

---

# Step-wise Adaptive Integration of Supervised Fine-tuning and Reinforcement Learning for Task-Specific LLMs

---

Jack Chen<sup>1,3,†</sup>, Fazhong Liu<sup>2,†</sup>, Naruto Liu<sup>1,3</sup>, Yuhan Luo<sup>2</sup>, Erqu Qin<sup>1,3</sup>, Harry Zheng<sup>1,3</sup>, Tian Dong<sup>2</sup>, Haojin Zhu<sup>2</sup>, Yan Meng<sup>2,\*</sup>, and Xiao Wang<sup>1,3,\*</sup>

<sup>1</sup>Shanghai Goku Technologies Limited

<sup>2</sup>Shanghai Jiao Tong University

<sup>3</sup>Shanghai AllMind Artificial Intelligence Technology Co., Ltd.

<sup>1,3</sup>{chenzhi2, liutian2, qinxiao2, zhengziwei2, wangxiao}@gokudata.com

<sup>2</sup>{liufazhong, gilerure, tian.dong, zhu-hj, yan\_meng}@sjtu.edu.cn

## Abstract

Large language models (LLMs) excel at mathematical reasoning and logical problem-solving. The current popular training paradigms primarily use supervised fine-tuning (SFT) and reinforcement learning (RL) to enhance the models' reasoning abilities. However, when using SFT or RL alone, there are respective challenges: SFT may suffer from overfitting, while RL is prone to mode collapse. The state-of-the-art methods have proposed hybrid training schemes. However, static switching faces challenges such as poor generalization across different tasks and high dependence on data quality. In response to these challenges, inspired by the *curriculum learning-quiz* mechanism in human reasoning cultivation, We propose SASR, a step-wise adaptive hybrid training framework that theoretically unifies SFT and RL and dynamically balances the two throughout optimization. SASR uses SFT for initial warm-up to establish basic reasoning skills, and then uses an adaptive dynamic adjustment algorithm based on gradient norm and divergence relative to the original distribution to seamlessly integrate SFT with the online RL method GRPO. By monitoring the training status of LLMs and adjusting the training process in sequence, SASR ensures a smooth transition between training schemes, maintaining core reasoning abilities while exploring different paths. Experimental results demonstrate that SASR outperforms SFT, RL, and static hybrid training methods.

## 1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in complex reasoning tasks, including mathematical problem-solving [1, 2], symbolic manipulation [3, 4], and multi-step logical inference [5–7]. These advancements are largely driven by sophisticated training paradigms that combine supervised fine-tuning (SFT) with reinforcement learning (RL). SFT provides models with high-quality, step-by-step reasoning demonstrations, often in the form of chain-of-thought (CoT) annotations, which help the model learn structured problem-solving strategies. Meanwhile, RL further refines these capabilities through reward-driven optimization, aligning model outputs with human preferences or task-specific objectives. This hybrid approach has become a cornerstone for state-

---

† These authors contributed equally.

\* Co-corresponding Authors.

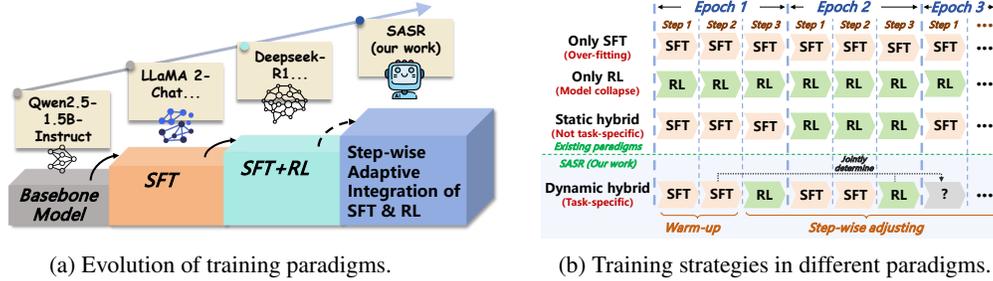
of-the-art (SOTA) models like GPT-4 [8], DeepSeek [9], and Claude [10], achieving unprecedented performance on benchmarks.

However, as the demand for training LLMs on specific tasks increases, where such tasks often lack large-scale, high-quality datasets but still require strong reasoning capabilities, the currently employed training paradigms face the following challenges. For primitive training paradigms (e.g., SFT and RL), SFT heavily relies on carefully crafted chain-of-thought (CoT) annotations and verified gold-standard answers, making it susceptible to overfitting [11–13]. In contrast, RL suffers from issues such as reward hacking [14] and mode collapse [15], which may cause LLMs to lose their reasoning capability during training. In response to above challenges, researchers have proposed Entropy bonus [16], Curriculum Learning [17], PTX Loss [18], etc. However, these methods act as auxiliary enhancements, addressing surface-level symptoms without structurally rethinking the underlying training paradigms. Thus, recent studies have begun to explore hybrid training paradigms that integrate supervised and reinforcement learning. For instance, the emerging DeepSeek model demonstrates strong reasoning capabilities through a training scheme that combines SFT and RL. However, the effectiveness of such two-stage frameworks remains uncertain in the absence of high-quality, carefully curated datasets. Moreover, their generalization across different training tasks has not been well established.

In this study, to address the above challenges, we propose an effective hybrid training method, SASR, inspired by the way human reasoning abilities are developed through structured learning followed by practice. In our framework, SFT plays the role of guided learning using reference materials, while RL serves as a form of quiz-like reinforcement that enhances generalization. SASR unifies these two stages and adaptively adjusts their proportions based on the model’s training dynamics. Just as students need to study sufficient reference materials, such as worked examples with solutions, before developing independent reasoning skills, SASR begins with a warm-up phase using SFT to establish basic reasoning capabilities. Following this phase, SASR continues training by combining SFT and reinforcement learning through GRPO. This stage mirrors how students, after reviewing worked examples, engage in solving new problems to enhance their generalization ability. However, removing reference materials entirely after the warm-up phase may cause the model to drift away from sound reasoning patterns. To prevent this, SASR dynamically adjusts the proportion of SFT and GRPO throughout training, guided by the model’s evolving state. Specifically, the ratio is updated at each training step by comparing the current gradient norms with those recorded during the warm-up phase.

Through this method, SASR can monitor the gradient characteristics and learning trends during the model’s learning process to dynamically guide training, while also achieving a smooth transition between the two training schemes, thus balancing the maintenance of basic reasoning capabilities and the exploration of multiple reasoning paths. In comparison, current hybrid training methods [9, 19, 20] use static training schemes and hard switching schemes. Static training schemes set fixed training steps for each training task and predefine the paradigm for each epoch. Meanwhile, hard-switching training schemes have no transition phase when switching from one paradigm to another, directly executing the switch. Such hybrid training schemes have deficiencies in the generalization of multi-task training, the smoothness of the switching process, and the resolution of the forget-stagnation problem. We conducted extensive experiments on two base models, DeepSeek and Qwen, across three standard datasets: GSM8K, MATH, and Knight-and-knives(KK). Our experiments covered mathematical calculations and logic reasoning-based question answering, demonstrating that SASR significantly improves performance and generalization ability compared to SFT, RL, and static hybrid training (For example, on mathematical reasoning specific tasks, the accuracy of SASR is improved by an average of 12.45% compared to the baseline, and by an average of 15.30% compared to RL. On complex datasets such as MATH and KK, it is improved by an average of 8.0% compared to static hybrid methods). Our contributions are as follows:

- We propose a novel adaptive dynamic training method, SASR, which theoretically connects SFT and RL for the first time and enhances LLM reasoning abilities through adaptive smooth hybrid training and demonstrating the superiority of dynamic training.
- Inspired by the human curriculum learning-quiz process, we designed a dynamic switching indicator based on the relationship between the training state during the warm-up phase and the gradient norm, which helps address the trade-off between forgetting and stagnation in LLM training.



(a) Evolution of training paradigms.

(b) Training strategies in different paradigms.

Figure 1: Comparison between our proposed SASR and existing training paradigms.

- Additionally, we deployed our method on multiple tasks involving mathematical reasoning and logical inference solving and conducted extensive experiments, demonstrating the superiority of SASR.

## 2 Related Work

### 2.1 Mathematical Reasoning & Logical Inference Solving

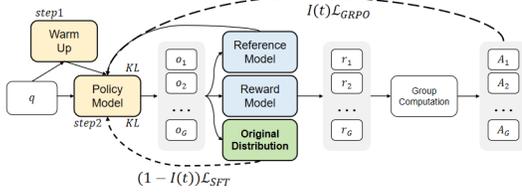
Enhancing the mathematical problem solving and logical reasoning capabilities of LLMs has become a key focus in recent research. Various methods such as Chain-of-Thought (CoT) prompting [6] with its variants tree-of-thought [21] and graph-of-thought [22], and self-consistency mechanisms [7], have demonstrated promising results on tasks like math word problems and arithmetic reasoning by encouraging coherent intermediate steps. However, these methods also face limitations. For instance, early-stage LLMs may generate unreliable explanations when performing few-shot textual reasoning [23]. Additionally, while combining self-supervised learning with reward-model-based reinforcement learning can guide LLMs in solving mathematical problems [24], concerns remain regarding reward hacking and the difficulty in capturing fine-grained logical inferences [9].

### 2.2 Training Paradigms

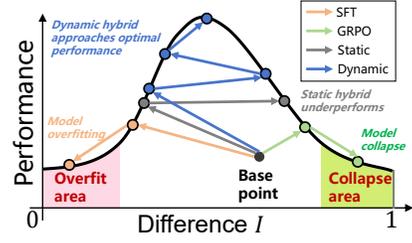
Supervised Fine-Tuning (SFT) is a foundational technique for adapting pre-trained language models to downstream tasks by training on high-quality demonstration data. Specifically, LLMs update their parameters by minimizing the discrepancy between predictions and ground-truth labels via gradient-based optimization. Due to its high efficiency and low cost, SFT has been widely adopted for fine-tuning LLMs in specialized areas such as mathematical reasoning [25–28]. However, SFT alone may struggle with open-ended tasks where optimal responses are less well-defined [29].

In 2022, OpenAI introduce ChatGPT [30], a large language dialogue model that catalyzed the development and adoption of a Reinforcement Learning from Human Feedback (RLHF) [31], as a novel training paradigm in the field of LLM training. Advanced Reinforcement Learning (RL) methods such as Proximal Policy Optimization (PPO) [32], and Group Relative Policy Optimization (GRPO) [33] have since been integrated into refinement of LLMs. GRPO, which aims to enhance policy optimization by leveraging group-relative advantages, can notably improve mathematical reasoning capabilities with less memory consumption [33]. However, it can still encounter challenges like reward hacking [34] and pattern collapse [33].

Given the limitations of utilizing only SFT or RL approaches, recent work explores hybrid approaches that combine both paradigms to enhance both instruction-following and reasoning. DeepSeek-R1 [9] exemplifies this trend by employing a novel training framework that combines SFT and GRPO. Similarly, Reinforced Fine-Tuning (ReFT) [19] demonstrates that RL-augmented fine-tuning can outperform pure SFT. These approaches highlight the potential of combining supervised learning with RL to unlock advanced reasoning in LLMs. However, the static hard-switching training scheme has considerable room for improvement in terms of dynamic task adaptation and progressive capability transfer.



(a) Visualizations of SASR’s architecture: warm-up in step1, and the training state is monitored through the condition function  $I(t)$  to adaptively adjust the training paradigm in step2.



(b) Advantages of dynamic training paradigm.

Figure 2: Visualizations of SASR’s architecture and theoretical analysis.

### 3 SASR: Step-wise Adaptive Integration of SFT and RL

In this section, we first present an overview of our proposed step-wise adaptive hybrid training framework, SASR, which is inspired by the development of human reasoning abilities through structured learning followed by practice. We then theoretically analyze the advantages of SASR over primitive training paradigms (i.e., SFT and GRPO) and static hybrid approaches, and further validate our insights through a series of case studies. Finally, we describe how SASR adaptively adjusts the ratio between SFT and RL based on training dynamics after the warm-up phase (see Algorithm 1).

#### 3.1 Overview of SASR

As illustrated in Figure 2a, SASR consists of two components: a warm-up phase based on SFT, and a subsequent hybrid training phase that integrates both SFT and GRPO. We formally define these two phases below. These definitions serve as the foundation for the theoretical analysis in the next subsection.

**Warm-up phase.** Since SASR is designed for training LLMs on task-specific scenarios where high-quality datasets are often unavailable, it begins with SFT on a small-scale dataset consisting of *(question, chain-of-thought)* pairs  $(x, e)$ , where  $x$  represents the input question token sequence and  $e$  denotes the corresponding chain-of-thought reasoning path that demonstrates the step-by-step solution process, in order to establish fundamental reasoning capabilities. The chain-of-thought is represented as a token sequence  $e = [a_1, \dots, a_L = \langle \text{eos} \rangle]$ , where each  $a_t$  corresponds to the  $t$ -th reasoning step token in the sequence, where each token is generated autoregressively:

$$a_t \sim \pi_{\theta}(\cdot | s_t), \quad s_{t+1} = [s_t, a_t] \quad (1)$$

where  $s_t$  represents the state (context) at step  $t$  containing all previously generated tokens, and  $\pi_{\theta}(\cdot | s_t)$  denotes the token generation probability distribution conditioned on  $s_t$ .

During the SFT phase, the optimization process aims to maximize the likelihood of ground-truth sequences, which is typically achieved by minimizing the negative log-likelihood (NLL) loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x,e) \sim \mathcal{D}} \left[ \sum_{t=1}^L \log \pi_{\theta}(a_t | s_t) \right], \quad (2)$$

where  $\mathcal{D}$  represents the training dataset distribution,  $\theta$  denotes the model parameters, and  $L$  is the length of the target sequence. The expectation is taken over both question and chain-of-thought pairs sampled from the dataset.

**Hybrid training phase.** After finishing the warm-up phase, SASR adopts a step-wise adaptive hybrid training employing both SFT and GRPO. In this phase, GRPO extends policy optimization through group-wise comparisons. For each input  $q$ , we sample  $G$  outputs from both current and old policies, then divide them into high-advantage ( $\mathcal{G}_+$ ) and low-advantage ( $\mathcal{G}_-$ ) groups based on their relative merits:

$$\mathcal{G}_{\pm} = \{o_{i,t} | \hat{A}_{i,t} \gtrless \text{median}(\{\hat{A}_{i,t}\})\}, \quad (3)$$

where  $o_{i,t}$  represents the  $t$ -th token of the  $i$ -th sampled output, and  $\hat{A}_{i,t}$  denotes the estimated advantage value measuring how much better the action is compared to the average at that state. The

objective combines advantage maximization with KL regularization to prevent excessive deviation:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \frac{1}{G} \sum_{i=1}^G \left[ \min \left( \frac{\pi_{\theta}}{\pi_{\theta_{\text{old}}}} \hat{A}_{i,t}, \text{clip} \left( \frac{\pi_{\theta}}{\pi_{\theta_{\text{old}}}}, 1 \pm \epsilon \right) \hat{A}_{i,t} \right) - \beta D_{\text{KL}}[\pi_{\theta} \parallel \pi_{\text{ref}}] \right], \quad (4)$$

where  $\pi_{\theta_{\text{old}}}$  is the previous policy before update,  $\pi_{\text{ref}}$  represents the reference policy (typically the initial SFT model),  $\epsilon$  controls the clipping range for policy updates, and  $\beta$  adjusts the strength of KL regularization. The ratio  $\frac{\pi_{\theta}}{\pi_{\theta_{\text{old}}}}$  measures how much the new policy deviates from the old one for each action.

To formally define the dynamic adaptive training algorithm, this paper introduces  $I(t)$  as the state function, which returns the training paradigm decision variable  $I(t)$  based on the current model’s training state  $t$ . Compared with traditional hybrid methods that use a fixed training paradigm within an epoch, SASR adopts a finer-grained training step  $s$  as a training unit, enabling more flexible adaptive adjustments. Finally, we define the overall loss function  $\mathcal{L}(\theta)$  of the dynamic training swithc framework in Equation 5.

$$\mathcal{L}(\theta) = \frac{1}{S} \sum_{s=1}^S [(1 - I(t)) \cdot \mathcal{L}_{\text{SFT}}(\theta) + I(t) \cdot \mathcal{L}_{\text{GRPO}}(\theta)] \quad (5)$$

### 3.2 Theoretical Analysis of SASR and Empirical Validation via Case Studies

In this subsection, to theoretically examine the advantages of SASR over existing training paradigms, we first establish the relationship between the gradient norm of the SFT loss and the Kullback–Leibler (KL) divergence. We then investigate how this relationship influences the reinforcement learning process. Specifically, we analyze the KL divergence between the model policy  $\pi_{\theta}$  and the data distribution  $\pi_{\text{data}}$ , and how this divergence impacts the gradient norm of SFT.

Initially, we define the SFT loss function as the cross-entropy loss:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x, y^*) \sim \mathcal{D}} \log \pi_{\theta}(y^* | x), \quad (6)$$

where  $\mathcal{D}$  represents the distribution of training data pairs  $(x, y^*)$  consisting of input questions  $x$  and their corresponding optimal reasoning paths  $y^*$ . The gradient of this loss with respect to model parameters  $\theta$  is:

$$\nabla_{\theta} \mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x, y^*)} \nabla_{\theta} \log \pi_{\theta}(y^* | x). \quad (7)$$

The KL divergence between the current policy  $\pi_{\theta}$  and the data distribution  $\pi_{\text{data}}$ , which measures how much the model’s behavior deviates from the original demonstrations, is given by:

$$D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{data}}) = \mathbb{E}_{(x, y^*)} \log \frac{\pi_{\theta}(y^* | x)}{\pi_{\text{data}}(y^* | x)}. \quad (8)$$

When  $\pi_{\text{data}}(y^* | x)$  remains fixed during training (as is typical with human demonstrations), its gradient simplifies to:

$$\nabla_{\theta} D_{\text{KL}} = \mathbb{E}_{(x, y^*)} \nabla_{\theta} \log \pi_{\theta}(y^* | x) = -\nabla_{\theta} \mathcal{L}_{\text{SFT}}. \quad (9)$$

establishing the fundamental relationship shown in Equation 10:

$$\|\nabla_{\theta} \mathcal{L}_{\text{SFT}}\| \propto D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{data}}). \quad (10)$$

This relationship indicates that the gradient norm of SFT is proportional to the KL divergence between the model policy and the data distribution. This implies that by minimizing the SFT loss, we are effectively reducing the discrepancy between the model policy and the data distribution, thereby aligning the model policy more closely with the data distribution. In GRPO, the KL divergence is applied to reduce the difference from the reference model, thereby avoiding mode collapse. However, recent works (such as DAPO) have demonstrated that when training long reasoning chain (CoT) models, the model distribution may significantly deviate from the initial model, rendering the constraint of the KL penalty term unnecessary. For small models, removing the KL loss can reduce the *learning tax* during training, enabling the model to more efficiently perform distribution migration and thus achieving better performance on specific tasks. However, reinforcement learning without

constraints is more prone to catastrophic forgetting during policy updates. To this end, we are the first to combine KL divergence (SFT) with GRPO, and by finely monitoring the training status, we ensure that the model can dynamically balance between free exploration and stable constraints during the GRPO training process, while fully utilizing the training data. In below, we analyze why SASR outperforms existing training paradigms as shown in Figure 2b.

**Avoiding SFT-induced overfitting.** Current research proved that pure SFT suffers from *overfitting* to limited CoT demonstrations [19]. GRPO’s exploration of diverse reasoning paths ( $G$  samples per input) breaks this limitation.

**Mitigating model collapse caused by RL.** Standard RL tends toward *mode collapse* and reward hacking. For cases where the LLM has a large gap from the original data distribution, our theory proves that SFT can help the model regressed to the data distribution required for training. Hybrid approach maintains proximity to the data distribution through the KL constraint:

$$D_{KL}(\pi_{\theta} \parallel \pi_{\text{data}}) \leq \frac{1}{\beta} \mathcal{L}_{\text{SFT}}(\theta_0) \quad (11)$$

This prevents degenerate solutions while allowing reward-guided exploration beyond SFT’s capability. In reinforcement learning, this relationship is particularly significant. The non-negativity of KL divergence ensures that the model policy does not deviate too far from the data distribution, thus preventing mode collapse during policy updates.

**Overcoming the suboptimality of static hybrid training.** Our gradient-based adaptation:

$$p_t = \left( \frac{\|\nabla \mathcal{L}_{\text{SFT}}^t\|}{\|\nabla \mathcal{L}_{\text{SFT}}^t\| + \gamma \|\nabla \mathcal{L}_{\text{SFT}}^0\|} \right) \quad (12)$$

where  $\sigma$  is the sigmoid function, provides smooth transitions between:

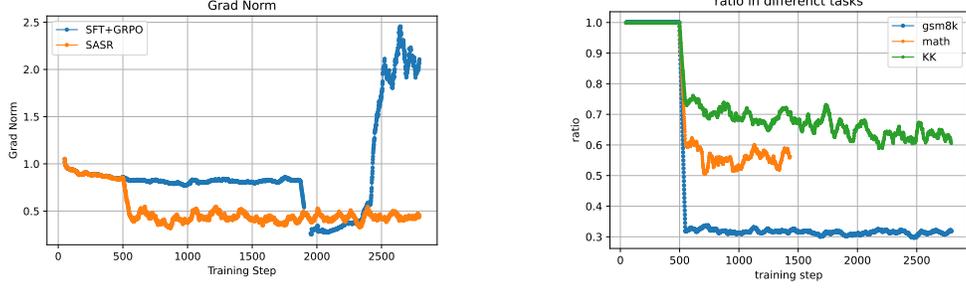
- *Exploration-dominant* phase ( $p_t \rightarrow 1$ ):  $\|\nabla \mathcal{L}_{\text{SFT}}\|$  is relatively large, meaning that LLM is currently far from the original distribution of the data and requires enhanced supervised learning.
- *Exploitation-dominant* phase ( $p_t \rightarrow 0$ ):  $\|\nabla \mathcal{L}_{\text{SFT}}\|$  is relatively small, meaning that LLM is currently close to the original distribution of the data and requires enhanced exploration.

Compared with the static switching mixed training paradigm, our SASR has three advantages. First, SASR based on the process of human cultivation of reasoning abilities, divides mixed training into multiple steps according to the training state, and carries out supervised learning and exploration simultaneously to solve the *catastrophic forgetting-stagnation trade-off* problem, preventing catastrophic forgetting while achieving gradual improvement. Meanwhile, compared with some current attempts at mixed training, we focus on the smoothing of the switching process, reducing the negative impact of the switching process on training (as shown in Figure 3a). Finally, as shown in Figure 3b, the probability trend changes differently on different datasets, which reflects that the optimal warm-up time and switching timing vary across different training tasks. Adopting a static switching training paradigm would face the problem of generalization.

### 3.3 Dynamic Ratio Selection

Our training framework combines SFT with GRPO through an adaptive mechanism. The key innovation is the dynamic balancing between these two approaches based on the model’s current performance metrics. For the SFT-RL dynamic hybrid training paradigm, how to set the dynamic switching indicator to effectively enhance model training performance is a key challenge. Apart from the naive rule-based training schedule, we have conducted experiments on important elements of the training state in the training process, including grad norm, steps, etc.

**SSR: rule-based training schedule.** To begin with, we propose a naive training paradigm, namely Step-wise integration of SFT and Reinforcement learning . For each training step, SSR simply alternates between SFT and GRPO in the training stage, so that each algorithm shares the same number of training steps. Although SSR have ensured a equal integration, it lacks a strategy to adjust the contribution of SFT and GRPO, which would probably cause overfitting or model collapse.



(a) Gradient norm under different paradigms.

(b) Ratios under various tasks.

Figure 3: Visualizations of cases study in theoretical analysis of SASR

---

**Algorithm 1:** The training procedure of SASR.

---

**Input:**  $\mathcal{D}_{train} = \{(x, e, \mathbf{y})\}$ : Tuples of  $(question, CoT, answer)$ ,  $W$ : number of warm-up steps,  
 $T$ : number of total steps,  $\pi_{\theta}^{(0)}$ : Initial policy,  $G$ : group size for GRPO

**Output:**  $\pi_{\theta}$ : Final policy

```

1  $\pi_{\theta} = \pi_{\theta}^{(0)}$ 
2 // Warm-up stage
3 for  $i \leftarrow 1$  to  $W$  do
4    $x, e, \mathbf{y} \sim \mathcal{D}_{train}$  // Sample mini-batch from  $\mathcal{D}_{train}$ 
5    $\theta = \text{OPTIMIZATION\_STEP}(\mathcal{L}_{SFT}(\theta))$  // Update policy parameters
6   if  $i == W$  then
7      $G_{warmup} \leftarrow \|\nabla_{\theta} \mathcal{L}_{SFT}(\theta)\|$  // Record final gradient norm
8 // Adaptive training stage
9 for  $t \leftarrow 1$  to  $T$  do
10  Compute  $p = \frac{G_{last-SFT}}{G_{last-SFT} + \gamma G_{warmup}}$ , Sample  $\alpha \sim \text{Uniform}(0, 1)$  // Compute adaptation probability
11  if  $\alpha < p$  then
12     $x, e, \mathbf{y} \sim \mathcal{D}_{train}$  // Sample mini-batch
13     $\theta = \text{OPTIMIZATION\_STEP}(\mathcal{L}_{SFT}(\theta))$  // Update policy
14     $G_{last-SFT} \leftarrow \|\nabla_{\theta} \mathcal{L}_{SFT}(\theta)\|$  // Update gradient norm
15  else
16     $x, -, \mathbf{y} \sim \mathcal{D}_{train}$  // Sample question-answer pair
17    Generate  $\{\hat{e}_i\}_{i=1}^G \sim \pi_{\theta}(x)$  // Generate  $G$  responses
18     $\{\hat{\mathbf{y}}_i\}_{i=1}^G \leftarrow \text{EXTRACT}(\{\hat{e}_i\})$  // Extract answers
19    Compute rewards  $\{R(\hat{\mathbf{y}}_i)\}_{i=1}^G$ 
20    Form groups  $\mathcal{G}_+, \mathcal{G}_-$  based on reward percentiles
21     $\theta = \text{OPTIMIZATION\_STEP}(\mathcal{L}_{GRPO}(\theta))$  // Update with GRPO objective
22 return  $\pi_{\theta}$ 

```

---

**SSR\_cosine: cosine training Schedule.** We improved this algorithm by applying a cosine decay schedule to the probability of switching to a SFT training step. Building upon the curriculum learning-quiz framework, we set SFT with a higher probability in the early stages of training. After the model learns the response format and logical reasoning, the model is further encouraged to switch to GRPO. The key limitation lies in its neglect of training states, which prevents it from responding to variations in model training states.

$$I(t) = 0.5 \left( 1 + \cos \left( \pi \frac{s}{S} \right) \right) (U - L) + L, \quad (13)$$

where  $S$  represents the current step,  $S$  represents the maximum number in training process, and  $U$  and  $L$  respectively represent the predefined upper and lower bounds parameters.

**The adaptive training schedule ultimately adopted by SASR.** Considering the generalization across different training tasks, the adaptive training algorithm (Algorithm 1) implements our theoretical framework through gradient norm. At each training step, the training algorithm dynamically monitors the benchmark gradient norm and the KL divergence of the current policy relative to the original data distribution at each training step, and uses the calculated training weight to make choices between SFT and GRPO. When the  $D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{data}})$  is relatively large compared to the benchmark gradient norm, it indicates that the training process is still in the “learning” phase, and SASR will increase the weight of SFT to ensure the model’s basic reasoning ability. When the  $D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{data}})$  is relatively small compared to the benchmark gradient norm, it means that the model has basically mastered the knowledge of the dataset, and the training process should enter the “quiz” phase. SASR will gradually increase the weight of GRPO to enhance the model’s thinking and multi-path exploration abilities. Under the ‘quiz’ mode, the algorithm generates G diverse responses through policy sampling. It then calculates relative advantages through grouped comparisons and updates the policy using the truncated GRPO objective (Equation 4).

## 4 Experimental Results

### 4.1 Experimental settings

**Dataset.** We gathered three representative datasets for the experiment: GSM8K [35], KK [36] and MATH [26]. The GSM8K dataset comprises elementary school-level mathematical problems that primarily require arithmetic computations. A distinctive characteristic of this dataset is that the final answers are constrained to nonnegative integers, with relatively lenient format requirements for response presentation. MATH consists of harder problems from math competitions and its answer contains mathematical formulas. In contrast, KK serves as a logic reasoning benchmark specifically designed to assess models’ deductive reasoning abilities. Unlike GSM8K and MATH, KK imposes highly specific output format requirements to ensure unambiguous and complete responses.

**Model Selection.** We selected the Qwen2.5-1.5B-Instruct model for KK, the Qwen2.5-0.5B-Instruct model for MATH and the DeepSeek-R1-Distill-Qwen-1.5B model for GSM8K as baseline models, respectively. Compared with other existing models of same scales, each models demonstrated reasonable capabilities in chain-of-thought reasoning and instruction following. Subsequent evaluation on the GSM8K dataset revealed that DeepSeek-R1-Distill-Qwen-1.5B achieved higher accuracy, thus establishing it as the baseline model for this dataset. For KK and MATH datasets, although both models exhibited comparable performance, Qwen2.5-Instruct series models demonstrated superior instruction-following capabilities, particularly in generating responses with better format. Therefore, is was chosen as the baseline model for the KK dataset. Our evaluation code base refers to the testing standards of the current mainstream models.

**Model Training:** Although both the MATH and GSM8K datasets provide solutions with CoT, evaluation reveals that the model performance declined after SFT. First of all, the pre-trained model may have already been trained on these datasets which could lead to overfitting. Furthermore, as Wadhwa et al. [37] found out, the quality of CoT could also affect SFT results. To address these limitations, we prefer to distill large language models to generete CoT instead of standard solutions. For GSM8K, we obtain high-quality CoT annotations from the gsm8k\_distilled dataset provided by Camel-AI. For MATH, we distill CoT from the Qwen2.5-Math-1.5B model and filter them according to the correctness of the answers. Consistent performance improvements are observed when we replace standard solutions with distilled CoT. Referring to the static hybrid paradigm adopted by the advanced LLM DeepSeek-R1 [9], in the static hybrid training, we switch training methods (SFT & RL) on a per-epoch basis, specifically conducting 2 epochs of SFT followed by 1 epoch of GRPO.

### 4.2 Experimental Results

#### 4.2.1 Performance on Mathematical Reasoning Tasks

We trained models on Mathematical Reasoning Tasks respectively to evaluate our method. The main results are shown in Table 1. In the commonly used benchmark tests of mathematical reasoning, the classical training paradigm SFT can enhance the ability of the model, but the improvement is limited. However, the sole use of RL (GRPO) has caused the degradation of the ability of the base model due to the problem of pattern collapse. Hybrid training can further enhance the reasoning ability of

Table 1: Answer accuracy of different models on the task-specific problems.

Model	$ \theta $	GSM8K	MATH	KK	Avg.
GPT-4o	200B	0.818	0.620	0.33	0.589
Deepseek-V3	671B	0.908	0.870	0.57	0.783
Baseline	1.5B/ 0.5B	0.638	0.146	0.03	0.271
SFT	1.5B/ 0.5B	0.752	0.212	0.28	0.414
GRPO	1.5B/ 0.5B	0.557	0.170	0.09	0.272
Static hybrid	1.5B/ 0.5B	<b>0.814</b>	0.160	0.33	0.326
SSR	1.5B/ 0.5B	0.779	0.196	0.38	0.452
SSR_cosine	1.5B/ 0.5B	0.795	0.204	0.39	0.463
SASR	1.5B/ 0.5B	0.803	<b>0.230</b>	<b>0.42</b>	<b>0.484</b>

Table 2: Answer accuracy of different models on the Knight-and-Knives problem with various level of difficulty.

Model	difficulty level							Avg.
	2 ppl	3 ppl	4 ppl	5 ppl	6 ppl	7 ppl	8 ppl	
Deepseek-Math-7B-Instruct	0.12	0.08	0.05	0.02	0.00	0.00	0.00	0.04
NuminaMath-7B-CoT	0.14	0.03	0.02	0.00	0.00	0.00	0.00	0.03
Qwen2.5-Base-7B	0.23	0.11	0.06	0.05	0.02	0.01	0.00	0.07
Qwen2.5-7B-Instruct-1M	0.23	0.25	0.13	0.09	0.03	0.04	0.02	0.11
GPT-4o	0.73	0.46	0.40	0.31	0.19	0.12	0.07	0.33
Deepseek-V3	0.90	0.71	0.68	0.57	0.39	0.37	0.37	0.57
Qwen2.5-1.5B-Instruct	0.13	0.05	0.02	0.00	0.00	0.00	0.00	0.03
SFT	0.65	0.42	0.38	0.24	0.14	0.08	0.07	0.28
GRPO	0.32	0.17	0.07	0.05	0.03	0.01	0.01	0.09
Static hybrid	0.71	0.47	0.37	0.25	<b>0.26</b>	0.10	0.12	0.33
SSR	<b>0.84</b>	0.54	0.47	0.34	0.19	<b>0.15</b>	0.12	0.38
SSR_cosine	0.74	0.60	0.49	0.36	0.22	0.14	0.15	0.39
SASR	0.83	<b>0.68</b>	<b>0.55</b>	<b>0.39</b>	0.19	0.11	<b>0.17</b>	<b>0.42</b>

the model. Among them, the performance of the designed training schedule is superior to that of direct hybrid (SSR). It is observed that SASR significantly enhanced the DeepSeek-R1-Distill-Qwen-1.5B model, increasing the accuracy rate from 63.8% to 80.3%, reaching a level close to GPT-4o. Due to carefully designed CoT distillation, SFT achieved remarkably improvement on the MATH dataset. The experimental results provide empirical evidence that SASR further exceeds SFT, with a measurable improvement of 1.8%.

#### 4.2.2 Performance on Logical Inference Tasks

For the KK dataset, the models are trained on 3 to 7-person KK problems and evaluated on 2 to 8-person KK problems. We exclude 2 person and 8 person problems from training datasets to observe whether the model could generalize to those two cases. We follow the base evaluation method of KK dataset [36] to decide whether the response of the model is accurate. The results of our methods and other baseline models are in Table 2. Our experiments suggested that our SASR has achieved better results compared to SFT, GRPO and static hybrid training paradims. Consequently, SASR has an average accuracy improvement of 9% compared to GPT-4o.

## 5 Conclusion and Limitations

In this work, we theoretically connect SFT and GRPO by modeling human reasoning cultivation, proposing a step-wise adaptive hybrid training framework for task-specific LLMs. SASR outperforms SFT, RL, and static hybrid methods on GSM8K, MATH, and KK datasets in reasoning tasks. By monitoring training status and step-level adjustment, SASR ensures smooth transitions between

schemes while maintaining core reasoning abilities. Besides, our SASR has certain limitations. The effectiveness of our method in combination with other reinforcement learning methods (such as PPO, DAPO) within a hybrid framework remains to be explored. Additionally, further research is needed in the broader application areas of LLMs (such as question answering).

## References

- [1] B. Romera-Paredes, M. Barekatin, A. Novikov, et al., Mathematical discoveries from program search with large language models, *Nature* 625 (2024) 468–475. doi:10.1038/s41586-023-06924-6.
- [2] J. Ahn, R. Verma, R. Lou, D. Liu, R. Zhang, W. Yin, Large language models for mathematical reasoning: Progresses and challenges, in: N. Falk, S. Papi, M. Zhang (Eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, Association for Computational Linguistics, St. Julian’s, Malta, 2024, pp. 225–237. URL: <https://aclanthology.org/2024.eacl-srw.17/>.
- [3] L. Pan, A. Albalak, X. Wang, W. Wang, Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning, in: H. Bouamor, J. Pino, K. Bali (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, Association for Computational Linguistics, Singapore, 2023, pp. 3806–3824. URL: <https://aclanthology.org/2023.findings-emnlp.248/>. doi:10.18653/v1/2023.findings-emnlp.248.
- [4] V. Gaur, N. Saunshi, Symbolic math reasoning with language models, in: *2022 IEEE MIT Undergraduate Research Technology Conference (URTC)*, 2022, pp. 1–5. doi:10.1109/URTC56832.2022.10002218.
- [5] C. Wang, Y. Deng, Z. Lyu, L. Zeng, J. He, S. Yan, B. An, Q\*: Improving multi-step reasoning for llms with deliberative planning, 2024. URL: <https://arxiv.org/abs/2406.14283>. arXiv:2406.14283.
- [6] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), *Advances in Neural Information Processing Systems*, volume 35, Curran Associates, Inc., 2022, pp. 24824–24837. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf).
- [7] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, in: *The Eleventh International Conference on Learning Representations*, 2023. URL: <https://openreview.net/forum?id=1PL1NIMMrw>.
- [8] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaf-tan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Nee-lakantan, R. Ngo, H. Noh, L. Ouyang, C. O’Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pan-tuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass,

- V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, B. Zoph, Gpt-4 technical report, 2024. URL: <https://arxiv.org/abs/2303.08774>. arXiv:2303.08774.
- [9] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, S. Ye, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Xu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, Z. Zhang, Deepseek-rl: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL: <https://arxiv.org/abs/2501.12948>. arXiv:2501.12948.
- [10] Anthropic, Claude 3.7 sonnet, 2025. URL: <https://www.anthropic.com/claude/sonnet>, accessed: 2025-02-26.
- [11] Z. Li, C. Chen, T. Xu, Z. Qin, J. Xiao, R. Sun, Z.-Q. Luo, Entropic distribution matching for supervised fine-tuning of LLMs: Less overfitting and better diversity, in: NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability, 2024. URL: <https://openreview.net/forum?id=dulz3WVhMR>.
- [12] T. Fu, D. Cai, L. Liu, S. Shi, R. Yan, Disperse-then-merge: Pushing the limits of instruction tuning via alignment tax reduction, 2024. URL: <https://arxiv.org/abs/2405.13432>. arXiv:2405.13432.
- [13] Z. Gekhman, G. Yona, R. Aharoni, M. Eyal, A. Feder, R. Reichart, J. Herzig, Does fine-tuning LLMs on new knowledge encourage hallucinations?, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 7765–7784. URL: <https://aclanthology.org/2024.emnlp-main.444/>. doi:10.18653/v1/2024.emnlp-main.444.
- [14] J. Hu, Y. Zhang, Q. Han, D. Jiang, X. Zhang, H.-Y. Shum, Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, arXiv preprint arXiv:2503.24290 (2025).
- [15] Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, M. Lin, Understanding rl-zero-like training: A critical perspective, arXiv preprint arXiv:2503.20783 (2025).
- [16] M. Luo, S. Tan, J. Wong, X. Shi, W. Y. Tang, M. Roongta, C. Cai, J. Luo, T. Zhang, L. E. Li, et al., Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl, Notion Blog (2025).

- [17] K. Team, A. Du, B. Gao, B. Xing, C. Jiang, C. Chen, C. Li, C. Xiao, C. Du, C. Liao, et al., Kimi k1. 5: Scaling reinforcement learning with llms, arXiv preprint arXiv:2501.12599 (2025).
- [18] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, *Advances in neural information processing systems* 35 (2022) 27730–27744.
- [19] T. Q. Luong, X. Zhang, Z. Jie, P. Sun, X. Jin, H. Li, Reft: Reasoning with reinforced fine-tuning, 2024. URL: <https://arxiv.org/abs/2401.08967>. arXiv:2401.08967.
- [20] A. Havrilla, Y. Du, S. C. Raparthy, C. Nalmpantis, J. Dwivedi-Yu, E. Hambro, S. Sukhbaatar, R. Raileanu, Teaching large language models to reason with reinforcement learning, in: *AI for Math Workshop @ ICML 2024*, 2024. URL: <https://openreview.net/forum?id=mjqoceuMnI>.
- [21] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, K. Narasimhan, Tree of thoughts: Deliberate problem solving with large language models, 2023. URL: <https://arxiv.org/abs/2305.10601>. arXiv:2305.10601.
- [22] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, T. Hoefler, Graph of thoughts: solving elaborate problems with large language models, in: *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'24/IAAI'24/EAAI'24*, AAAI Press, 2024. URL: <https://doi.org/10.1609/aaai.v38i16.29720>. doi:10.1609/aaai.v38i16.29720.
- [23] X. Ye, G. Durrett, The unreliability of explanations in few-shot prompting for textual reasoning, in: *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Curran Associates Inc., Red Hook, NY, USA, 2022.
- [24] J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving, I. Higgins, Solving math word problems with process- and outcome-based feedback, 2022. URL: <https://arxiv.org/abs/2211.14275>. arXiv:2211.14275.
- [25] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, J. Schulman, Training verifiers to solve math word problems, 2021. URL: <https://arxiv.org/abs/2110.14168>. arXiv:2110.14168.
- [26] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, J. Steinhardt, Measuring mathematical problem solving with the math dataset, *NeurIPS* (2021).
- [27] Z. Yuan, H. Yuan, C. Li, G. Dong, K. Lu, C. Tan, C. Zhou, J. Zhou, Scaling relationship on learning mathematical reasoning with large language models, 2023. arXiv:2308.01825.
- [28] Z. Gou, Z. Shao, Y. Gong, yelong shen, Y. Yang, M. Huang, N. Duan, W. Chen, ToRA: A tool-integrated reasoning agent for mathematical problem solving, in: *The Twelfth International Conference on Learning Representations*, 2024. URL: <https://openreview.net/forum?id=Ep0TtjVoap>.
- [29] H. Sun, M. van der Schaar, Inverse-rllignment: Inverse reinforcement learning from demonstrations for llm alignment, arXiv preprint arXiv:2405.15624 (2024).
- [30] OpenAI, Chatgpt: Optimizing language models for dialogue., 2022.
- [31] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, in: *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Curran Associates Inc., Red Hook, NY, USA, 2022.
- [32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017. URL: <https://arxiv.org/abs/1707.06347>. arXiv:1707.06347.

- [33] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, D. Guo, Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL: <https://arxiv.org/abs/2402.03300>. arXiv:2402.03300.
- [34] J. Wen, R. Zhong, A. Khan, E. Perez, J. Steinhardt, M. Huang, S. R. Bowman, H. He, S. Feng, Language models learn to mislead humans via rlhf, arXiv preprint arXiv:2409.12822 (2024).
- [35] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, J. Schulman, Training verifiers to solve math word problems, arXiv preprint arXiv:2110.14168 (2021).
- [36] T. Xie, Z. Gao, Q. Ren, H. Luo, Y. Hong, B. Dai, J. Zhou, K. Qiu, Z. Wu, C. Luo, Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning, 2025. URL: <https://arxiv.org/abs/2502.14768>. arXiv:2502.14768.
- [37] S. Wadhwa, S. Amir, B. C. Wallace, Investigating mysteries of cot-augmented distillation, arXiv preprint arXiv:2406.14511 (2024).