpenThoughts2-1M	1M	DeepSeek-R1	–
open-thoughts/OpenThoughts3-1.2M	1.2M	QwQ-32B	–
alibaba-pai/OmniThought	2M	Multiple	[27]
alibaba-pai/OmniThought-0528	365K	DeepSeek-R1-0528	[27]
bespokelabs/Bespoke-Stratos-17k	17K	DeepSeek-R1	–
simplescaling/s1K-1.1	1K	DeepSeek-R1	[25]
EricLu/SCP-116K	116K	o1-mini, QwQ-32B-Preview	[33]

small fraction of the entire dataset [34–38]. This method balances efficiency and quality effectively by significantly reducing human involvement while maintaining dataset reliability.

#### • Automatic annotation

Early automatic annotation efforts primarily targeted complex tasks without generating long reasoning trajectories, such as tool usage and Application Programming Interface (API) calls [39–41]. However, as annotators' reasoning capabilities improve, advanced LRMAs are now able to generate lengthy CoT trajectories in a zero-shot or few-shot manner [42]. For instance, leveraging DeepSeek-R1 for automatic annotation is an efficient solution since it can generate extended CoT sequences within large context windows [3]. This annotation paradigm, often referred to as the “knowledge distillation”, has been widely used for training various types of smaller language models [43–49]. Notably, recent work [50] demonstrates that SRMs can be trained effectively purely on synthetic datasets generated by LRMAs, without reliance on human-annotated data.

Another emerging trend is the use of specialized agents that generate training data through complex operations such as planning, tool use, reflection, and iterative refinement. In these setups, LRMAs not only annotate the training data but also provide explicit documentation of the decision-making processes underlying their annotations [51–55]. These interaction steps themselves can be regarded as explicit reasoning processes, enriching the training data with valuable intermediate supervision signals.

In addition, since long reasoning trajectories inherently consist of multiple reasoning steps, annotating the correctness of each intermediate step provides fine-grained feedback that facilitates more effective learning for SRMs. Although these annotation tasks are more demanding, they are particularly beneficial for training Process Reward Models (PRMs) [56]. An early example [57] annotates the correctness of each step in mathematical problem solving. More recent studies employ Monte Carlo sampling techniques to evaluate intermediate reasoning steps based on the average outcomes of inference trajectories starting from those steps [58,59]. Extensions of these methods leverage Monte Carlo Tree Search (MCTS) and its variants, using tree search strategies to improve inference quality and thereby generate higher-quality annotations [60,61]. Taken together, these techniques suggest that inference-time scaling approaches, such as sampling and tree search, can be effectively leveraged for systematic dataset curation to train stronger SRMs.

A summary of these methods is presented in Table 3.

**Table 3** Comparison of methods for knowledge sources acquisition

Method	Description	Pros (Advantages)	Cons (Challenges)
Human Annotation	Manual annotation of detailed reasoning steps in datasets.	High data quality	Expensive, slow, impractical for long CoT annotations
Human-Model Collaboration	LRMs generate initial annotations via few-shot prompting; humans correct errors.	Balances quality and efficiency	Still requires substantial human correction effort
One-Pass Automatic Annotation	Fully automated generation of CoT reasoning by LRMs.	Highly scalable and efficient; enables training on synthetic data	May need advanced techniques for high data quality
Stepwise Fine-Grained Annotation	Annotate correctness of each intermediate reasoning step using sampling or MCTS for finer supervision.	Provides detailed, stepwise feedback	More demanding annotation task

## 3.2 Supervised learning

Supervised Fine-Tuning (SFT) is a widely adopted supervised learning technique to align models with task-specific instructions and objectives. With the growing availability of high-quality reasoning datasets, it has become natural to extend traditional SFT approaches to a CoT-based fine-tuning paradigm. In this paradigm, SRMs are trained not only to produce final answers but also to explicitly generate intermediate reasoning steps in response to given instructions. This explicit modeling of intermediate steps significantly enhances the models' reasoning capabilities, enabling them to handle complex multi-step tasks more effectively and transparently. Formally, given a training dataset  $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$ , where each input  $x^{(i)}$  is paired with an output sequence  $y^{(i)}$  consisting of CoT reasoning steps followed by a final answer (distilled from teacher models), the objective is to train a model to generate the entire CoT and the answer. Let the model parameters be denoted by  $\theta$ , and the model's probability of generating a token  $y_t^{(i)}$  conditioned on the input  $x^{(i)}$  and the previously generated tokens  $y_{<t}^{(i)}$  be  $p_\theta(y_t^{(i)}|x^{(i)}, y_{<t}^{(i)})$ . The loss function for a single training example is then given by the negative log-likelihood of the target sequence:

$$\mathcal{L}_{\text{SFT}}(\theta) = - \sum_{i=1}^N \sum_{t=1}^T \log p_\theta(y_t^{(i)}|x^{(i)}, y_{<t}^{(i)}), \quad (1)$$

where  $T$  is the length of the full output sequence. Prior work [3] has demonstrated that combining CoT-based SFT with knowledge distillation techniques results in highly capable SRMs that outperform models trained solely on final-answer supervision.

Nonetheless, SFT suffers from several inherent limitations. First, it heavily depends on the availability of large-scale, high-quality labeled datasets, whose creation is often costly, labor-intensive, and time-consuming. For instance, the distilled DeepSeek-R1 models were trained on datasets comprising up to 800K samples, highlighting the considerable data scale required for effective supervision and generalization. Second, fine-tuning SRMs is computationally expensive compared with general-purpose language models, especially when dealing with long input sequences containing detailed CoT trajectories. This computational burden has motivated the community to adopt Parameter-Efficient Fine-Tuning (PEFT) strategies [62,63] such as LoRA [64], QLoRA [65], AdaLoRA [66], among others, which introduce low-dimensional trainable parameters  $\Delta\theta$  that adapt the base model without modifying the full parameter set. In PEFT, let  $\theta_0$  denote the base model

parameters and  $\theta = \theta_0 + \Delta\theta$  be the adapted parameters after PEFT. The training objective remains to minimize the negative log-likelihood loss over the training dataset  $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$ :

$$\mathcal{L}_{\text{PEFT}}(\Delta\theta) = - \sum_{i=1}^N \sum_{t=1}^T \log p_{\theta_0 + \Delta\theta}(y_t^{(i)} | x^{(i)}, y_{<t}^{(i)}), \quad (2)$$

where only  $\Delta\theta$  is optimized, which typically has much lower dimensionality, significantly reducing memory footprint and computational cost.

Another notable drawback of SFT is its tendency to encourage SRMs to “exploit” rather than “explore” the training distribution, often leading to models that overfit the training data and fail to generalize to novel problem instances or broader solution spaces. This limitation has spurred increased interest in Reinforcement learning (RL)-based approaches, which promote exploration through reward-driven interactions and can potentially yield more robust and generalizable reasoning abilities. By balancing exploitation and exploration, RL-based methods hold promise for enhancing the reasoning skills of SRMs beyond the capabilities attainable by SFT alone.

### 3.2.1 Reinforcement learning (RL)

RL offers a powerful alternative training approach for SRMs by enabling models to learn optimal strategies through trial and error, thus improving their generalization abilities beyond simply mimicking gold-standard responses during SFT. Influential public repositories that support RL training for SRMs include verl [67], ReaLHF [68], AReaL [69], Light-R1 [70], Open-Reasoner-Zero [70], OpenRLHF [71], and many others.

#### 1) Starting with classical RL methods

The work [72] has popularized Reinforcement Learning from Human Feedback (RLHF) by employing Proximal Policy Optimization (PPO) [73] to effectively align LLMs with human preferences. Given a policy (model) parameterized by  $\theta$ , PPO aims to maximize the expected reward while constraining policy updates to prevent large deviations. Let  $\pi_\theta(a | s)$  denote the policy probability of taking action  $a$  in state  $s$ , and let  $\pi_{\theta_{\text{old}}}$  be the policy before the update. The PPO objective with clipping is defined as:

$$\mathcal{L}_{\text{PPO}}(\theta) = \mathbb{E}_{(s,a) \sim \pi_{\theta_{\text{old}}}} \left[ \min \left( r_\theta(s, a) \hat{A}(s, a), \right. \right. \\ \left. \left. \text{clip} \left( r_\theta(s, a), 1 - \epsilon, 1 + \epsilon \right) \hat{A}(s, a) \right) \right], \quad (3)$$

where

$$r_\theta(s, a) = \frac{\pi_\theta(a | s)}{\pi_{\theta_{\text{old}}}(a | s)}, \quad (4)$$

is the probability ratio between the new and old policies,  $\hat{A}(s, a)$  is an estimator of the advantage function (measuring how much better action  $a$  is compared to the average), and  $\epsilon > 0$  is a hyperparameter controlling the clipping range to ensure stable updates. In the RLHF context, the state  $s$  corresponds to the input prompt and dialogue/history so far, the action  $a$  corresponds to the next token generated by the model, and the reward is derived from a reward model trained on human preference data. This approach has served as

a foundational method for improving model behavior in a human-centric manner.

To further reduce the reliance on costly and time-consuming manual annotation, recent paradigms like Reinforcement Learning from AI Feedback (RLAF) [74] leverage model-generated labels to train reward models. This strategy substantially mitigates the burden of human labeling while maintaining strong alignment signals. Importantly, these foundational advances in RL training for language models translate seamlessly to SRMs without requiring significant modification, making them broadly applicable in this domain.

To simplify the training process, Direct Preference Optimization (DPO) [75] introduces a more straightforward alternative to traditional reward model optimization by using a margin-based loss function that aligns models more directly with preference data. Consider a dataset of  $N$  preference tuples  $\mathcal{D} = \{(x^{(i)}, y_+^{(i)}, y_-^{(i)})\}_{i=1}^N$ , where for input  $x^{(i)}$ ,  $y_+^{(i)}$  is the preferred output and  $y_-^{(i)}$  is the less preferred output, both including CoTs and final answers. The DPO loss for the  $i$ th example is defined as:

$$\mathcal{L}_{\text{DPO}}(\theta, i) = -\log \sigma \left( \log p_\theta(y_+^{(i)} | x^{(i)}) - \log p_\theta(y_-^{(i)} | x^{(i)}) - \beta \right), \quad (5)$$

where  $\sigma(\cdot)$  is the sigmoid function and  $\beta > 0$  is a margin hyperparameter that controls the strength of preference enforcement. The overall loss is given by the average over all preference pairs:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{DPO}}(\theta, i). \quad (6)$$

This approach reduces the complexity and instability often associated with reward modeling and reinforcement learning. Building upon DPO, a series of extensions such as KTO [76], ODPO [77], and SimPO [78] have been developed, which are also highly suited for autoregressive SRMs due to their efficient optimization properties. However, despite their advantages, these methods are not explicitly tailored or optimized for the unique challenges posed by the lengthy CoT sequences typically generated by SRMs. This gap highlights an important area for future research aimed at designing RL-based techniques that can more effectively handle and leverage long, structured reasoning trajectories.

2) Enhancing multi-step reasoning with outcome reward supervision  
A straightforward RL approach leverages the final outcome of a reasoning process as the sole reward signal, while disregarding the potentially informative intermediate steps. Reinforced Fine-Tuning (ReFT) [79] exemplifies this strategy by first warming up the model using supervised fine-tuning, followed by an online RL stage utilizing PPO. During training, ReFT samples multiple reasoning trajectories per question and assigns rewards based on the correctness of the final answer, effectively guiding the model to improve its ultimate predictions.

Building upon this, VinePPO [80] tackles a critical issue of bias in PPO’s value network concerning intermediate reasoning steps. By employing Monte Carlo sampling, VinePPO provides unbiased value estimates, which helps to better stabilize training and enhance performance on multi-step reasoning tasks. Similarly, Critical Plan Step Learning (CPL) [81] leverages MCTS to systematically explore

and evaluate planning steps within multi-step reasoning processes. CPL iteratively optimizes both policy and value models by incorporating intermediate-step evaluations, leading to substantial improvements in reasoning accuracy and robustness.

Recently, Group Relative Policy Optimization (GRPO) [23] offers an alternative by completely eliminating the need for a separate critic model. Instead, it estimates rewards by comparing a group of inference outputs generated for the same prompt, thereby significantly reducing computational overhead. GRPO has been successfully applied to train powerful SRMs such as DeepSeekMath-7B, as well as ultra-large LRMAs like DeepSeek-R1 [3].<sup>1)</sup>

### 3) Fine-grained RL with process reward supervision

A notable limitation of relying solely on outcome-based rewards is the sparsity of feedback signals, which can slow down or hinder effective learning. To address this, PRMs [56] provide granular, step-level feedback by evaluating the quality of intermediate reasoning steps throughout the CoT trajectories generated by SRMs. This richer form of supervision, often referred to as process reward supervision, enables more informative guidance at every stage of the reasoning process. For example, Math-Shepherd [58] employs step-by-step verification coupled with reinforcement learning to enhance mathematical reasoning performance by leveraging PRMs to assess and reinforce intermediate correctness. Similarly, Self-Explore [82] utilizes PRMs to identify and correct “first pits”, i.e., early mistakes in problem-solving chains, rewarding corrective steps. This approach facilitates self-supervised fine-tuning, dramatically reducing the dependency on extensive human annotations.

Process Advantage Verifiers (PAVs) [83] focus on evaluating incremental progress at the step level to selectively encourage improvements in solution correctness during RL training. Beyond online RL methods, several off-policy techniques inspired by DPO [75] have integrated process reward supervision. For instance, SVPO [84] employs MCTS to explore diverse reasoning paths and annotate corresponding step-level preferences, enabling more fine-grained preference-based optimization. Additional search-based process supervision strategies have been explored in recent works [85–87], collectively pushing the frontier toward more effective and interpretable multi-step reasoning frameworks.

### 3.3 Comparison between SFT and RL

SFT and RL represent two complementary paradigms for training

SRMs, each with distinct strengths and intrinsic limitations. A summary between SFT and various RL algorithms is presented in Table 4.

Firstly, SFT excels at leveraging large-scale labeled datasets to directly teach models to imitate high-quality reasoning paths, particularly effective when rich CoT annotations are available. By optimizing the likelihood of expert-generated sequences, SFT enables efficient and stable convergence, often producing models with strong initial capabilities. However, SFT heavily relies on expensive, labor-intensive data collection and is prone to overfitting the observed training distribution. This can limit a model’s capacity to explore alternative reasoning strategies and generalize to novel or more challenging tasks.

In contrast, RL introduces a flexible framework where SRMs learn through reward-driven trial and error, enabling better balance between exploitation of learned behaviors and exploration of novel solution paths. Moreover, RL can exploit outcome-based or fine-grained process rewards, providing richer supervision that is often unavailable in SFT. Nevertheless, RL training is typically more complex, computationally intensive, and less stable, particularly when handling the long, structured CoT sequences.

To empirically compare SFT and RL training paradigms for SRMs, we conduct experiments across multiple benchmarks using Qwen2.5-7B-Instruct as the backbone model. We train models using SFT with CoT annotations from the Bespoke-Stratos-17k dataset and the OpenThoughts-114k dataset on HuggingFace datasets, respectively. Additionally, we test direct GRPO training on the model backbone without any CoT-based SFT, which results in much poorer performance and failure to achieve convergence. We also assess hybrid training pipelines that combine SFT initialization followed by GRPO-based RL refinement, using a randomly sampled subset of 1,000 problems from the same training set. All experiments are implemented in PyTorch and conducted on a server with eight A800 GPUs (80GB). The experimental results and their respective training time, summarized in Table 5, provide empirical insights into the trade-offs between direct imitation and reward-driven exploration, demonstrating that combining SFT initialization with subsequent RL refinement yields the best overall performance.

## ■ 4 Boosting SRM inference with scale

Complex reasoning tasks often require multi-step computations and sophisticated inference strategies. In this section, we explore key

**Table 4** Summary and comparison of SFT vs. RL methods for SRMs

Category	Representative methods	Supervision signal	Pros	Cons
SFT	CoT-based SFT	Gold CoT annotations	Strong alignment; explicit chain-of-thought	Data-hungry; expensive compute for long CoT
RL (Classical)	RLHF/RLAIF (PPO), KTO, ODPO, SimPO	Human preference	Learns preferences; fewer manual labels	Not tailored for long CoT; still needs reward models
RL (Outcome Reward)	ReFT, VinePPO, CPL, GRPO	Final - answer correctness	Encourages exploration; GRPO removes critic	Sparse feedback; ignores intermediate steps
RL (Process Reward)	Self-Explore, PAVs, SVPO, GRPO	Step - level (process) rewards	Dense, fine-grained supervision; early error correction	Requires PRMs; higher computational cost

<sup>1)</sup> GRPO is applicable to both outcome supervision and process supervision settings, providing flexibility across different training paradigms.

**Table 5** Empirical analysis of how the reasoning abilities of SRMs progressively improve through SFT and RL

Training setting	AIME2024	MATH-500	GPQA Diamond	LiveCodeBench	Training time/h
Zero-shot learning	10.0	73.6	33.3	30.7	—
Training Set: Bespoke-Stratos-17k					
SFT	20.0	80.0	37.4	31.3	0.5
<b>SFT+GRPO</b>	<b>26.7</b>	<b>84.4</b>	<b>40.9</b>	<b>36.0</b>	0.5+0.5
Training Set: OpenThoughts-114k					
SFT	31.3	83.0	42.4	39.9	3
<b>SFT+GRPO</b>	<b>43.3</b>	<b>86.0</b>	<b>42.9</b>	<b>44.2</b>	3+0.5

approaches for effectively scaling the reasoning capabilities of SRMs during inference. The takeaways are summarized in Fig. 6.

#### 4.1 Chain-of-thought (CoT) prompting

Chain-of-thought prompting, an influential extension of few-shot prompting [88], has demonstrated broad applicability well beyond classical algorithmic and structured reasoning tasks. This approach initially emphasized the explicit generation of intermediate reasoning steps as a way to improve interpretability and performance [89,90]. Though simple random sampling of reasoning trajectories is computationally straightforward, it often proves inefficient and suboptimal because it distributes limited test-time computational budgets across many less promising branches [91,92]. To overcome these challenges, recent research has concentrated on mechanisms for prioritizing the most promising reasoning paths or intermediate steps, effectively narrowing the search space to enhance both efficiency and solution quality [93–95]. CoT-SC [93] extends conventional CoT prompting by adopting a tree-structured framework, where multiple chain-of-thought branches are expanded from the same initial prompt in parallel. Among these branches, the one that leads to the best performance or most accurate final answer is selected, thus effectively leveraging exploration while managing computational costs. Similarly, SoT [96] innovates by directing SRMs to first generate an answer skeleton. This skeleton provides a high-level scaffold, which is then efficiently completed via parallel API calls or batched decoding to fill in detailed points, thereby speeding up inference without sacrificing reasoning quality. More recently, numerous works have explored Tree of Thoughts [94,97], which employ sophisticated tree search strategies to decompose complex questions into smaller, manageable sub-questions. Each sub-question is then addressed through distinct prompts, significantly enhancing the diversity and depth of reasoning processes and enabling models

to tackle more challenging problems with greater accuracy. As shown in Fig. 7, we evaluate the performance of CoT and ToT prompting on challenging reasoning benchmarks. The results show that ToT outperforms CoT, especially in mathematical reasoning, due to its ability to decompose complex problems into simpler subproblems. However, it is worth noting that ToT often results in longer reasoning processes than CoT, typically requiring 3 to 5 times more inference steps in our experiments.

#### 4.2 Agent-based reasoning

Agent-based reasoning methods can be broadly classified into two categories: agent collaboration for managing distinct roles, and the design or learning of agent communication graphs.

- Agent collaboration

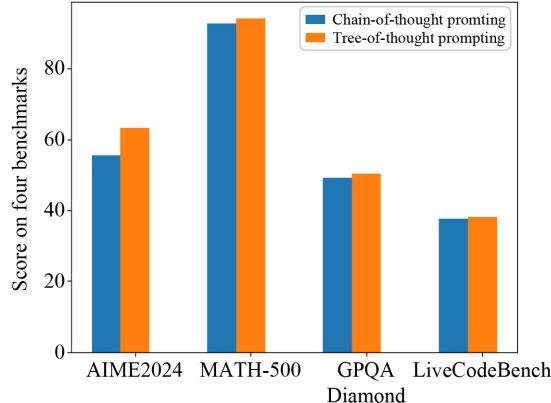
Collaborative communication among multiple agents has recently emerged as a highly effective mechanism to enhance the reasoning performance and robustness of individual SRMs [98–100]. These approaches typically revolve around two fundamental types of communication paradigms: intra-flow and inter-flow communication.

Intra-flow communication refers to the message exchanges occurring among agents within a single conversation round or interaction step. Several common communication topologies have been proposed and studied: 1) Immediate output, where agents operate independently without direct communication and produce responses solely based on their own internal reasoning capabilities [98,101]; 2) Chain-style connection, which establishes a sequential communication flow in which messages or information are propagated step-by-step from one agent to the next in a linear pipeline [102–104]; 3) Tree-style connection, where communication follows a hierarchical structure guided by a supervisory root or manager agent that directs and coordinates subordinate agents in a

#### Takeaways (Inference)

To boost the reasoning capabilities of SRMs during inference, a diverse array of strategies has been employed. Chain-of-Thought (CoT) prompting has become a widely adopted technique to guide SRMs in generating intermediate reasoning steps, substantially enhancing their performance on complex reasoning tasks. Agent-based reasoning methods further advance SRMs' capabilities by leveraging structured interactions among multiple agents, fostering collaborative problem solving. Additionally, inference-time scaling techniques—including self-enhanced tree search and step-wise verifiers—have been developed to optimize computational resources and improve reasoning efficiency during inference.

**Fig. 6** Takeaways for boosting the reasoning capabilities of SRMs during inference



**Fig. 7** A simple comparison between CoT and ToT prompting based on DeepSeek-R1-Distill-Qwen-7B on various reasoning benchmarks, including mathematical reasoning benchmarks AIME2024 and MATH-500, the scientific question answering benchmark GPQA Diamond, and the coding benchmark LiveCodeBench

multi-level fashion [105–107]; 4) Graph-style connection, which models agents as nodes in a flexible graph topology where information can flow dynamically along edges, allowing complex interaction patterns and richer information exchange [108–110]. 5) Sequential-style connection, These works introduce a novel framework that reconceptualizes multi-agent corporation using a sequential approach instead of a graph-based structure, dynamically choosing the optimal agent role at every step and allowing agents to selectively retrieve pertinent information from prior steps [111,112].

Inter-flow communication, on the other hand, focuses on the propagation and transformation of information across successive rounds or iterations of agent interactions. Typical inter-flow patterns include: 1) Full connection, where each agent receives all utterances or messages generated by every other agent in the previous interaction round, facilitating comprehensive information sharing [98]; 2) Partial connection, involving selective filtering or scoring mechanisms that evaluate and rank agents' outputs, allowing only the most relevant or highest-quality responses to be propagated forward [113,114]; 3) Summarization, where dialogues or communication history from prior rounds are compressed into concise summaries, enabling more efficient and scalable communication among agents in subsequent rounds without sacrificing important contextual information [115–117].

#### • Agent graphs

Enhancing agent cooperation through learned or pre-designed graph connectivity has long been recognized as an effective strategy in multi-agent systems. Prior to the emergence of LLM-powered agents, significant research focused on optimizing communication topologies leveraging advanced techniques such as graph diffusion [118], weighted graph neural networks [119], and transformer-based architectures [120]. These approaches aim to optimize the flow of information and coordination among agents, facilitating more coherent and effective group behavior.

With the advent of LLM-powered agents, new frameworks have begun to implicitly or explicitly employ graph structures to represent

complex interaction patterns during simultaneous communications. For instance, ChatEval [121] and AutoGen [105] utilize graph-like representations to manage multi-agent conversations, enabling coordinated reasoning in a distributed manner. Similarly, STOP [122] and DSPy [123] jointly optimize both prompt design and inference structure, effectively shaping the underlying communication graph to improve agent collaboration. More explicitly, MacNet [124] and GPTSwarm [109] model agent interactions using directed acyclic graphs, which facilitate structured, hierarchical information flow and reasoning processes.

Distinguishing these modern developments from classical approaches, CDC [118] focuses on dynamically modifying communication graphs via diffusion processes, adapting connectivity patterns in an online fashion. TWG-Q [119] emphasizes temporal weight learning within weighted graph convolutional networks to capture evolving agent relations over time. In contrast, CommFormer [120] adopts a novel paradigm by learning a static communication graph prior to inference, trading off adaptivity for enhanced efficiency and scalability. This pre-learned graph serves as a powerful prior for agent collaboration, setting it apart from both traditional dynamic graph methods and more recent LLM-driven interaction models. Overall, these varied strategies illustrate the rich landscape of agent graph modeling, offering diverse tools to optimize multi-agent communication and reasoning in increasingly complex environments.

#### 4.3 Inference time scaling

Optimizing the allocation of computational resources during inference is a critical avenue for achieving substantial efficiency gains [91]. Self-enhanced tree search methods [125,126] effectively integrate multiple reasoning trajectories by leveraging sparse activation mechanisms, allowing for more efficient execution without sacrificing performance. Complementing this, step-wise verifiers dynamically prune the search space by filtering out less promising paths early in the inference process [30,127], significantly reducing unnecessary computation. Along similar lines, two-stage elimination techniques employing pairwise comparisons iteratively refine candidate solutions to enhance inference quality and robustness [128]. Moreover, iterative refinement approaches [129–132] have demonstrated noteworthy success in solving complex tasks by progressively improving model outputs through multiple passes. The S1 method [25] further introduces a simple yet effective test-time scaling strategy, imposing inference length constraints to more judiciously utilize computational resources.

Differentiating themselves from naive repeated sampling, scaling-based inference methods empower models to iteratively generate and refine solution candidates conditioned on prior attempts [133,134]. Prominent algorithms such as MCTS [135–137] and guided beam search [138] effectively unify aspects of both sequential and parallel inference scaling [25], utilizing tree-based search strategies to explore solution spaces efficiently [92,139]. Among these, REBASE [92] stands out by introducing an innovative process reward model that adeptly balances exploration and exploitation through intelligent pruning during tree search. Empirically, REBASE consistently

outperforms both standard sampling-based methods and traditional MCTS, establishing a new state-of-the-art in inference efficiency.

Reward models occupy a central role in guiding these inference-time scaling techniques and broadly fall into four key categories: outcome-based, process-based, endogenous-based and multi-object reward models. Outcome reward models [140,141] primarily assess the quality of final answers using scoring functions, playing a crucial role in Best-of-N selection frameworks where multiple candidate responses are generated and the best is chosen. In contrast, process reward models [58,92,140] provide richer supervision by evaluating intermediate reasoning steps. This fine-grained guidance proves especially valuable for navigating tree-based search spaces and supporting iterative inference, as it enables the selective pruning of unpromising paths and fosters more targeted refinement of partial solutions. The endogenous-based rewards generated by the model itself based on internal states or learning processes rather than directly provided by the external environment. A representative method is curiosity-driven rewarding models that it predicts environmental dynamics with errors via predicting the next state [142]. Multi-objective rewards are often aimed at task objectives that are not single. The reward function is designed as a weighted sum of multiple sub-objectives or a more complex combination [143,144]. Together, these reward model families underpin the success of modern inference scaling methods, driving both improved efficiency and enhanced reasoning quality.

## ■ 5 Domain-specific applications

While LRM demand broad knowledge, domain-specific SRMs emphasize deep expertise within targeted fields. This section highlights prominent SRM applications across diverse domains. The takeaways are summarized in Fig. 8.

### 5.1 Healthcare

Hippocrates [149] provides open access to datasets, codebases, models, and training protocols. It is trained on a comprehensive medical corpus including Medical Guidelines, PMCPatients [156], and PubMedQA-contexts [157], totaling approximately 300 million tokens. The Hippo series follows a training pipeline of continual pre-training, SFT, and RLHF. Fine-tuned versions of Mistral and LLaMA-2 compete robustly against several 70B parameter models; for instance, Hippo-Mistral-7B attains 59.9% accuracy on MedQA, exceeding Meditron-70B's 58.5% [158].

BioMedLM [159] is a 2.7B parameter model pre-trained on

PubMed data [160]. AdaLM [151] advances domain-specific modeling via continued training focused on medical data, demonstrating that an adaptation-then-distillation strategy yields superior results. MentalLLaMA [146] pioneers two important contributions: 1) the first IMHI dataset for mental health analysis, and 2) the first open-source model enabling explainable analyses of social media content related to mental health.

HuatuoGPT-o1 [31] introduces verifiable medical problems alongside a domain-specific medical verifier that assesses model outputs' accuracy. This verifiability supports a two-step approach for advancing medical reasoning: 1) employing the verifier to guide the search for complex reasoning paths, and 2) leveraging reinforcement learning with verifier-based reward signals to iteratively improve the model's capacity to handle intricate medical reasoning scenarios.

### 5.2 Science

SciGLM [147] is a scientific SRM developed to overcome data scarcity via a self-reflective instruction annotation framework. Utilizing GPT-4 [161], it generates step-by-step reasoning for unlabeled scientific questions through a three-stage process with structured prompts: CoT prompting to elicit stepwise reasoning, reflective prompting to detect and correct errors, and answer integration to consolidate corrected solutions for accurate outputs.

Llemma [162], adapted from CodeLlama [163], focuses on advanced mathematical reasoning. Through extended pre-training, its 7B parameter model is fine-tuned on 55 billion tokens from the newly constructed Proof-Pile-2 dataset, which encompasses scientific publications, mathematical Web content, and computational math resources. Llemma consistently outperforms similarly sized models on benchmarks such as MATH [164], GSM8k [165], OCWCourses [166], MMLU-STEM [164], and SAT.

ChemLLM [148] is a chemistry-focused SRM employing its proprietary ChemData framework, which reformats chemical knowledge into conversational question answering pairs. Built atop InternLM2-Base-7B [167], ChemLLM first strengthens its core competencies via pre-training on 1.7M question answering pairs from HuggingFace's multi-domain corpus. Subsequent SFT incorporates both ChemData and the multi-domain corpus to preserve generalization while specializing in chemistry. ChemLLM achieves remarkable performance in interdisciplinary chemistry tasks, rivaling GPT-4 [161] across multiple domains and consistently surpassing GPT-3.5 [72]. Notably, it attains a 92.6 score on Mol2caption, nearly

#### Takeaways (Domain-Specific Applications)

The standard approach for developing domain-specific SRMs involves adapting general models to specialized datasets. Consequently, many researchers construct custom datasets [145, 146, 147, 148], typically annotated using advanced LLMs (e.g., GPT-4) and subsequently employed for continual pre-training or fine-tuning of models such as LLaMA-2-7B [149]. To ensure high-quality outputs, specialized annotation frameworks—like the one used by SciGLM [147]—have been developed. For domains with substantial textual corpora, an effective strategy includes training a base model from scratch, followed by SFT [150]. Another promising avenue involves a dual process of distilling general capabilities from LLMs while systematically integrating specialized knowledge from domain-specific corpora [151]. However, knowledge transfer to smaller SRMs often introduces domain-specific biases and hallucinations, necessitating tailored debiasing methods to enhance robustness and fairness [152, 153]. Incorporating comprehensive bias auditing and ethical safeguards throughout development is crucial to mitigate inherited vulnerabilities and ensure responsible deployment [154, 155].

**Fig. 8** Takeaways for developing SRMs for domain-specific applications

matching GPT-4's level.

AstroLLaMA [145] targets astronomy applications. Based on LLaMA-2-7B [168], it undergoes extended pre-training on over 300K astronomy abstracts from arXiv. AstroLLaMA supports diverse astronomy tasks such as automated paper summarization and conversational assistance for research.

### 5.3 Coding

The use of SRMs for coding presents a viable alternative to LLMs, owing to their lower computational demands and potential for domain-specific optimization. While LLMs excel in code generation and programming assistance, SRMs offer advantages such as faster inference, lower operational costs, and better suitability for real-time applications where quick responses are essential. Key representative works are discussed next. For example, Phi-1 [169] is a 1.3-billion-parameter Transformer model that focuses on basic Python programming and achieves strong performance on benchmarks like HumanEval [169], which comprises 164 coding challenges. Later iterations, such as Phi-1.5 and Phi-2, further improve these abilities, while Phi-3 highlighted SRMs' capacity to compete with larger models [170]. The newest release, Phi-3.5-mini, with 3.8 billion parameters, leverages advanced fine-tuning and optimization methods to excel in long-context tasks, matching the performance of larger models like Llama-3.1-8B-instruct [171] and outperforming smaller ones such as Gemma-2 [172].

Another development approach involves adapting general-purpose SRMs for coding tasks through fine-tuning [19,20,173]. A notable example is the CodeLlama series [173], derived from Llama2 [168], which undergoes extensive domain-specific training to specialize in programming languages like Python. These models are optimized for tasks such as syntax error detection, code recommendations, and infilling—where they predict and insert missing code segments.

### 5.4 Other domains

1) Legal applications: LaWGPT [174] is a family of models developed to enhance legal vocabulary coverage, pretrained on extensive Chinese legal corpora to improve semantic understanding in the legal domain. Lawyer LLaMA [175] serves as a Chinese legal SRM trained on comprehensive legal datasets to assist in legal guidance, case evaluation, and drafting legal documents. ChatLaw [176] is a series of open-source legal SRMs, including ChatLaw-13B and ChatLaw-33B, trained on a large corpus of legal news, forum discussions, and judicial interpretations.

2) Financial applications: Fin-R1 [177] produces a high-quality CoT dataset, carefully distilled and filtered from multiple authoritative financial sources, focused on professional financial reasoning tasks. A financial SRM is also specifically trained to fulfill industry requirements such as decision-making support and numerical accuracy.

Together, these domain SRMs highlight the powerful synergy of specialized datasets, continual training, and tailored architectures, enabling high-quality reasoning within focused areas of expertise.

### 5.5 Ethical considerations and their migration for applications

The knowledge transfer of LRM s to SRMs with fewer parameters

often introduces significant challenges such as bias and hallucination [152,178–179]. These biases tend to manifest differently across domains, necessitating targeted debiasing approaches tailored to domain-specific characteristics [153,180].

In healthcare, medical decision-making is shaped by a complex interplay of patient adherence, clinicians' experiential knowledge, ethical considerations, and inherent cognitive biases. Studies have shown that SRMs processing clinical questions containing cognitive biases exhibit significantly reduced accuracy compared to unbiased question formulations [181]. For instance, the BiasMedQA dataset [153] includes clinical vignettes annotated with a range of cognitive biases. SRMs tasked with these vignettes must identify correct diagnoses while coping with the embedded biased context, highlighting a critical challenge in domain-specific model robustness.

Similarly, in the financial domain, a phenomenon known as “company-specific biases” has been observed [180]. Here, language models' general knowledge of firms can influence sentiment analysis of financial texts, where sentiment scores differ notably when company names are present versus anonymized. This discrepancy, quantified as company-specific bias, not only affects text interpretation but also has broader economic implications. Researchers have developed economic models demonstrating that such biases can systematically distort investor behavior and ultimately impact stock prices when these skewed sentiments are widely adopted.

While this survey primarily focuses on technical advancements in SRMs, it is crucial to acknowledge that SRMs often inherit biases, fairness issues, and privacy vulnerabilities from their larger LRM predecessors. Such inherited problems may propagate into downstream applications if left unchecked. Addressing these challenges requires integrating comprehensive bias auditing protocols throughout the SRM development lifecycle. This includes rigorous evaluation on diverse, representative, and fairness-focused benchmarks [154,182]. Promising mitigation strategies comprise counterfactual data augmentation [183], bias-aware training methodologies [184], and embedding ethical constraints directly into reasoning procedures [155]. Coverage of these approaches exceeds the scope of this survey; therefore, readers interested in a deeper exploration are encouraged to consult dedicated reviews and studies on bias, fairness, and ethical AI in language models [185–187].

## ■ 6 Future research directions

In this section, we highlight several promising avenues for future research on SRMs, aiming to address current limitations and unlock their full potential.

### 1) Enhanced distillation techniques

While current distillation methods have effectively transferred knowledge from LRMs into more compact and efficient SRMs, there remains considerable room for improvement. First, due to their reduced size and limited capacity, SRMs often struggle to master highly challenging reasoning tasks. A promising direction involves developing iterative knowledge transfer frameworks that progressively distill increasingly complex reasoning capabilities from

LRMs to SRMs. By starting with fundamental reasoning problems and gradually incorporating more sophisticated CoT reasoning techniques, it is possible to yield SRMs that are both more robust and generalizable across diverse tasks. Second, although SFT-based distillation has achieved notable success, integrating RL within the distillation loop could further enhance performance. This integration would enable the identification and correction of specific weaknesses in SRMs through targeted feedback, thereby fostering continual improvement in reasoning abilities. Finally, enriching SRMs by incorporating external knowledge sources, such as knowledge graphs, domain-specific ontologies, or curated knowledge bases, during the distillation process could better contextualize reasoning. This enrichment would endow models with a broader background knowledge and more accurate inference skills, ultimately leading to stronger generalization and adaptability in real-world applications.

## 2) Adaptive RL strategies

Adaptive RL methods present a highly promising avenue for advancing the capabilities of SRMs. Given the inherently limited capacity of SRMs to effectively explore solution spaces that deviate significantly from their initial training distributions, it becomes critical to develop adaptive mechanisms that dynamically balance exploration and exploitation in a manner tailored to the unique characteristics of SRMs. For instance, implementing adjustable exploration rates that adapt based on factors such as task complexity, model confidence, or real-time performance metrics could enable SRMs to more effectively navigate complex learning landscapes, thereby avoiding premature convergence to suboptimal solutions. Furthermore, the design of task-specific reward functions plays a crucial role in guiding SRMs to capture nuanced decision criteria, which in turn drives measurable improvements in both accuracy and computational efficiency. In addition to these strategies, incorporating continual learning frameworks that allow SRMs to iteratively update their knowledge base and policy decisions based on new interactions, external feedback, and shifting domain distributions will be essential. Such continual adaptation mechanisms will support sustained, long-term reasoning performance, ensuring that SRMs remain resilient and effective as they encounter novel challenges and evolving real-world environments.

## 3) Learning and inference in low-resource settings

Expanding the applicability of SRMs to low-resource scenarios represents a critical and timely research frontier. Many real-world domains are characterized by scarce high-quality data, limited computational resources, and constrained deployment environments, all of which pose significant challenges to effective reasoning model development. One promising direction is to investigate methods for cross-domain knowledge transfer, such as leveraging CoT datasets and models pre-trained on related or higher-resource domains. Such approaches can substantially enrich the reasoning capabilities of SRMs in low-resource settings without incurring the high costs associated with extensive manual annotation. While parameter-efficient fine-tuning techniques for SFT have been extensively studied [64,65], their efficacy and adaptation in RL settings for

SRMs remain largely unexplored. Systematic investigations are therefore necessary to elucidate when and how these parameter-efficient methods can effectively reduce computational burdens while simultaneously preserving or even enhancing learning performance during RL-based training of SRMs. Addressing these challenges will be crucial for enabling robust, scalable, and accessible SRMs capable of functioning effectively under low-resource constraints.

In addition to model training in resource-constrained environments, recent advances in model compression and optimization techniques have gained increasing importance, particularly lower-precision inference methods. For instance, QLLM [188] effectively quantizes models to 4 bits on a single GPU. DecoupleQ [189] is a Post-Training Quantization (PTQ) method that improves accuracy for quantized models at 2 bits. More recently, PTQ-1.61 [190] has been proposed as an ultra-low-bit PTQ approach, enabling weight quantization to 1.61 bits. ParetoQ [191] serves as a general evaluation framework for comparing models across various quantization levels, including 1-bit, 1.58-bit, 2-bit, 3-bit, and 4-bit settings. These techniques considerably reduce memory footprint and computational overhead while preserving reasoning performance. Future research should rigorously explore integrating ultra-low-bit quantization with reasoning-specific fine-tuning to maintain or enhance model accuracy. Furthermore, combining quantization with complementary optimization methods, such as knowledge distillation or PEFT [192], presents promising directions to improve the accessibility and applicability of SRMs across diverse deployment scenarios.

## 4) Agent-based efficient inference

Agent-based cooperative reasoning has demonstrated substantial promise in improving the performance of SRMs on complex and challenging tasks. Despite these advances, current multi-agent systems often suffer from significant token overheads and increased operational costs, which pose serious limitations on their scalability and widespread adoption. To address these challenges, future research should prioritize the seamless integration of efficient communication and coordination strategies within mainstream multi-agent SRM frameworks. This includes the development of mechanisms to actively filter out redundant or potentially malicious interactions, thereby ensuring the integrity and efficiency of agent collaboration [110,193]. Specifically, optimizing adjacency matrices that represent communication graphs could allow these systems to dynamically identify and prune both redundant agents and unnecessary cross-round interactions. Such dynamic pruning would not only improve token efficiency but also enhance overall task performance by focusing computational resources on the most relevant information exchange. These advancements are critical to enabling more scalable, reliable, and cost-effective deployment of collaborative SRMs, paving the way for their application in diverse real-world scenarios with complex reasoning requirements.

## 5) Expanding domain-specific SRMs

The ongoing development of domain-specific SRMs presents significant opportunities to enhance their capabilities across three

critical dimensions: inference efficiency, task-specific performance, and reinforcement learning optimization. Current researches indicate substantial potential for architectural innovations that could further accelerate SRMs' inference speeds through dynamic sparse attention mechanisms [194] and advanced quantization techniques [195], particularly when processing domain-specific data patterns. The pursuit of superior task performance necessitates deeper investigation into meta-learning frameworks [196] and novel model thinking [197] that can more effectively capture domain knowledge structures while maintaining model efficiency. Reinforcement learning approaches offer promising pathways for refining SRMs' outputs to better align with domain expert expectations, though this requires developing more sophisticated reward functions capable of quantifying nuanced domain requirements [198–199]. Practical deployment considerations, including energy-efficient inference architectures and privacy-preserving specialization techniques, represent equally important research frontiers that will determine the real-world viability of these models [200]. We stress the importance of balancing model performance with inference-efficiency techniques for small domain models and of jointly optimizing both to better meet enterprise requirements.

#### 6) From Uni-modal to multi-modal reasoning

An important and emerging direction for SRMs is their extension to multi-modal reasoning applications, where models process and integrate information from diverse data modalities such as text, images, audio, and video. To date, SRM research has predominantly focused on uni-modal, text-based reasoning tasks, leaving a significant gap in their applicability to interdisciplinary challenges that require complex interactions across multiple sensory inputs. Notably, VisualThinker-R1-Zero [201] is among the first works to observe the “aha moment” using RL in visual reasoning with a 2B non-SFT model. Similarly, VLM-R1 [202] facilitates multi-modal SRMs to generate R1-style reasoning processes for visual tasks. Visual-RFT [203] further extends the combined training paradigm of SFT and RL to the multi-modal domain. We argue that this emerging field remains in its early stages. Future research should focus on designing specialized training paradigms, inference mechanisms, and architectural adaptations tailored for efficient and robust multi-modal reasoning, especially within the parameter constraints of SRMs.

#### 7) Deployment and application on edge devices

As SRMs inherently require fewer computational resources than LRM, they present a promising opportunity for deployment on edge devices such as smartphones, IoT gadgets, and embedded systems. Future research should explore optimizing SRMs for such environments, focusing on reducing latency, memory footprint, and power consumption without compromising reasoning accuracy. Techniques such as efficient model quantization tailored for reasoning workloads, adaptive inference strategies, and hardware-aware training could enable real-time, on-device reasoning capabilities in resource-constrained settings [204–205]. Moreover, applications on personalized assistants, offline knowledge retrieval, and privacy-sensitive scenarios stand to benefit substantially from

on-device SRMs. Addressing challenges related to robust and continuous learning in decentralized edge contexts [206] will also be critical to realizing practical and widespread adoption of SRMs beyond traditional server-based deployments.

### ■ 7 Concluding remarks

In conclusion, this survey has provided a comprehensive overview of Small Reasoning Models (SRMs), highlighting their rapid advancements and growing significance within the NLP community. The development of SRMs opens new avenues for deploying high-performance reasoning models in resource-constrained environments, making them crucial for both academic research and practical commercial applications. As SRMs continue to evolve, it is imperative that future research not only focuses on enhancing their core reasoning capabilities but also explores innovative strategies to integrate SRMs effectively across a broader spectrum of NLP tasks and real-world scenarios. Such integration could unlock new possibilities for efficient, explainable, and accessible AI systems.

While this survey offers a broad overview of SRMs and their diverse applications, it is important to acknowledge several inherent limitations. First, the field of reasoning models, both large and small, is rapidly evolving, with continuous innovations and novel methodologies emerging at a fast pace. Consequently, it is inevitable that some of the very latest advancements may not be fully captured in our review. Second, our analysis primarily relies on publicly available, peer-reviewed literature and open-source releases. This dependence may introduce a publication bias, as proprietary or unpublished developments in SRMs, particularly by industry labs, might not be represented here. Finally, although we cover a wide range of application domains, some niche or emerging areas may be underrepresented or omitted, reflecting current research focus and the availability of documented results.

The exploration and advancement of SRMs carry substantial implications for the broader NLP community and AI research at large. Our survey has underscored the potential of SRMs to democratize access to state-of-the-art reasoning capabilities by significantly lowering the computational requirements compared to their larger counterparts. This democratization enables institutions and researchers with limited hardware resources to participate more fully in cutting-edge AI development, thus promoting greater inclusivity and diversity in AI research. Additionally, SRMs contribute toward the sustainability of AI technologies by aligning with efforts to reduce the carbon footprint associated with training and deploying large-scale models. These positive impacts highlight the value and importance of continued investment in SRM research and application.

### ■ Competing interests

The authors declare that they have no competing interests or financial conflicts to disclose.

### ■ Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation,

distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## ■ References

- [1] Zhao W X, Zhou K, Li J, Tang T, Wang X, Hou Y, Min Y, Zhang B, Zhang J, Dong Z, Du Y, Yang C, Chen Y, Chen Z, Jiang J, Ren R, Li Y, Tang X, Liu Z, Liu P, Nie J Y, Wen J R. A survey of large language models. 2023, arXiv preprint arXiv: 2303.18223
- [2] Xu F, Hao Q, Zong Z, Wang J, Zhang Y, Wang J, Lan X, Gong J, Ouyang T, Meng F, Shao C, Yan Y, Yang Q, Song Y, Ren S, Hu X, Li Y, Feng J, Gao C, Li Y. Towards large reasoning models: a survey of reinforced reasoning with large language models. 2025, arXiv preprint arXiv: 2501.09686
- [3] DeepSeek-AI. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning. 2025, arXiv preprint arXiv: 2501.12948
- [4] Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, Chi E H, Le Q V, Zhou D. Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1800
- [5] Fu Y, Peng H, Ou L, Sabharwal A, Khot T. Specializing smaller language models towards multi-step reasoning. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 420
- [6] Magister L C, Mallinson J, Adamek J, Malmi E, Severyn A. Teaching small language models to reason. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2023, 1773–1781
- [7] Shridhar K, Stolfo A, Sachan M. Distilling reasoning capabilities into smaller language models. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023. 2023, 7059–7073
- [8] Zhang Q, Liu Z, Pan S. The rise of small language models. IEEE Intelligent Systems, 2025, 40(1): 30–37
- [9] Yan J, Wang C, Zhang T, He X, Huang J, Zhang W. From complex to simple: unraveling the cognitive tree for reasoning with small language models. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023. 2023, 12413–12425
- [10] Zhang B, Liu Z, Cherry C, Firat O. When scaling meets LLM finetuning: the effect of data, model and finetuning method. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–20
- [11] Hu L, He H, Wang D, Zhao Z, Shao Y, Nie L. LLM vs small model? Large language model based text augmentation enhanced personality detection model. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 18234–18242
- [12] Plaat A, Wong A, Verberne S, Broekens J, van Stein N, Bäck T. Reasoning with large language models, a survey. 2024, arXiv preprint arXiv: 2407.11511
- [13] Huang J, Chang K C C. Towards reasoning in large language models: a survey. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023. 2023, 1049–1065
- [14] Giadikiaroglou P, Lymperaiou M, Filandrianos G, Stamou G. Puzzle solving using reasoning of large language models: a survey. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 11574–11591
- [15] Ahn J, Verma R, Lou R, Liu D, Zhang R, Yin W. Large language models for mathematical reasoning: progresses and challenges. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop. 2024, 225–237
- [16] Zhang X, Wang D, Dou L, Zhu Q, Che W. A survey of table reasoning with large language models. Frontiers of Computer Science, 2025, 19(9): 199348
- [17] Wei S, Tong Y, Zhou Z, Xu Y, Gao J, Wei T, He T, Lv W. Federated reasoning LLMs: a survey. Frontiers of Computer Science, 2025, 19(12): 1912613
- [18] Hui B, Yang J, Cui Z, Yang J, Liu D, Zhang L, Liu T, Zhang J, Yu B, Dang K, Yang A, Men R, Huang F, Ren X, Ren X, Zhou J, Lin J. Qwen2.5-coder technical report. 2024, arXiv preprint arXiv: 2409.12186
- [19] Guo D, Zhu Q, Yang D, Xie Z, Dong K, Zhang W, Chen G, Bi X, Wu Y, Li Y K, Luo F, Xiong Y, Liang W. DeepSeek-coder: when the large language model meets programming—the rise of code intelligence. 2024, arXiv preprint arXiv: 2401.14196
- [20] Lozhkov A, Li R, Allal L B, Cassano F, Lamy-Poirier J, Tazi N, Tang A, Pykhtar D, Liu J, Wei Y, Liu T, Tian M, Kocetkov D, Zucker A, Belkada Y, Wang Z, Liu Q, Abulkhanov D, Paul I, Li Z, Li W D, Risdal M, Li J, Zhu J, Zhuo T Y, Zheltonozhskii E, Dade N O O, Yu W, Krauß L, Jain N, Su Y, He X, Dey M, Abati E, Chai Y, Muennighoff N, Tang X, Oblokulov M, Akiki C, Marone M, Mou C, Mishra M, Gu A, Hui B, Dao T, Zebaze A, Dehaene O, Patry N, Xu C, McAuley J, Hu H, Scholak T, Paquet S, Robinson J, Anderson C J, Chapados N, Patwary M, Tajbakhsh N, Jernite Y, Ferrandis C M, Zhang L, Hughes S, Wolf T, Guha A, von Werra L, de Vries H. StarCoder 2 and the stack v2: the next generation. 2024, arXiv preprint arXiv: 2402.19173
- [21] Jiang J, Wang F, Shen J, Kim S, Kim S. A survey on large language models for code generation. 2024, arXiv preprint arXiv: 2406.00515
- [22] Yang A, Zhang B, Hui B, Gao B, Yu B, Li C, Liu D, Tu J, Zhou J, Lin J, Lu K, Xue M, Lin R, Liu T, Ren X, Zhang Z. Qwen2.5-math technical report: toward mathematical expert model via self-improvement. 2024, arXiv preprint arXiv: 2409.12122
- [23] Shao Z, Wang P, Zhu Q, Xu R, Song J, Bi X, Zhang H, Zhang M, Li Y K, Wu Y, Guo D. DeepSeekMath: pushing the limits of mathematical reasoning in open language models. 2024, arXiv preprint arXiv: 2402.03300
- [24] Ying H, Zhang S, Li L, Zhou Z, Shao Y, Fei Z, Ma Y, Hong J, Liu K, Wang Z, Wang Y, Wu Z, Li S, Zhou F, Liu H, Zhang S, Zhang W, Yan H, Qiu X, Wang J, Chen K, Lin D. InternLM-Math: open math large language models toward verifiable reasoning. 2024, arXiv preprint arXiv: 2402.06332

- [25] Muennighoff N, Yang Z, Shi W, Li X L, Fei-Fei L, Hajishirzi H, Zettlemoyer L, Liang P, Candès E, Hashimoto T. s1: simple test-time scaling. 2025, arXiv preprint arXiv: 2501.19393
- [26] Zhao Y, Yin H, Zeng B, Wang H, Shi T, Lyu C, Wang L, Luo W, Zhang K. Marco-01: towards open reasoning models for open-ended solutions. 2024, arXiv preprint arXiv: 2411.14405
- [27] Cai W, Wang C, Yan J, Huang J, Fang X. Reasoning with OmniThought: a large CoT dataset with verbosity and cognitive difficulty annotations. 2025, arXiv preprint arXiv: 2505.10937
- [28] Rein D, Hou B L, Stickland A C, Petty J, Pang R Y, Dirani J, Michael J, Bowman S R. GPQA: a graduate-level google-proof Q&A benchmark. 2023, arXiv preprint arXiv: 2311.12022
- [29] Jain N, Han K, Gu A, Li W D, Yan F, Zhang T, Wang S, Solar-Lezama A, Sen K, Stoica I. LiveCodeBench: holistic and contamination free evaluation of large language models for code. In: Proceedings of the 13th International Conference on Learning Representations. 2025, 1–41
- [30] Lightman H, Kosaraju V, Burda Y, Edwards H, Baker B, Lee T, Leike J, Schulman J, Sutskever I, Cobbe K. Let's verify step by step. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–24
- [31] Chen J, Cai Z, Ji K, Wang X, Liu W, Wang R, Hou J, Wang B. HuatuoGPT-01, towards medical complex reasoning with LLMs. 2024, arXiv preprint arXiv: 2412.18925
- [32] Yuan W, Yu J, Jiang S, Padthe K, Li Y, Wang D, Kulikov I, Cho K, Tian Y, Weston J E, Li X. NaturalReasoning: reasoning in the wild with 2.8M challenging questions. 2025, arXiv preprint arXiv: 2502.13124
- [33] Lu D, Tan X, Xu R, Yao T, Qu C, Chu W, Xu Y, Qi Y. SCP-116K: a high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain. 2025, arXiv preprint arXiv: 2501.15587
- [34] Mikulová M, Straka M, Štěpánek J, Štěpánková B, Hajic J. Quality and efficiency of manual annotation: pre-annotation bias. In: Proceedings of the 13th Language Resources and Evaluation Conference. 2022, 2909–2918
- [35] Kim H, Mitra K, Chen R L, Rahman S, Zhang D. MEGAnno+: a human-LLM collaborative annotation system. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. 2024, 168–176
- [36] Li J. Human-LLM hybrid text answer aggregation for crowd annotations. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 15609–15622
- [37] Wang X, Kim H, Rahman S, Mitra K, Miao Z. Human-LLM collaborative annotation through effective verification of LLM labels. In: Proceedings of 2024 CHI Conference on Human Factors in Computing Systems. 2024, 303
- [38] Movva R, Koh P W, Pierson E. Annotation alignment: comparing LLM and human annotations of conversational safety. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 9048–9062
- [39] Schick T, Dwivedi-Yu J, Dessí R, Raileanu R, Lomeli M, Hambro E, Zettlemoyer L, Cancedda N, Scialom T. Toolformer: language models can teach themselves to use tools. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 2997
- [40] Wang H, Qin Y, Lin Y, Pan J Z, Wong K F. Empowering large language models: tool learning for real-world interaction. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 2983–2986
- [41] Qiao S, Gui H, Lv C, Jia Q, Chen H, Zhang N. Making language models better tool learners with execution feedback. In: Proceedings of 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). 2024, 3550–3568
- [42] Kwon T, Palo N D, Johns E. Language models as zero-shot trajectory generators. IEEE Robotics and Automation Letters, 2024, 9(7): 6728–6735
- [43] Hsieh C Y, Li C L, Yeh C K, Nakhost H, Fujii Y, Ratner A, Krishna R, Lee C Y, Pfister T. Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023. 2023, 8003–8017
- [44] Ho N, Schmid L, Yun S Y. Large language models are reasoning teachers. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2023, 14852–14882
- [45] Li L H, Hessel J, Yu Y, Ren X, Chang K W, Choi Y. Symbolic chain-of-thought distillation: small models can also “think” step-by-step. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2023, 2665–2679
- [46] Yue Y, Wang C, Huang J, Wang P. Distilling instruction-following abilities of large language models with task-aware curriculum planning. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024. 2024, 6030–6054
- [47] Yue Y, Wang C, Huang J, Wang P. Building a family of data augmentation models for low-cost LLM fine-tuning on the cloud. In: Proceedings of the 31st International Conference on Computational Linguistics: Industry Track. 2025, 431–444
- [48] Yang Z, Pang T, Feng H, Wang H, Chen W, Zhu M, Liu Q. Self-distillation bridges distribution gap in language model fine-tuning. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024, 1028–1043
- [49] Tang Z, Zhang X, Wang B, Wei F. MathScale: scaling instruction tuning for mathematical reasoning. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 1954
- [50] Havrilla A, Raparthy S, Nalmpantis C, Dwivedi-Yu J, Zhuravynski M, Hambro E, Raileanu R. GLoRe: when, where, and how to improve LLM reasoning via global and local refinements. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 709
- [51] Qiao S, Zhang N, Fang R, Luo Y, Zhou W, Jiang Y E, Lv C, Chen H. AutoAct: automatic agent learning from scratch for QA via self-planning. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 3003–3021
- [52] Li C, Dong G, Xue M, Peng R, Wang X, Liu D. DotaMath: decomposition of thought with code assistance and self-correction for mathematical reasoning. 2024, arXiv preprint arXiv: 2407.04078
- [53] Song Y, Yin D, Yue X, Huang J, Li S, Lin B Y. Trial and error: exploration-based trajectory optimization for LLM agents. 2024, arXiv preprint arXiv: 2403.02502
- [54] Motwani S R, Smith C, Das R J, Rybchuk M, Torr P H S, Laptev I,

- Pizzati F, Clark R, de Witt C S. MALT: improving reasoning with multi-agent LLM training. 2024, arXiv preprint arXiv: 2412.01928
- [55] Kumar A, Zhuang V, Agarwal R, Su Y, Co-Reyes J D, Singh A, Baumli K, Iqbal S, Bishop C, Roelofs R, Zhang L M, McKinney K, Shrivastava D, Paduraru C, Tucker G, Precup D, Behbahani F, Faust A. Training language models to self-correct via reinforcement learning. In: Proceedings of the 13th International Conference on Learning Representations. 2025, 1–27
- [56] Zhang Z, Zheng C, Wu Y, Zhang B, Lin R, Yu B, Liu D, Zhou J, Lin J. The lessons of developing process reward models in mathematical reasoning. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025. 2025, 10495–10516
- [57] Luo H, Sun Q, Xu C, Zhao P, Lou J G, Tao C, Geng X, Lin Q, Chen S, Tang Y, Zhang D. *WizardMath*: empowering mathematical reasoning for large language models via *Reinforced Evol-Instruct*. In: Proceedings of the 13th International Conference on Learning Representations. 2025, 1–37
- [58] Wang P, Li L, Shao Z, Xu R, Dai D, Li Y, Chen D, Wu Y, Sui Z. Math-shepherd: verify and reinforce LLMs step-by-step without human annotations. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024, 9426–9439
- [59] Wang Z, Li Y, Wu Y, Luo L, Hou L, Yu H, Shang J. Multi-step problem solving through a verifier: An empirical analysis on model-induced process supervision. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024. 2024, 7309–7319
- [60] Zhang D, Zhoubian S, Hu Z, Yue Y, Dong Y, Tang J. ReSTMCTS\*: LLM self-training via process reward guided tree search. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 2066
- [61] Chen Z, White M, Mooney R, Payani A, Su Y, Sun H. When is tree search useful for LLM planning? it depends on the discriminator. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024, 13659–13678
- [62] Wang C, Yan J, Zhang W, Huang J. Towards better parameter-efficient fine-tuning for large language models: a position paper. 2023, arXiv preprint arXiv: 2311.13126
- [63] Han Z, Gao C, Liu J, Zhang J, Zhang S Q. Parameter-efficient fine-tuning for large models: a comprehensive survey. 2024, arXiv preprint arXiv: 2403.14608
- [64] Hu E J, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. LoRA: low-rank adaptation of large language models. In: Proceedings of the 10th International Conference on Learning Representations. 2022, 1–13
- [65] Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. QLORA: efficient finetuning of quantized LLMs. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 441
- [66] Zhang Q, Chen M, Bukharin A, He P, Cheng Y, Chen W, Zhao T. Adaptive budget allocation for parameter-efficient fine-tuning. In: Proceedings of the 11th International Conference on Learning Representations. 2023, 1–17
- [67] Sheng G, Zhang C, Ye Z, Wu X, Zhang W, Zhang R, Peng Y, Lin H, Wu C. HybridFlow: a flexible and efficient RLHF framework. In: Proceedings of the 20th European Conference on Computer Systems. 2024, 1279–1297
- [68] Mei Z, Fu W, Li K, Wang G, Zhang H, Wu Y. RealLFH: optimized RLHF training for large language models through parameter reallocation. 2024, arXiv preprint arXiv: 2406.14088
- [69] Mei Z, Fu W, Li K, Wang G, Zhang H, Wu Y. Real: efficient RLHF training of large language models with parameter reallocation. In: Proceedings of the 8th Conference on Machine Learning and Systems. 2025
- [70] Wen L, Cai Y, Xiao F, He X, An Q, Duan Z, Du Y, Liu J, Tang L, Lv X, Zou H, Deng Y, Jia S, Zhang X. Light-R1: curriculum SFT, DPO and RL for long cot from scratch and beyond. 2025, arXiv preprint arXiv: 2503.10460
- [71] Hu J, Wu X, Wang W, Xianyu, Zhang D, Cao Y. OpenRLHF: an easy-to-use, scalable and high-performance RLHF framework. 2024, arXiv preprint arXiv: 2405.11143
- [72] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C L, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A, Schulman J, Hilton J, Kelton F, Miller L, Simens M, Askell A, Welinder P, Christiano P, Leike J, Lowe R. Training language models to follow instructions with human feedback. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 2011
- [73] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. 2017, arXiv preprint arXiv: 1707.06347
- [74] Bai Y, Kadavath S, Kundu S, Askell A, Kernion J, Jones A, Chen A, Goldie A, Mirhoseini A, McKinnon C, Chen C, Olsson C, Olah C, Hernandez D, Drain D, Ganguli D, Li D, Tran-Johnson E, Perez E, Kerr J, Mueller J, Ladish J, Landau J, Ndousse K, Lukosuite K, Lovitt L, Sellitto M, Elhage N, Schiefer N, Mercado N, DasSarma N, Lasenby R, Larson R, Ringer S, Johnston S, Kravec S, El Showk S, Fort S, Lanham T, Telleen-Lawton T, Conerly T, Henighan T, Hume T, Bowman S R, Hatfield-Dodds Z, Mann B, Amodei D, Joseph N, McCandlish S, Brown T, Kaplan J. Constitutional AI: harmlessness from AI feedback. 2022, arXiv preprint arXiv: 2212.08073
- [75] Rafailov R, Sharma A, Mitchell E, Ermon S, Manning C D, Finn C. Direct preference optimization: your language model is secretly a reward model. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 2338
- [76] Ethayarajh K, Xu W, Muennighoff N, Jurafsky D, Kiela D. KTO: model alignment as prospect theoretic optimization. 2024, arXiv preprint arXiv: 2402.01306
- [77] Amini A, Vieira T, Cotterell R. Direct preference optimization with an offset. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024. 2024, 9954–9972
- [78] Meng Y, Xia M, Chen D. SimPO: simple preference optimization with a reference-free reward. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 3946
- [79] Trung L Q, Zhang X, Jie Z, Sun P, Jin X, Li H. ReFT: reasoning with reinforced fine-tuning. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 7601–7614
- [80] Kazemnejad A, Aghajohari M, Portelance E, Sordoni A, Reddy S, Courville A, Le Roux N. VinePPO: unlocking RL potential for LLM

- reasoning through refined credit assignment. 2024, arXiv preprint arXiv: 2410.01679
- [81] Wang T, Chen J, Han X, Bai J. CPL: critical plan step learning boosts LLM generalization in reasoning tasks. 2024, arXiv preprint arXiv: 2409.08642
- [82] Hwang H, Kim D, Kim S, Ye S, Seo M. Self-explore to avoid the pit: Improving the reasoning capabilities of language models with fine-grained rewards. 2024, arXiv preprint arXiv: 2404.10346
- [83] Setlur A, Nagpal C, Fisch A, Geng X, Eisenstein J, Agarwal R, Agarwal A, Berant J, Kumar A. Rewarding progress: scaling automated process verifiers for LLM reasoning. In: Proceedings of the 13th International Conference on Learning Representations. 2025, 1–31
- [84] Chen G, Liao M, Li C, Fan K. Step-level value preference optimization for mathematical reasoning. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024. 2024, 7889–7903
- [85] Xie Y, Goyal A, Zheng W, Kan M Y, Lillicrap T P, Kawaguchi K, Shieh M. Monte Carlo tree search boosts reasoning via iterative preference learning. 2024, arXiv preprint arXiv: 2405.00451
- [86] Wang C, Deng Y, Lyu Z, Zeng L, He J, Yan S, An B. Q\*: improving multi-step reasoning for LLMs with deliberative planning. 2024, arXiv preprint arXiv: 2406.14283
- [87] Guan X, Zhang L L, Liu Y, Shang N, Sun Y, Zhu Y, Yang F, Yang M. rStar-math: small LLMs can master math reasoning with self-evolved deep thinking. 2025, arXiv preprint arXiv: 2501.04519
- [88] Brown T B, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D M, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D. Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 159
- [89] Chiang T R, Chen Y N. Semantically-aligned equation generation for solving and reasoning math word problems. In: Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019, 2656–2668
- [90] Nye M I, Andreassen A J, Gur-Ari G, Michalewski H, Austin J, Bieber D, Dohan D, Lewkowycz A, Bosma M, Luan D, Sutton C, Odena A. Show your work: Scratchpads for intermediate computation with language models. 2021, arXiv preprint arXiv: 2112.00114
- [91] Snell C, Lee J, Xu K, Kumar A. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. 2024, arXiv preprint arXiv: 2408.03314
- [92] Wu Y, Sun Z, Li S, Welleck S, Yang Y. An empirical analysis of compute-optimal inference for problem-solving with language models. 2024, arXiv preprint arXiv: 2408.00724
- [93] Wang X, Wei J, Schuurmans D, Le Q V, Chi E H, Narang S, Chowdhery A, Zhou D. Self-consistency improves chain of thought reasoning in language models. In: Proceedings of the 11th International Conference on Learning Representations. 2023, 1–24
- [94] Yao S, Yu D, Zhao J, Shafran I, Griffiths T L, Cao Y, Narasimhan K. Tree of thoughts: Deliberate problem solving with large language models. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 517
- [95] Sel B, Al-Tawaha A, Khattar V, Jia R, Jin M. Algorithm of thoughts: enhancing exploration of ideas in large language models. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 1797
- [96] Ning X, Lin Z, Zhou Z, Wang Z, Yang H, Wang Y. Skeleton-of-thought: prompting LLMs for efficient parallel generation. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–51
- [97] Long J. Large language model guided tree-of-thought. 2023, arXiv preprint arXiv: 2305.08291
- [98] Du Y, Li S, Torralba A, Tenenbaum J B, Mordatch I. Improving factuality and reasoning in language models through multiagent debate. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 467
- [99] Liang T, He Z, Jiao W, Wang X, Wang Y, Wang R, Yang Y, Shi S, Tu Z. Encouraging divergent thinking in large language models through multi-agent debate. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 17889–17904
- [100] Wang Z, Mao S, Wu W, Ge T, Wei F, Ji H. Unleashing the emergent cognitive synergy in large language models: a task-solving agent through multi-persona self-collaboration. In: Proceedings of 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). 2024, 257–279
- [101] Zhang J, Xu X, Zhang N, Liu R, Hooi B, Deng S. Exploring collaboration mechanisms for LLM agents: a social psychology view. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024, 14544–14607
- [102] Qian C, Liu W, Liu H, Chen N, Dang Y, Li J, Yang C, Chen W, Su Y, Cong X, Xu J, Li D, Liu Z, Sun M. ChatDev: communicative agents for software development. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024, 15174–15186
- [103] Hong S, Zhuge M, Chen J, Zheng X, Cheng Y, Wang J, Zhang C, Wang Z, Yau S K S, Lin Z, Zhou L, Ran C, Xiao L, Wu C, Schmidhuber J. MetaGPT: meta programming for a multi-agent collaborative framework. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–29
- [104] Holt S, Luyten M R, van der Schaar M. L2MAC: large language model automatic computer for extensive code generation. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–61
- [105] Wu Q, Bansal G, Zhang J, Wu Y, Zhang S, Zhu E, Li B, Jiang L, Zhang X, Wang C. AutoGen: enabling next-gen LLM applications via multi-agent conversation framework. 2023, arXiv preprint arXiv: 2308.08155
- [106] Yan Y, Zhang Y, Huang K. Depending on yourself when you should: Mentoring LLM with RL agents to become the master in cybersecurity games. 2024, arXiv preprint arXiv: 2403.17674
- [107] Zhou Z, Hu B, Zhao C, Zhang P, Liu B. Large language model as a policy teacher for training reinforcement learning agents. In:

- Proceedings of the 33rd International Joint Conference on Artificial Intelligence. 2024, 627
- [108] Jiang D, Ren X, Lin B Y. LLM-blender: ensembling large language models with pairwise ranking and generative fusion. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2023, 14165–14178
- [109] Zhuge M, Wang W, Kirsch L, Faccio F, Khizbulin D, Schmidhuber J. GPTSwarm: language agents as optimizable graphs. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 2597
- [110] Wang Z, Wang Y, Liu X, Ding L, Zhang M, Liu J, Zhang M. AgentDropout: dynamic agent elimination for token-efficient and high-performance LLM-based multi-agent collaboration. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2025, 24013–24035
- [111] Wang S, Tan Z, Chen Z, Zhou S, Chen T, Li J. AnyMAC: cascading flexible multi-agent collaboration via next-agent prediction. In: Proceedings of 2025 Conference on Empirical Methods in Natural Language Processing. 2025, 11567–1157
- [112] Jiang M, Ruan Y, Lastras L, Kapanipathi P, Hashimoto T. Putting it all into context: simplifying agents with LCIMs. 2025, arXiv preprint arXiv: 2505.08120
- [113] Zheng C, Liu Z, Xie E, Li Z, Li Y. Progressive-hint prompting improves reasoning in large language models. 2023, arXiv preprint arXiv: 2304.09797
- [114] Liu Z, Zhang Y, Li P, Liu Y, Yang D. Dynamic LLM-agent network: an LLM-agent collaboration framework with agent team optimization. 2023, arXiv preprint arXiv: 2310.02170
- [115] Shinn N, Labash B, Gopinath A. Reflexion: an autonomous agent with dynamic memory and self-reflection. 2023, arXiv preprint arXiv: 2303.11366
- [116] Fu Y, Peng H, Khot T, Lapata M. Improving language model negotiation with self-play and in-context learning from AI feedback. 2023, arXiv preprint arXiv: 2305.10142
- [117] Chen P, Zhang S, Han B. CoMM: collaborative multi-agent, multi-reasoning-path prompting for complex problem solving. In: Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2024. 2024, 1720–1738
- [118] Pesce E, Montana G. Learning multi-agent coordination through connectivity-driven communication. Machine Learning, 2023, 112(2): 483–514
- [119] Liu Y, Dou Y, Li Y, Xu X, Liu D. Temporal dynamic weighted graph convolution for multi-agent reinforcement learning. In: Proceedings of the 44th Annual Meeting of the Cognitive Science Society. 2022, 743–749
- [120] Hu S, Shen L, Zhang Y, Tao D. Learning multi-agent communication from graph modeling perspective. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–16
- [121] Chan C M, Chen W, Su Y, Yu J, Xue W, Zhang S, Fu J, Liu Z. ChatEval: towards better LLM-based evaluators through multi-agent debate. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–15
- [122] Zelikman E, Lorch E, Mackey L, Kalai A T. Self-taught optimizer (STOP): recursively self-improving code generation. 2023, arXiv preprint arXiv: 2310.02304
- [123] Khattab O, Singhvi A, Maheshwari P, Zhang Z, Santhanam K, Vardhamanan S, Haq S, Sharma A, Joshi T T, Moazam H, Miller H, Zaharia M, Potts C. DSPy: compiling declarative language model calls into self-improving pipelines. 2023, arXiv preprint arXiv: 2310.03714
- [124] Qian C, Xie Z, Wang Y, Liu W, Zhu K, Xia H, Dang Y, Du Z, Chen W, Yang C, Liu Z, Sun M. Scaling large language model-based multi-agent collaboration. In: Proceedings of the 13th International Conference on Learning Representations. 2025, 1–18
- [125] Lample G, Lachaux M A, Lavril T, Martinet X, Hayat A, Ebner G, Rodriguez A, Lacroix T. HyperTree proof search for neural theorem proving. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1910
- [126] Bi Z, Han K, Liu C, Tang Y, Wang Y. Forest-of-thought: scaling test-time compute for enhancing LLM reasoning. 2024, arXiv preprint arXiv: 2412.09078
- [127] Pope R, Douglas S, Chowdhery A, Devlin J, Bradbury J, Heek J, Xiao K, Agrawal S, Dean J. Efficiently scaling transformer inference. In: Proceedings of the 6th Conference on Machine Learning and Systems. 2023
- [128] Chen Y, Pan X, Li Y, Ding B, Zhou J. A simple and provable scaling law for the test-time compute of large language models. 2024, arXiv preprint arXiv: 2411.19477
- [129] Welleck S, Lu X, West P, Brahman F, Shen T, Khashabi D, Choi Y. Generating sequences by learning to self-correct. In: Proceedings of the 11th International Conference on Learning Representations. 2023, 1–19
- [130] Madaan A, Tandon N, Gupta P, Hallinan S, Gao L, Wiegrefe S, Alon U, Dziri N, Prabhumoye S, Yang Y, Gupta S, Majumder B P, Hermann K, Welleck S, Yazdanbakhsh A, Clark P. SELF-REFINE: iterative refinement with self-feedback. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 2019
- [131] Chen X, Lin M, Schärli N, Zhou D. Teaching large language models to self-debug. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–81
- [132] Chen J, Ren J, Chen X, Yang C, Sun R, Arik S Ö. SETS: leveraging self-verification and self-correction for improved test-time scaling. 2025, arXiv preprint arXiv: 2501.19306
- [133] Hou Z, Lv X, Lu R, Zhang J, Li Y, Yao Z, Li J, Tang J, Dong Y. T1: advancing language model reasoning through reinforcement learning and inference scaling. 2025, arXiv preprint arXiv: 2501.11651
- [134] Lee K H, Fischer I, Wu Y H, Marwood D, Baluja S, Schuurmans D, Chen X. Evolving deeper LLM thinking. 2025, arXiv preprint arXiv: 2501.09891
- [135] Choi S, Fang T, Wang Z, Song Y. KCTS: knowledge-constrained tree search decoding with token-level hallucination detection. In: Proceedings of 2023 Conference on Empirical Methods in Natural Language Processing. 2023, 14035–14053
- [136] Zhang S, Chen Z, Shen Y, Ding M, Tenenbaum J B, Gan C. Planning with large language models for code generation. In: Proceedings of the 11th International Conference on Learning Representations. 2023, 1–28
- [137] Zhou A, Yan K, Shlapentokh-Rothman M, Wang H, Wang Y X. Language agent tree search unifies reasoning, acting, and planning in

- language models. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 2572
- [138] Xie Y, Kawaguchi K, Zhao Y, Zhao J X, Kan M Y, He J, Xie M Q. Self-evaluation guided beam search for reasoning. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 1802
- [139] Gandhi K, Lee D, Grand G, Liu M, Cheng W, Sharma A, Goodman N D. Stream of search (SoS): learning to search in language. 2024, arXiv preprint arXiv: 2404.03683
- [140] Xin H, Guo D, Shao Z, Ren Z, Zhu Q, Liu B, Ruan C, Li W, Liang X. DeepSeek-prover: advancing theorem proving in LLMs through large-scale synthetic data. 2024, arXiv preprint arXiv: 2405.14333
- [141] Ankner Z, Paul M, Cui B, Chang J D, Ammanabrolu P. Critique-out-loud reward models. 2024, arXiv preprint arXiv: 2408.11791
- [142] Wan Y, Wu J, Abdulhai M, Shani L, Jaques N. Enhancing personalized multi-turn dialogue with curiosity reward. 2025, arXiv preprint arXiv: 2504.03206
- [143] Heo D, Rim D N, Choi H. Dynamic preference multi-objective reinforcement learning for internet network management. 2025, arXiv preprint arXiv: 2506.13153
- [144] Li C, Zhang H, Xu Y, Xue H, Ao X, He Q. Gradient-adaptive policy optimization: towards multi-objective alignment of large language models. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2025, 11214–11232
- [145] Nguyen T D, Ting Y S, Ciucă I, O'Neill C, Sun Z C, Jabłomska M, Kruk S, Perkowski E, Miller J, Li J J, Peek J, Iyer K, Różański T, Khetarpal P, Zaman S, Brodrick D, Rodríguez Méndez S J, Bui T, Goodman A, Accomazzi A, Naiman J, Cranney J, Schawinski K, Răileanu R, UniverseTBD. AstroLLaMA: towards specialized foundation models in astronomy. In: Proceedings of the 2nd Workshop on Information Extraction from Scientific Publications. 2023, 49–55
- [146] Yang K, Zhang T, Kuang Z, Xie Q, Huang J, Ananiadou S. MentaLLaMA: interpretable mental health analysis on social media with large language models. In: Proceedings of the ACM Web Conference. 2024, 4489–4500
- [147] Zhang D, Hu Z, Zhoubian S, Du Z, Yang K, Wang Z, Yue Y, Dong Y, Tang J. SciGLM: training scientific language models with self-reflective instruction annotation and tuning. 2024, arXiv preprint arXiv: 2401.07950
- [148] Zhang D, Liu W, Tan Q, Chen J, Yan H, Yan Y, Li J, Huang W, Yue X, Zhou D, Zhang S, Su M, Zhong H, Li Y, Ouyang W. ChemLLM: a chemical large language model. 2024, arXiv preprint arXiv: 2402.06852
- [149] Acikgoz E C, Ince O B, Bench R, Boz A A, Kesenci I, Erdem A, Erdem E. Hippocrates: an open-source framework for advancing large language models in healthcare. 2024, arXiv preprint arXiv: 2404.16621
- [150] Yang Y, Sun H, Li J, Liu R, Li Y, Liu Y, Huang H, Gao Y. MindLLM: pre-training lightweight large language model from scratch, evaluations and domain applications. 2023, arXiv preprint arXiv: 2310.15777
- [151] Yao Y, Huang S, Wang W, Dong L, Wei F. Adapt-and-distill: developing small, fast and effective pretrained language models for domains. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. 2021, 460–470
- [152] Dige O, Arneja D, Yau T F, Zhang Q, Bolandraftar M, Zhu X, Khattak F K. Can machine unlearning reduce social bias in language models? In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 954–969
- [153] Schmidgall S, Harris C, Essien I, Olshvng D, Rahman T, Kim J W, Ziae R, Eshraghian J, Abadir P, Chellappa R. Addressing cognitive bias in medical language models. 2024, arXiv preprint arXiv: 2402.08113
- [154] Manerba M M, Stanczak K, Guidotti R, Augenstein I. Social bias probing: fairness benchmarking for language models. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 14653–14671
- [155] Upreti N, Ciupa J, Belle V. Towards developing ethical reasoners: Integrating probabilistic reasoning and decision-making for complex AI systems. In: Proceedings of the 17th International Conference on Agents and Artificial Intelligence-Volume 1: ICAART. 2025, 588–599
- [156] Zhao Z, Jin Q, Yu S. PMC-patients: a large-scale dataset of patient notes and relations extracted from case reports in PubMed central. 2022, arXiv preprint arXiv: 2202.13876
- [157] Jin Q, Dhingra B, Liu Z, Cohen W W, Lu X. PubMedQA: a dataset for biomedical research question answering. In: Proceedings of 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019, 2567–2577
- [158] Chen Z, Hernández Cano A, Romanou A, Bonnet A, Matoba K, Salvi F, Pagliardini M, Fan S, Köpf A, Mohtashami A, Sallinen A, Sakhaeirad A, Swamy V, Krawczuk I, Bayazit D, Marmet A, Montariol S, Hartley M A, Jaggi M, Bosselut A. MEDITRON-70B: scaling medical pretraining for large language models. 2023, arXiv preprint arXiv: 2311.16079
- [159] Bolton E, Venigalla A, Yasunaga M, Hall D, Xiong B, Lee T, Daneshjou R, Frankle J, Liang P, Carbin M, Manning C D. BioMedLM: a 2.7B parameter language model trained on biomedical text. 2024, arXiv preprint arXiv: 2403.18421
- [160] Gao L, Biderman S, Black S, Golding L, Hoppe T, Foster C, Phang J, He H, Thite A, Nabeshima N, Presser S, Leahy C. The pile: an 800GB dataset of diverse text for language modeling. 2021, arXiv preprint arXiv: 2101.00027
- [161] OpenAI. GPT-4 technical report. 2023, arXiv preprint arXiv: 2303.08774
- [162] Azerbayev Z, Schoelkopf H, Paster K, Santos M D, McAleer S M, Jiang A Q, Deng J, Biderman S, Welleck S. Llemma: an open language model for mathematics. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–28
- [163] Rozière B, Gehring J, Gloeckle F, Sootla S, Gat I, Tan X E, Adi Y, Liu J, Remez T, Rapin J, Kozhevnikov A, Evtimov I, Bitton J, Bhattacharyya M, Canton Ferrer C, Grattafiori A, Xiong W, Défossez A, Copet J, Azhar F, Touvron H, Martin L, Usunier N, Scialom T, Synnaeve G. Code Llama: open foundation models for code. 2023, arXiv preprint arXiv: 2308.12950
- [164] Hendrycks D, Burns C, Basart S, Zou A, Mazeika M, Song D, Steinhardt J. Measuring massive multitask language understanding. In: Proceedings of the 9th International Conference on Learning Representations. 2021, 1–27
- [165] Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, Kaiser L, Plappert M, Tworek J, Hilton J, Nakano R, Hesse C, Schulman J.

- Training verifiers to solve math word problems. 2021, arXiv preprint arXiv: 2110.14168
- [166] Lewkowycz A, Andreassen A, Dohan D, Dyer E, Michalewski H, Ramasesh V V, Slone A, Anil C, Schlag I, Gutman-Solo T, Wu Y, Neyshabur B, Gur-Ari G, Misra V. Solving quantitative reasoning problems with language models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 278
- [167] Cai Z, Cao M, Chen H, Chen K, Chen K, Chen X, Chen X, Chen Z, Chen Z, Chu P, Dong X, Duan H, Fan Q, Fei Z, Gao Y, Ge J, Gu C, Gu Y, Gui T, Guo A, Guo Q, He C, Hu Y, Huang T, Jiang T, Jiao P, Jin Z, Lei Z, Li J, Li J, Li L, Li S, Li W, Li Y, Liu H, Liu J, Hong J, Liu K, Liu K, Liu X, Lv C, Lv H, Lv K, Ma L, Ma R, Ma Z, Ning W, Ouyang L, Qiu J, Qu Y, Shang F, Shao Y, Song D, Song Z, Sui Z, Sun P, Sun Y, Tang H, Wang B, Wang G, Wang J, Wang J, Wang R, Wang Y, Wang Z, Wei X, Weng Q, Wu F, Xiong Y, Xu C, Xu R, Yan H, Yan Y, Yang X, Ye H, Ying H, Yu J, Yu J, Zang Y, Zhang C, Zhang L, Zhang P, Zhang P, Zhang R, Zhang S, Zhang S, Zhang W, Zhang W, Zhang X, Zhang X, Zhao H, Zhao Q, Zhao X, Zhou F, Zhou Z, Zhuo J, Zou Y, Qiu X, Qiao Y, Lin D. InternLM2 technical report. 2024, arXiv preprint arXiv: 2403.17297
- [168] Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, Bashlykov N, Batra S, Bhargava P, Bhosale S, Bikell D, Blecher L, Canton Ferrer C, Chen M, Cucurull G, Esiobu D, Fernandes J, Fu J, Fu W, Fuller B, Gao C, Goswami V, Goyal N, Hartshorn A, Hosseini S, Hou R, Inan H, Kardas M, Kerkez V, Khabsa M, Kloumann I, Korenev A, Koura P S, Lachaux M A, Lavril T, Lee J, Liskovich D, Lu Y, Mao Y, Martinet X, Mihaylov T, Mishra P, Molybog I, Nie Y, Poulton A, Reizenstein J, Rungta R, Saladi K, Schelten A, Silva R, Smith E M, Subramanian R, Tan X E, Tang B, Taylor R, Williams A, Kuan J X, Xu P, Yan Z, Zarov I, Zhang Y, Fan A, Kambadur M, Narang S, Rodriguez A, Stojnic R, Edunov S, Scialom T. Llama 2: Open foundation and fine-tuned chat models. 2023, arXiv preprint arXiv: 2307.09288
- [169] Gunasekar S, Zhang Y, Aneja J, Mendes C C T, Giorno A D, Gopi S, Javaheripi M, Kauffmann P, de Rosa G, Saarikivi O, Salim A, Shah S, Behl H S, Wang X, Bubeck S, Eldan R, Kalai A T, Lee Y T, Li Y. Textbooks are all you need. 2023, arXiv preprint arXiv: 2306.11644
- [170] Abdin M, Jacobs S A, Awan A A, Aneja J, Awadallah A, Awadalla H, Bach N, Bahree A, Bakhtiari A, Behl H, Benhaim A, Bilenko M, Bjorck J, Bubeck S, Cai M, Mendes C C T, Chen W, Chaudhary V, Chopra P, Giorno A D, de Rosa G, Dixon M, Eldan R, Iter D, Garg A, Goswami A, Gunasekar S, Haider E, Hao J, Hewett R J, Huynh J, Javaheripi M, Jin X, Kauffmann P, Karampatziakis N, Kim D, Khademi M, Kurilenko L, Lee J R, Lee Y T, Li Y, Liang C, Liu W, Lin E, Lin Z, Madan P, Mitra A, Modi H, Nguyen A, Norick B, Patra B, Perez-Becker D, Portet T, Pryzant R, Qin H, Radmilac M, Rosset C, Roy S, Ruwase O, Saarikivi O, Saied A, Salim A, Santacroce M, Shah S, Shang N, Sharma H, Song X, Tanaka M, Wang X, Ward R, Wang G, Witte P A, Wyatt M, Xu C, Xu J, Yadav S, Yang F, Yang Z, Yu D, Zhang C, Zhang C, Zhang J, Zhang L L, Zhang Y, Zhang Y, Zhang Y, Zhou X. Phi-3 technical report: a highly capable language model locally on your phone. 2024, arXiv preprint arXiv: 2404.14219
- [171] Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, et al. The Llama 3 herd of models. 2024, arXiv preprint arXiv: 2407.21783
- [172] Gemma Team. Gemma 2: improving open language models at a practical size. 2024, arXiv preprint arXiv: 2408.00118
- [173] Rozière B, Gehring J, Gloeckle F, Sootla S, Gat I, Tan X E, Adi Y, Liu J, Sauvestre R, Remez T, Rapin J, Kozhevnikov A, Evtimov I, Bitton J, Bhatt M, Ferrer C C, Grattafiori A, Xiong W, Défossez A, Copet J, Azhar F, Touvron H, Martin L, Usunier N, Scialom T, Synnaeve G. Code Llama: open foundation models for code. 2024, arXiv preprint arXiv: 2308.12950
- [174] Zhou Z, Shi J X, Song P X, Yang X W, Jin Y X, Guo L Z, Li Y F. LawGPT: a Chinese legal knowledge-enhanced large language model. 2024, arXiv preprint arXiv: 2406.04614
- [175] Huang Q, Tao M, An Z, Zhang C, Jiang C, Chen Z, Wu Z, Feng Y. Lawyer LLaMA technical report. 2023, arXiv preprint arXiv: 2305.15062
- [176] Cui J, Li Z, Yan Y, Chen B, Yuan L. ChatLaw: open-source legal large language model with integrated external knowledge bases. 2023, arXiv preprint arXiv: 2306.16092
- [177] Liu Z, Guo X, Lou F, Zeng L, Niu J, Wang Z, Xu J, Cai W, Yang Z, Zhao X, Li C, Xu S, Chen D, Chen Y, Bai Z, Zhang L. Fin-R1: a large language model for financial reasoning through reinforcement learning. 2025, arXiv preprint arXiv: 2503.16252
- [178] Lin Z, Guan S, Zhang W, Zhang H, Li Y, Zhang H. Towards trustworthy LLMs: a review on debiasing and dehallucinating in large language models. Artificial Intelligence Review, 2024, 57(9): 243
- [179] Liu G, Xue Z, Zhang X, Wang R, Johnson K. Smaller large language models can do moral self-correction. In: Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025). 2025, 56–65
- [180] Nakagawa K, Hirano M, Fujimoto Y. Evaluating company-specific biases in financial sentiment analysis using large language models. In: Proceedings of 2024 IEEE International Conference on Big Data. 2024, 6614–6623
- [181] Mahajan A, Obermeyer Z, Daneshjou R, Lester J, Powell D. Cognitive bias in clinical large language models. npj Digital Medicine, 2025, 8(1): 428
- [182] Wang S, Wang P, Zhou T, Dong Y, Tan Z, Li J. CEB: compositional evaluation benchmark for fairness in large language models. In: Proceedings of the 13th International Conference on Learning Representations. 2025, 1–42
- [183] Li A, Zhao J, Liang B, Gui L, Wang H, Zeng X, Liang X, Wong K F, Xu R. Mitigating biases of large language models in stance detection with counterfactual augmented calibration. In: Proceedings of 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies. 2025, 7075–7092
- [184] Martinez R D, Goriely Z, Caines A, Buttery P, Beinborn L. Mitigating frequency bias and anisotropy in language model pre-training with syntactic smoothing. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 5999–6011
- [185] Kibriya H, Khan W Z, Siddiq A, Khan M K. Privacy issues in large language models: a survey. Computers and Electrical Engineering, 2024, 120: 109698
- [186] Gao Z, Liu X, Lan Y, Yang Z. A brief survey on safety of large language models. CIT Journal of Computing and Information Technology, 2024, 32(1): 47–64
- [187] Fan M, Chen C, Wang C, Huang J. On the trustworthiness landscape of state-of-the-art generative models: a comprehensive survey. 2023, arXiv preprint arXiv: 2307.16680

- [188] Liu J, Gong R, Wei X, Dong Z, Cai J, Zhuang B. QLLM: accurate and efficient low-bitwidth quantization for large language models. In: Proceedings of the 12th International Conference on Learning Representations. 2024, 1–23
- [189] Guo Y, Kong F, Li X, Li H, Chen W, Tian X, Cai J, Zhang Y, Liu S. decoupleQ: towards 2-bit post-training uniform quantization via decoupling parameters into integer and floating points. 2024, arXiv preprint arXiv: 2404.12759
- [190] Zhao J, Zhang M, Wang M, Shang Y, Zhang K, Guan W, Wang Y, Zhang M. PTQ1.61: push the real limit of extremely low-bit post-training quantization methods for large language models. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2025, 4483–4502
- [191] Liu Z, Zhao C, Huang H, Chen S, Zhang J, Zhao J, Roy S, Jin L, Xiong Y, Shi Y, Xiao L, Tian Y, Soran B, Krishnamoorthi R, Blankevoort T, Chandra V. ParetoQ: scaling laws in extremely low-bit LLM quantization. 2025, arXiv preprint arXiv: 2502.02631
- [192] Jeon H, Kim Y, Kim J J. L4Q: parameter efficient quantization-aware fine-tuning on large language models. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2025, 2002–2024
- [193] Zhang G, Yue Y, Li Z, Yun S, Wan G, Wang K, Cheng D, Yu J X, Chen T. Cut the crap: an economical communication pipeline for LLM-based multi-agent systems. In: Proceedings of the 13th International Conference on Learning Representations. 2025, 1–40
- [194] Liu L, Qu Z, Chen Z, Ding Y, Xie Y. Transformer acceleration with dynamic sparse attention. 2021, arXiv preprint arXiv: 2110.11299
- [195] Liu R, Sun Y, Zhang M, Bai H, Yu X, Yu T, Yuan C, Hou L. Quantization hurts reasoning? An empirical study on quantized reasoning models. 2025, arXiv preprint arXiv: 2504.04823
- [196] Africa D D, Weiss Y, Buttery P, Martinez R D. Learning dynamics of meta-learning in small model pretraining. 2025, arXiv preprint arXiv: 2508.02189
- [197] Yang W, Yue X, Chaudhary V, Han X. Speculative thinking: enhancing small-model reasoning with large model guidance at inference time. 2025, arXiv preprint arXiv: 2504.12329
- [198] Sinani N, Salma S, Boutot P, Mustafiz S. Towards a domain-specific modelling environment for reinforcement learning. In: Proceedings of the 13th International Conference on Model-Based Software and Systems Engineering. 2025, 40–51
- [199] Cheng Z, Hao S, Liu T, Zhou F, Xie Y, Yao F, Bian Y, Zhuang Y, Dey N, Zha Y, Gu Y, Zhou K, Wang Y, Li Y, Fan R, She J, Gao C, Saparov A, Li H, Killian T W, Yurochkin M, Liu Z, Xing E P, Hu Z. Revisiting reinforcement learning for LLM reasoning from a cross-domain perspective. 2025, arXiv preprint arXiv: 2506.14965
- [200] Aralimatti R, Shakhadri S A G, KR K, Angadi K B. Fine-tuning small language models for domain-specific AI: an edge AI perspective. 2025, arXiv preprint arXiv: 2503.01933
- [201] Zhou H, Li X, Wang R, Cheng M, Zhou T, Hsieh C J. R1-zero's "aha moment" in visual reasoning on a 2B non-SFT model. 2025, arXiv preprint arXiv: 2503.05132
- [202] Shen H, Liu P, Li J, Fang C, Ma Y, Liao J, Shen Q, Zhang Z, Zhao K, Zhang Q, Xu R, Zhao T. VLM-R1: a stable and generalizable R1-style large vision-language model. 2025, arXiv preprint arXiv: 2504.07615
- [203] Liu Z, Sun Z, Zang Y, Dong X, Cao Y, Duan H, Lin D, Wang J. Visual-RFT: Visual reinforcement fine-tuning. 2025, arXiv preprint arXiv: 2503.01785
- [204] Zhang X, Liu J, Xiong Z, Huang Y, Xie G, Zhang R. Edge intelligence optimization for large language model inference with batching and quantization. In: Proceedings of 2024 IEEE Wireless Communications and Networking Conference. 2024, 1–6
- [205] Hu Y, Yuan Z, Gao W, Zhang S, Liu Y. An integer-only quantization framework for edge deployment of large language models. In: Proceedings of 2025 IEEE International Symposium on Circuits and Systems. 2025, 1–5
- [206] Kandala S V, Medaranga P, Varshney A. TinyLLM: a framework for training and deploying language models at the edge computers. 2024, arXiv preprint arXiv: 2412.15304



Chengyu WANG received his PhD degree in software engineering from East China Normal University, China in 2020. He is currently working on deep learning algorithms on various topics with Platform for AI, Alibaba Cloud Computing. He has authored or coauthored more than 90 research papers in top-tier international conferences and journals, such as ACL, KDD, WWW, SIGIR, CVPR, AAAI, TKDE, WSDM, CIKM, and EMNLP. His research interests include natural language processing, multi-modality learning, and large language models.



Taolin ZHANG received his PhD degree in software engineering from East China Normal University, China in 2023. He is currently a faculty member at the Hefei University of Technology, China. His research interests include natural language understanding and multi-modality applications such as vision-language model editing. He has published more than 20 research papers in top-tier international conferences, such as ACL, AAAI, EMNLP, CVPR, and CIKM.



Richang HONG is currently a Professor at Hefei University of Technology, China. He received his PhD degree from the University of Science and Technology of China, China in 2008. He has published over 100 papers in his research areas, which include multimedia question answering, video content analysis, and pattern recognition. He is a member of the ACM and was a recipient of the Best Paper Award at ACM Multimedia 2010.



Jun HUANG received his PhD degree in modern physics from the University of Science and Technology of China, China in 2008. He was an Associate Research Fellow with the China Academy of Engineering Physics, Mianyang, China. He currently leads a team for developing AI algorithms for the Platform for AI with Alibaba Cloud Computing, responsible for developing innovative algorithms and platforms. His research focuses on high-performance distributed implementation of deep learning algorithms and applying them to real applications.