Decoupling Understanding from Reasoning via Problem Space Mapping for Small-Scale Model Reasoning

Li Wang[1], Changhao Zhang[2], Zengqi Xiu[1], Kai Lu[3,4], Xin Yu[1], Kui Zhang[1], Wenjun Wu[1,5,6]*

[1]Beihang University, Beijing, China
[2]UCL Hawkes Institute and Department of Medical Physics and Biomedical Engineering, University College London, UK
[3]State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[4]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[5]Hangzhou International Innovation Institute, Beihang University, Hangzhou, China
[6]Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing, China
{wangli_42, xiuzengqi, nlsdeyuxin, zhangkui, wwj09315}@buaa.edu.cn, changhao.zhang.24@ucl.ac.uk, lukai24@mails.ucas.ac.cn

Abstract

Despite recent advances in the reasoning capabilities of Large Language Models (LLMs), improving the reasoning ability of Small Language Models (SLMs, e.g., up to 1.5B parameters) remains challenging. A key obstacle lies in the complexity and variability of natural language: essentially equivalent problems often appear in diverse surface forms, often obscured by redundant or distracting details. This imposes a dual burden on SLMs: they must first extract the core problem from complex linguistic input, and then perform reasoning based on that understanding. The resulting vast and noisy problem space hinders optimization, particularly for models with limited capacity. To address this, we propose a new framework that decouples understanding from reasoning by mapping natural language problems into a canonical problem space-a semantically simplified yet expressive domain. This enables SLMs to focus on reasoning over standardized inputs, free from linguistic variability. Within this framework, we introduce DURIT (Decoupled Understanding from Reasoning via Iterative Training), a three-step algorithm that iteratively: (1) mapping natural language problems via reinforcement learning, (2) aligns reasoning trajectories through self-distillation, and (3) trains reasoning policies in the problem space. The mapper and reasoner are co-trained in an alternating loop throughout this process. Experiments show that DURIT substantially improves SLMs' performance on both in-domain and out-of-domain mathematical and logical reasoning tasks. Beyond improving reasoning capabilities, DURIT also improves the robustness of reasoning, validating decoupling understanding from reasoning as an effective strategy for strengthening SLMs.

Code — https://github.com/monster476/DURIT

Extended version — https://arxiv.org/pdf/2508.10019

Introduction

Large Language Models (LLMs) (Yang et al. 2025a) have demonstrated remarkable advances in reasoning capabilities (Bi et al. 2025; Luo et al. 2025a; Wen et al. 2024). However, most existing research has primarily focused on relatively large models (Guan et al. 2025; Li 2025; Shen et al. 2025), while the reasoning abilities of Small Language Models (SLMs, e.g., $\leq$ 1.5B) remain not fully explored. Despite their limited capacity, SLMs hold significant promise in edge-deployed scenarios and latency-sensitive applications due to their compact size and fast inference (Sun et al. 2020; Xu et al. 2024). Nevertheless, enhancing their reasoning capabilities remains a significant challenge due to their limited parameter capacity.

Recent efforts to improve LLM reasoning have focused on enhancing Chains of Thought (CoT) (Wei et al. 2022), using techniques like search-based reasoning (Li 2025; Guan et al. 2025) and error correction (Ma et al. 2025; Yang et al. 2025b). However, due to limited capacity, SLMs struggle to generate complex reasoning traces, making such approaches less effective. Knowledge Distillation (KD) is a common strategy for improving SLMs by transferring reasoning abilities from larger teacher LLMs via teacher-generated traces (e.g., CoT) or token-level supervision. However, distribution and capacity mismatches between teacher and student models pose challenges for both data and teacher selection. KD heavily depends on diverse, high-quality data (Gu et al. 2025): overly simple examples may cause overfitting to shallow patterns (Shumailov et al. 2024), while complex CoT traces may exceed the capacity of SLMs and hinder learning (Li et al. 2025). Teacher-student mismatches can further degrade performance (Cho and Hariharan 2019; Chen et al. 2025), underscoring the challenge of distilling high-quality reasoning into SLMs.

Unlike KD, reinforcement learning (RL) enables models to autonomously explore solutions, often yielding stronger generalization (Chu et al. 2025; Huan et al. 2025). The strong performance of DeepSeek-R1 (Shao et al. 2024) further underscores RL's potential in enhancing the reasoning capabilities of LLMs. However, SLMs face unique challenges: they must comprehend the semantic complexity of natural language problems and perform multi-step reason-

*Corresponding author.

ing despite limited capacity. The vast state space induced by natural language severely limits the efficiency of RL. Even superficial variations in problem phrasing can mislead models (Mirzadeh et al. 2024; Liu et al. 2025a), which often rely on shallow heuristics rather than genuine understanding. This suggests that models may fail to grasp the essence of the problems and are easily distracted by surface-level linguistic variations. In contrast, humans readily generalize across diverse surface forms once they grasp the essence of the problems. This contrast raises a key question: how can models acquire such essential understanding and generalize in a human-like manner? We address this by proposing a new perspective—rather than reasoning directly over the high-dimensional, noisy space of natural language, we first map problems into a lower-dimensional, standardized problem space. This transformation reduces spurious variability and constrains the search space by clustering essentially similar problems into more representative and canonical forms. As a result, it compresses the state space, highlights the essence of the problem, and alleviates the burden of superficial language understanding, thereby improving exploration efficiency. Crucially, our approach is orthogonal to existing CoT-based methods: problem space transformation acts as a front-end normalization layer, enabling more effective and robust downstream reasoning.

In this paper, we propose a general framework that maps natural language problems into a more abstract, low-dimensional, and semantically canonical space, effectively reducing the complexity of the original problem space. Within this space, models can learn and reason more efficiently. We instantiate this framework with a novel three-step alternating training algorithm: (1) a problem-space mapper is trained using RL and implicit templates to map natural language problems into standardized, low-dimensional forms; (2) self-distillation transfers this mapping capability into a SLM; and (3) a reasoning model is trained via RL to operate within the problem space. The mapper and the SLM are optimized in an alternating fashion, enabling iterative improvement in reasoning ability. To validate the effectiveness of DURIT, we conduct comprehensive empirical studies using models from the LLaMA (Grattafiori et al. 2024) and Qwen (Yang et al. 2025a) families, with parameter sizes ranging from 0.5B to 1.5B. Even when trained solely on the GSM8K (Cobbe et al. 2021) dataset, DURIT achieves significant gains across a range of in-domain and out-of-domain datasets, including those focused on mathematical and logical reasoning, and demonstrates strong generalization capabilities. Unlike traditional CoT-based approaches, DURIT introduces a paradigm that enhances reasoning by compressing the problem space. Our main contributions are summarized as follows:

- We propose a general framework that maps natural language problems into a standardized, low-dimensional space, reducing the effective state space and improving exploration and sample efficiency.
- We introduce DURIT, a three-step alternating training algorithm that decouples understanding from reasoning and progressively enhances the reasoning ability and robustness of SLMs through iterative co-training of a problem-space mapper and a reasoning model.
- Experiments show that DURIT yields substantial performance gains on both mathematical and logical reasoning tasks, across in-domain and out-of-domain settings, even with limited training data. In addition to improved accuracy, DURIT enhances reasoning robustness, indicating a deeper grasp of the problem's underlying essence and improved generalization across varied formulations.

Related Work

Prompt Optimization

Prompt optimization improves test-time performance by refining LLM inputs. Some approaches use paraphrasing (Yuan, Neubig, and Liu 2021; Deng et al. 2024), while others apply RL to explore prompt formats more effectively (Deng et al. 2022; Zhang et al. 2022). PRewrite (Kong et al. 2024) trains an LLM via PPO (Schulman et al. 2017) using response accuracy as reward, but incurs high inference cost due to LLM-based prompt generation. AbstRaL (Gao et al. 2025) improves reasoning robustness by abstracting problems into symbolic forms and delegating reasoning to external toolchains. Unlike prior work, our goal is to eliminate reliance on external tools and enable reasoning entirely within the natural language space. To achieve this, we train a problem space mapper via RL and distill its transformation behavior into the SLM, leading to improved reasoning performance and robustness.

Knowledge Distillation

Knowledge distillation transfers knowledge from a large teacher to a smaller student and can be divided into offline and online paradigms. Offline KD uses teacher-generated data. Std-CoT (Magister et al. 2023) fine-tunes students on CoT demonstrations, while NesyCD (Liao et al. 2025) distills general capabilities and incorporates external knowledge. Online KD requires the teacher to provide token-level supervision during inference. Vanilla-KD (Muralidharan et al. 2024) distills hidden states and output probabilities, BOND (Sessa et al. 2024) employs self-distillation based on the model's best responses, and STaR (Zelikman et al. 2024) fine-tunes on self-generated CoT traces with correct final answers to improve performance. In contrast to prior work, our method focuses on self-distillation to transfer knowledge internally, enabling the model to generalize its learned capabilities to unfamiliar tasks.

Reinforcement Learning for LLM Reasoning

Reinforcement Learning has proven effective in enhancing the capabilities of large language models. Reinforcement learning from human feedback (RLHF) (Bai et al. 2022; Ouyang et al. 2022) is now a standard approach for aligning model outputs with human preferences. Recent work such as DeepSeek-R1 (Shao et al. 2024) and Kimi K1.5 (Team et al. 2025) shows that techniques like GRPO can significantly boost reasoning ability, highlighting the promise of RL with verifiable rewards (RLVR). Building on this, many studies have proposed further refinements (Yu et al. 2025; Team

et al. 2025; Liu et al. 2025b). However, the vast and complex state space of natural language poses a major challenge to efficient exploration. To address this, we propose a problem space mapping that projects the original space into a lower-dimensional, more organized representation, thereby improving RL efficiency.

Decoupling Language Understanding From Reasoning

The inherent complexity of natural language presents a dual challenge for SLMs: interpreting subtle semantic nuances and performing reasoning, both constrained by limited model capacity. To address this, we propose a general framework that decouples understanding from reasoning. At its core is the notion of a problem space—a standardized, low-dimensional representation that abstracts surface variability while preserving essential semantics. By mapping fundamentally similar questions to nearby representations, the problem space reduces input complexity and offers a more interpretable and learning-efficient interface for downstream reasoning. As shown in Appendix, standardizing complex questions mitigates misinterpretation and improves reasoning accuracy. Formally, let $\mathcal{Q}$ denote the space of natural language questions, and let $\mathcal{P} \subset \mathcal{L}$ be a finite set of canonical forms drawn from the natural language space $\mathcal{L}$. We define a mapping $f : \mathcal{Q} \to \mathcal{P}$ that assigns each question $q \in \mathcal{Q}$ to a canonical representation $p = f(q) \in \mathcal{P}$. The construction of $\mathcal{P}$ and f is guided by the objective:

$$\begin{aligned} &\max_{f} \quad \mathbb{E}_{q\sim\mathcal{Q}}\left[\mathrm{Acc}(f(q);\theta)\right] \\ &\text{s.t.} \quad \begin{cases} \dim(\mathcal{P}) < \dim(\mathcal{Q}), \\ \mathrm{Dist}(f(q_1), f(q_2)) \le \epsilon, \quad \forall (q_1, q_2) \in \mathcal{S}, \end{cases} \end{aligned} \tag{1}$$

where the SLM with parameters θ has accuracy $\mathrm{Acc}(f(q);\theta)$ on the mapped input, and $\mathcal{S}$ is a set of fundamentally similar question pairs. The constraints encourage state compression and enforce a standardized structure within the problem space.

Based on this formulation, we propose a unified framework (Figure 1) that leverages a dedicated problem space mapper to project natural language questions into a standardized representation. By clustering fundamentally similar problems, this mapping reduces the exploration space and improves both sample and exploration efficiency during SLM training. As the model advances within this space, its ability to solve more complex problems increases, gradually shifting the underlying problem distribution. To adapt, our framework adopts an iterative training paradigm that alternates between updating the problem space mapper and refining the reasoning model, enabling their co-evolution. Reducing the problem space dimensionality enhances exploration and speeds up convergence. Follow (Cui et al. 2025), we model the problem using a bandit setting and apply a simplified Upper Confidence Bound, showing that the regret bound decreases with problem space dimensionality through the following theorem.

Theorem 1. *Let $\mathcal{Q}$ be a finite set of natural language problems, viewed as distinct states s, and let A denote the set of candidate responses. At each round $t \in \{1, \ldots, T\}$, a SLM observes a problem $s_t \in \mathcal{Q}$, selects an action $a_t \in A$, and receives a reward $r(s_t, a_t)$. Suppose learning is performed via a state-wise Upper Confidence Bound (UCB) algorithm in a contextual bandit setting. Then, in the state-independent worst case, the total regret after T rounds is bounded by*

$$R_T = O\left(\sqrt{|\mathcal{Q}| \cdot |A| \cdot T \cdot \ln T}\right),$$

where $|.|$ is the number of element of the set.

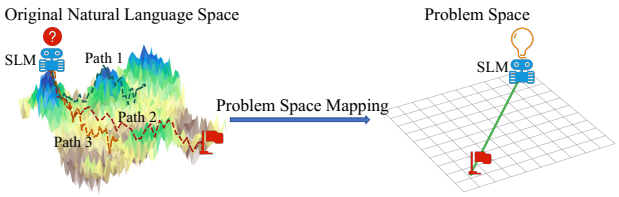


Figure 1: An illustration of our framework for decoupling understanding from reasoning. The original natural language space is complex and high-dimensional, making exploration difficult; mapping to a standardized, low-dimensional problem space compresses the state space and facilitates more efficient exploration.

The proof is provided in Appendix. While the UCB setting simplifies that of LLMs, Theorem 1 offers valuable insight into how reduced problem space dimensionality improves exploration. Specifically, mapping from $\mathcal{Q}$ to $\mathcal{P}$ compresses the space by a ratio $\alpha = |\mathcal{P}|/|\mathcal{Q}| < 1$, tightening the regret bound by a factor of $\sqrt{\alpha}$. This result supports our central motivation: leveraging standardized abstraction can make reasoning training more efficient for SLMs.

Methods

We propose Decoupled Understanding from Reasoning via Iterative Training (DURIT), which designs to enhance the reasoning ability of SLMs by decoupling problem understanding from reasoning. As illustrated in Figure 2, DURIT consists of three alternating steps: (1) **Problem Mapper Training:** a problem mapper M is trained via RL, guided by implicit templates, to map original natural language problems into problem space. (2) **Self-Distillation:** The transformation capability is internalized into reason the SLM R via self-distillation, enabling it to directly process complex problems without reliance on the external mapper M at inference time. (3) **RL Training:** the SLM R is further optimized using RL to improve its reasoning performance. The three steps are repeated iteratively, progressively strengthening the model's reasoning through alternating phases of understanding and reasoning. The complete pseudocode is provided in Appendix.

Step I: Problem Space Mapper Training

To decouple understanding from reasoning, a problem space mapper M is instantiated as an LLM that maps natural language questions into a standardized problem space. While

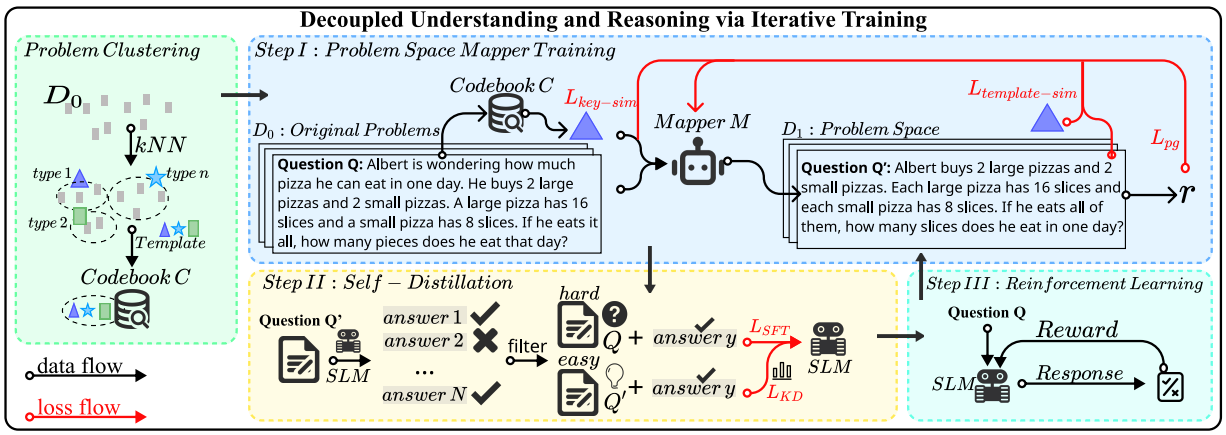


Figure 2: Framework of the DURIT Method. After KNN-based clustering, DURIT (1) compresses problems via implicit mapping, (2) distills this into the SLM, and (3) optimizes it through reinforcement learning with alternating co-training.

explicit templates enforce standardization, they are labor-intensive and may impede comprehension by SLMs. To balance standardization and flexibility, an implicit template mechanism is proposed, using a codebook to softly guide the output style of M. The mapping aims to 1) improve SLMs' understanding and 2) reduce the complexity of the problem space. To facilitate this, we cluster the training data based on fundamental question similarity using k-Nearest Neighbors (kNN) over representations $\mathbf{z}_i$ encoded from each question Q_i, its description, and answer via model M. As no ground-truth labels exist, we adopt GRPO (Shao et al. 2024) to optimize M based on the average correctness r_{acc} of frozen SLM's responses to mapped problem Q_i'. To prevent M from solving the problem directly, we apply a cheating penalty r_{cheating} if Q_i' includes solution-specific terms (e.g., keywords like "answer value") not present in Q_i. The total reward is:

$$r_i = r_{\text{acc}} + r_{\text{cheating}}. \tag{2}$$

However, RL alone cannot sufficiently enforce standardization. To simplify the problem space, implicit templates conditioned on cluster labels t_i are introduced. Specifically, a codebook C of n implicit template tokens $\{T_1, \ldots, T_n\}$ and corresponding query keys $\{k_1, \ldots, k_n\}$ is constructed, with both $\{T_i\}$ and $\{k_i\}$ randomly initialized parameters and optimized by loss. During training, for each problem Q_i, the template token T_{t_i} is selected and concatenated with the original input as $x_i = [Q_i; T_{t_i}]$, guiding M to produce the mapped question Q_i'. To encourage alignment between Q_i' and its assigned template, we define a template similarity loss based on the InfoNCE (He et al. 2020) objective:

$$\mathcal{L}_{\text{template-sim}} = -\log \frac{\exp\left(\frac{\langle \mathbf{z}_i, \mathbf{T}_i \rangle}{\tau}\right)}{\sum_{j=1}^{n} \exp\left(\frac{\langle \mathbf{z}_i, \mathbf{T}_j \rangle}{\tau}\right)}, \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, $\mathbf{z}_i$ is the normalized representation of the mapped problem Q_i' and τ is a temperature hyperparameter. At inference, with t_i unavailable, the best-matching implicit template is selected via cosine similarity between the input question embedding $\mathbf{q}_i$ (both $\mathbf{z}_i$ and $\mathbf{q}_i$ are approximated by averaging the word embeddings) and learned template query keys. A key similarity loss is introduced to facilitate key learning:

$$\mathcal{L}_{\text{key-sim}} = -\log \frac{\exp\left(\frac{\langle \mathbf{q}_i, \mathbf{k}_i \rangle}{\tau}\right)}{\sum_{j=1}^{n} \exp\left(\frac{\langle \mathbf{q}_i, \mathbf{k}_j \rangle}{\tau}\right)}, \tag{4}$$

Gradients from $\mathbf{k}_i$ are detached to prevent interference with the training of M, and only the template keys are updated. The overall loss function jointly optimizes the mapping policy and template-based constraints:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pg}} + \alpha_1 \mathcal{L}_{\text{key-sim}} + \alpha_2 \mathcal{L}_{\text{template-sim}}, \tag{5}$$

where $\mathcal{L}_{\text{pg}}$ denotes the policy gradient loss from GRPO, and α_1, α_2 are hyperparameters balancing different losses.

Step II: Self-Distillation Training

After training the problem space mapper M, its transformation capability is internalized into the SLM via self-distillation. Specifically, M transforms the original dataset $\mathcal{D}_0$ into a normalized form $\mathcal{D}_1 = \{Q_i' = M(Q_i) \mid Q_i \in \mathcal{D}_0\}$, where the mapped questions are designed to facilitate easier reasoning for the SLM. We then sample N responses using SLM on each $Q_i' \in \mathcal{D}_1$, and construct a filtered dataset: $\mathcal{D}_2 = \{(Q_i, Q_i', y_i) \mid Q_i \in \mathcal{D}_0,\ y_i = R(Q_i'),\ \texttt{answer}(y_i) = \texttt{True}\}$, where y_i denotes the model's response and $\texttt{answer}(y_i)$ evaluates its correctness. The core idea is to encourage SLM to replicate on Q_i the reasoning behavior it exhibits on Q_i'. To achieve this, we treat (Q_i', y_i) as a teacher pair and (Q_i, y_i) as the corresponding student pair. The SLM is trained using a combined loss of supervised fine-tuning L_{SFT} and KD L_{KD}:

$$p(x^k) = \frac{\exp(x^k/\tau)}{\sum_{j=1}^{|V|} \exp(x^j/\tau)}, \tag{6}$$

$$\mathcal{L}_i = \frac{1}{l} \sum_{k=1}^{l} \Big[(1-\lambda)\big(-\log p_s(x_i^k)\big) + \lambda \,\mathrm{KL}\big(p_t(x_i^k) \,\|\, p_s(x_i^k)\big) \Big]. \tag{7}$$

where l is the sequence length, x_i^k the k-th token of y_i, and p_s, p_t the student's and teacher's softmax outputs for prefix inputs Q_i and Q_i', respectively. The parameter λ balances the losses. This setup allows the student to internalize M without accessing it at inference.

Step III: Reinforcement Learning Training

After distilling the transformation capability into the SLM, the model is further trained via RL to explore and reason directly in the original problem space, leveraging its internalized understanding. Specifically, we fine-tune the SLM on the original training dataset $\mathcal{D}_0$ using the GRPO algorithm, with answer correctness serving as the reward signal. As the reasoning model improves, its ability to interpret and generalize evolves, potentially altering the optimal structure of the problem space. To accommodate this, the problem space mapper M and the reasoning model R are trained iteratively, enabling continual refinement of the problem space and progressive enhancement of reasoning capabilities.

Experiments

Datasets

All experiments train models solely on GSM8K (Cobbe et al. 2021). Evaluation considers both in-domain (IND) and out-of-domain (OOD) settings: GSM8K-Platinum (Vendrow et al. 2025) for IND, and MAWPS (Koncel-Kedziorski et al. 2016), SVAMP (Patel, Bhattamishra, and Goyal 2021), MATH500 (Hendrycks et al. 2021), and GAOKAO (Zhang et al. 2024) for OOD mathematical reasoning. Broader reasoning is evaluated on LogiQA (Liu et al. 2020). This setup enables systematic analysis of DURIT's impact on SLM reasoning across diverse domains.

Baselines and Metric

We compare DURIT against representative baselines in four categories: (1) CoT Distillation: including Std-CoT (Magister et al. 2023) and STaR (Zelikman et al. 2024), where N CoT responses per question are sampled and correct ones are filtered for fine-tuning; (2) Prompt Optimization: PRewrite (Kong et al. 2024) using RL to optimize prompts; (3) RL-Based Methods: GRPO (Shao et al. 2024); and (4) Knowledge Distillation: Vanilla-KD (Muralidharan et al. 2024), which requires online teacher LM inference. In our setup, the mapper model serves as the teacher. Following prior work (Sheng, Li, and Zeng 2025), answer accuracy is the primary metric. Further baseline details are in Appendix.

Implementations

To evaluate the generalization capability of DURIT, we test different base models, including recent strong instruction-following and reasoning-oriented models such as Qwen2.5-0.5B-Instruct (Yang et al. 2025a) and Llama3.2-1B-Instruct (Grattafiori et al. 2024). For the mapper model, we use Qwen2.5-3B-Instruct for the Qwen family, and Llama3.2-3B-Instruct for the Llama family to ensure architectural homogeneity. The codebook contains 32 implicit templates, with loss coefficients $\alpha_1 = 1\text{e}{-3}$ and $\alpha_2 = 1\text{e}{-2}$. Training is conducted in three steps: Step I runs for 1 epoch, Step II for 5 epochs, and Step III for 3 epochs. All experiments are carried out on 2 A100 GPUs with 40GB memory. For inference, we employ greedy decoding without vLLM (Kwon et al. 2023) acceleration. Additional implementation details, as well as more experimental results, parameter analyses, and training time comparisons, can be found in Appendix.

Main Results

As shown in Table 1, DURIT outperforms all baselines on both IND and OOD benchmarks. Remarkably, even when trained solely on the GSM8K dataset, DURIT consistently delivers substantial performance gains on all datasets. With just a single iteration, it achieves average accuracy improvements of 2.06% and 2.35% over the strongest baseline methods on Qwen2.5-0.5B-Instruct and Llama3.2-1B-Instruct, respectively. Importantly, DURIT achieves these gains without relying on external large models for CoT supervision. Instead, it fully exploits the model's own reasoning abilities to explore, adapt, and transfer prior knowledge. Remarkably, DURIT even outperforms distillation-based methods that depend on stronger teacher models such as DeepSeek-R1. As it operates entirely within the model itself, DURIT avoids additional API costs and infrastructure overhead, offering broad applicability and high cost-efficiency. DURIT's reasoning ability is further enhanced through a second iteration of training: even when continuing to use the GSM8K dataset, it yields an average accuracy gain of 0.36% on Qwen2.5-0.5B-Instruct and 0.69% on Llama3.2-1B-Instruct. Greater improvements are observed when using different datasets in the second iteration (see later Section), demonstrating DURIT's strong generalization across domains and its effectiveness in reducing the cognitive load of reasoning acquisition.

Reasoning Robustness Evaluation

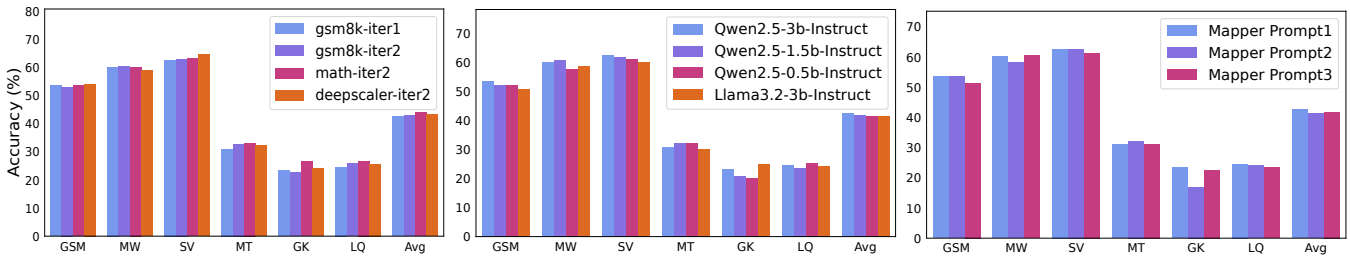
By projecting natural language questions into a more intrinsic and low-dimensional problem space, DURIT focuses on the essential semantics of the problem. This abstraction reduces variation from surface-level expressions and suppresses spurious or irrelevant cues, thereby enhancing the robustness of reasoning. To verify this claim, we evaluate on the GSM-Symbolic benchmark (Mirzadeh et al. 2024) using Qwen2.5-0.5B-Instruct and LLaMA3.2-1B-Instruct. As the original dataset contains only 100 examples and exhibits high variance, we follow (Gao et al. 2025; Liu et al. 2025a) and adopt the relative drop in average accuracy as a robustness metric. Results are reported in Table 2. DURIT attains an almost minimal relative drop in accuracy among all methods, indicating that its reasoning gains are accompanied by notably enhanced robustness. Results for additional model scales are provided in Appendix.

Performance Across Different Iterative Training Data

To evaluate the impact of iterative training datasets on DURIT, we conducted a second iteration using GSM8K,

Methods	In-Domain	Out-of-Domain					Average
	gsm8k-platinum	MAWPS	SVAMP	MATH500	GAOKAO	LogiQA	
# Qwen2.5-0.5B-Instruct based							
Base (Yang et al. 2025a)	45.74	54.23	54.67	27.80	18.55	14.44	35.91
CoT-Dis (Magister et al. 2023)	44.67	55.77	58.33	18.80	12.90	**30.41**	36.81
STaR (Zelikman et al. 2024)	51.86	57.88	61.67	29.60	18.55	23.50	40.51
GRPO (Shao et al. 2024)	51.03	58.08	61.00	27.40	21.77	22.73	40.34
PRewrite (Kong et al. 2024)	47.23	56.73	57.00	29.80	19.35	23.96	39.01
Vanilla-KD (Muralidharan et al. 2024)	49.30	57.69	61.67	30.4	**23.39**	20.74	40.53
DURIT (ours, iter=1)	**53.68**	60.19	62.67	31.00	**23.39**	24.58	42.59
DURIT (ours, iter=2)	53.10	**60.38**	**63.00**	**32.80**	22.58	25.81	**42.95**
# Llama3.2-1B-Instruct based							
Base (Grattafiori et al. 2024)	30.52	5.77	20.67	22.60	12.10	1.54	15.53
CoT-Dis (Magister et al. 2023)	48.06	56.92	57.67	24.60	12.90	**21.81**	36.99
STaR (Zelikman et al. 2024)	36.31	52.50	54.33	20.00	**16.94**	8.45	31.42
GRPO (Shao et al. 2024)	48.39	59.23	57.67	26.40	16.13	4.45	35.38
PRewrite (Kong et al. 2024)	35.81	41.34	46.00	18.80	12.10	3.53	26.26
Vanilla-KD (Muralidharan et al. 2024)	42.35	**64.23**	62.67	22.40	16.13	7.99	35.96
DURIT (ours, iter=1)	50.37	59.62	64.33	26.00	14.52	21.20	39.34
DURIT (ours, iter=2)	**52.36**	62.31	**66.00**	**27.60**	12.10	19.82	**40.03**

Table 1: Performance (%) of Qwen2.5-0.5B-Instruct and Llama3.2-1B-Instruct models across six representative benchmarks under various methods. The **bold** and underline indicate the best and second-best results, respectively.



(a) Impact of different iterative training data. (b) Impact of different mapper models. (c) Impact of different mapper prompt.

Figure 3: DURIT performance across six benchmarks with varying training data, mapper models, and prompts: GSM (GSM8K), MW (MAWPS), SV (SVAMP), MT (MATH500), GK (GAOKAO), LQ (LogiQA), and average (Avg).

MATH, and a filtered DeepScaleR (Luo et al. 2025b), following the first iteration on GSM8K. As Figure 3a shows, improvements are more pronounced when the second-iteration data differs from the first. This demonstrates DURIT's ability to decouple understanding from reasoning, effectively leveraging complementary data. Additionally, training with more diverse datasets consistently enhances overall performance and reasoning capabilities. Dataset details are provided in the Appendix.

Performance Across Different Mapper Models

To evaluate the impact of different mappers on DURIT, we fix the reasoning SLM as Qwen2.5-0.5B-Instruct and perform one iteration of DURIT updates with various mappers (Qwen2.5-3B/1.5B/0.5B-Instruct and Llama3.2-3B-Instruct) to assess both model scale and family effects. Given the relatively weak instruction-following of Qwen2.5-0.5B-Instruct, we warm-start it with 200 mapper data from Qwen2.5-3B-Instruct to improve initial alignment. Results (Figure 3b) show that mappers within the same family generally outperform others, and performance slightly improves with larger model size. Overall differences are marginal, demonstrating DURIT's robustness to mapper choice: even with a lightweight mapper like Qwen2.5-0.5B-Instruct, strong performance is achieved without relying on external larger models.

Performance Across Different Mapper Prompts

To assess the impact of mapper prompt design on DURIT, we test three prompt formulations (Appendix) using Qwen2.5-3B-Instruct as the mapper and Qwen2.5-0.5B-Instruct as the reasoning SLM, training each configuration

Method	Qwen-0.5B			Llama-1B		
	Orig	Symb	Δ%	Orig	Symb	Δ%
Base	46.0	41.6	**-9.6**	21.0	16.0	-23.7
CoT-Dis	47.0	40.6	-13.7	51.0	38.3	-24.9
STaR	51.0	41.0	-19.7	33.0	27.1	<u>-17.8</u>
GRPO	50.0	42.9	-14.3	44.0	35.8	-18.6
PRewrite	48.0	42.0	-12.0	39.0	21.9	-43.8
Vanilla-KD	51.0	42.2	-17.2	42.0	33.4	-20.5
DURIT	48.0	42.6	<u>-11.3</u>	44.0	40.8	**-7.2**

Table 2: Comparison of methods on Qwen2.5-0.5B-Instruct and Llama3.2-1B-Instruct. DURIT is trained with a single iteration. Orig: original test set; Symb: gsm-symbolic; Δ%: relative drop from Orig to Symb. **Bold** and <u>underline</u> indicate best and second-best results.

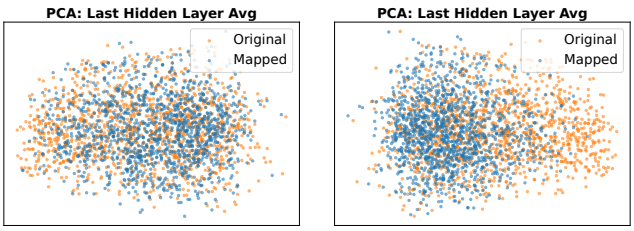
Variant	GSM	MW	SV	MT	GK	LQ	Avg.
DURIT	**53.68**	60.19	**62.67**	31.00	**23.39**	24.58	**42.59**
`w/o tem`	53.02	59.04	60.00	**31.40**	20.97	23.04	41.25
`w/o sd`	53.52	**60.58**	58.33	30.80	21.77	**24.88**	41.65
`w/o sft`	51.20	57.31	61.67	28.60	19.35	22.73	40.14
`w/o grpo`	49.30	57.69	61.67	30.40	**23.39**	20.74	40.53

Table 3: Ablation study of Qwen2.5-0.5B-Instruct on six benchmarks with a single DURIT iteration.

for one iteration. As Figure 3c shows, performance varies slightly: more explicit, standardized prompts generally perform better. All variants achieve strong performance, indicating DURIT's robustness to mapper prompt variations.

Ablation Studies

We conduct ablation studies using the Qwen2.5-0.5B-Instruct model to evaluate the contribution of each component in DURIT. For Step I, we assess the impact of removing the implicit template constraint component (`w/o tem`), while keeping the subsequent procedures in Step II and Step III of DURIT unchanged. For Step II, we examine the role of self-distillation by removing it and retaining only the SFT loss (`w/o sd`) and the necessity of SFT by removing sft loss(`w/o sft`). For Step III, we investigate the effect of removing GRPO training (`w/o grpo`). As shown in Table 3, ablating any single component leads to performance degradation. `w/o tem` disrupts the standardization of the problem space, resulting in less compact representations and lower exploration efficiency. `w/o sd` has minimal impact on in-domain performance but substantially impairs out-of-domain generalization, underscoring the role of self-distillation in reducing the comprehension burden and enhancing robustness. `w/o sft` may impose excessive disruption on the model's inherent reasoning mechanisms and simultaneously expose it to biased or incorrect reasoning patterns in the mapped questions, potentially resulting in further performance degradation. Finally, `w/o grpo` consistently reduces accuracy, confirming the necessity of RL to strengthen reasoning after self-distillation.



(a) PRewrite PCA (b) DURIT PCA

Figure 4: PCA Visualizations of Hidden Representations from Different Methods Using Qwen2.5-0.5B-Instruct.

Input	Original	PRewrite	DURIT
5NN Distance	75.16	73.68	**68.59**

Table 4: Average 5-NN distance of Qwen2.5-0.5B-Instruct final hidden states across different input questions. Lower values indicate tighter local clusters.

Towards Understanding the Effectiveness of Problem Space Mapping

To visualize how mapped questions are represented within the SLM, we analyze the final hidden layer representations of Qwen2.5-0.5B-Instruct on GSM8K-Platinum, using both the original inputs and their mapped versions produced by PRewrite and DURIT. We quantify the local compactness of these representations using the average k-nearest neighbor distance, as reported in Table 4. Additionally, we apply Principal Component Analysis (PCA) to project the high-dimensional hidden states into 2D for visualization, as shown in Figure 4. Compared to the original and PRewrite-mapped inputs, DURIT-mapped inputs yield significantly more compact clusters in the embedding space. This suggests that DURIT mapping helps remove redundant or irrelevant linguistic variability, effectively reducing the dimensionality of the problem space. As a result, the model may better capture the underlying essence of the problems, potentially leading to more efficient learning.

Conclusion

In this work, we propose a general problem space mapping framework, upon which we instantiate a concrete algorithm DURIT. DURIT consists of three key steps: (1) a problem space mapper trained via reinforcement learning with implicit template guidance, (2) self-distillation to internalize the mapping capability into a SLM, and (3) reasoning optimization of SLM within the reduced problem space. By alternating the training of the mapper and the SLM, DURIT enables iterative improvements in both reasoning capability and robustness. Empirical results demonstrate that DURIT consistently outperforms fine-tuned baselines, achieving substantial improvements in both in-domain and out-of-domain reasoning tasks, as well as enhanced robustness.

Acknowledgments

This work was supported by the National Science and Technology Major Project (Grant No. 2022ZD0117402), the National Natural Science Foundation of China (Grant No. 62441617), and the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing.

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bi, Z.; Han, K.; Liu, C.; Tang, Y.; and Wang, Y. 2025. Forest-of-Thought: Scaling Test-Time Compute for Enhancing LLM Reasoning. arXiv:2412.09078.

Chen, X.; Sun, Z.; Guo, W.; Zhang, M.; Chen, Y.; Sun, Y.; Su, H.; Pan, Y.; Klakow, D.; Li, W.; and Shen, X. 2025. Unveiling the Key Factors for Distilling Chain-of-Thought Reasoning. arXiv:2502.18001.

Cho, J. H.; and Hariharan, B. 2019. On the Efficacy of Knowledge Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Chu, T.; Zhai, Y.; Yang, J.; Tong, S.; Xie, S.; Schuurmans, D.; Le, Q. V.; Levine, S.; and Ma, Y. 2025. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. arXiv:2501.17161.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Cui, G.; Zhang, Y.; Chen, J.; Yuan, L.; Wang, Z.; Zuo, Y.; Li, H.; Fan, Y.; Chen, H.; Chen, W.; et al. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Deng, M.; Wang, J.; Hsieh, C.-P.; Wang, Y.; Guo, H.; Shu, T.; Song, M.; Xing, E. P.; and Hu, Z. 2022. RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning. arXiv:2205.12548.

Deng, Y.; Zhang, W.; Chen, Z.; and Gu, Q. 2024. Rephrase and Respond: Let Large Language Models Ask Better Questions for Themselves. arXiv:2311.04205.

Gao, S.; Bosselut, A.; Bengio, S.; and Abbe, E. 2025. AbstRaL: Augmenting LLMs' Reasoning by Reinforcing Abstract Thinking. arXiv:2506.07751.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gu, Y.; Zhou, H.; Meng, F.; Zhou, J.; and Huang, M. 2025. MiniPLM: Knowledge Distillation for Pre-Training Language Models. arXiv:2410.17215.

Guan, X.; Zhang, L. L.; Liu, Y.; Shang, N.; Sun, Y.; Zhu, Y.; Yang, F.; and Yang, M. 2025. rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking. arXiv:2501.04519.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv:2103.03874.

Huan, M.; Li, Y.; Zheng, T.; Xu, X.; Kim, S.; Du, M.; Poovendran, R.; Neubig, G.; and Yue, X. 2025. Does Math Reasoning Improve General LLM Capabilities? Understanding Transferability of LLM Reasoning. arXiv:2507.00432.

Koncel-Kedziorski, R.; Roy, S.; Amini, A.; Kushman, N.; and Hajishirzi, H. 2016. MAWPS: A Math Word Problem Repository. In Knight, K.; Nenkova, A.; and Rambow, O., eds., *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1152–1157. San Diego, California: Association for Computational Linguistics.

Kong, W.; Hombaiah, S.; Zhang, M.; Mei, Q.; and Bendersky, M. 2024. PRewrite: Prompt Rewriting with Reinforcement Learning. 594–601. Bangkok, Thailand: Association for Computational Linguistics.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J.; Zhang, H.; and Stoica, I. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 611–626.

Li, Y. 2025. Policy Guided Tree Search for Enhanced LLM Reasoning. arXiv:2502.06813.

Li, Y.; Yue, X.; Xu, Z.; Jiang, F.; Niu, L.; Lin, B. Y.; Ramasubramanian, B.; and Poovendran, R. 2025. Small Models Struggle to Learn from Strong Reasoners. arXiv:2502.12143.

Liao, H.; He, S.; Xu, Y.; Zhang, Y.; Liu, K.; and Zhao, J. 2025. Neural-Symbolic Collaborative Distillation: Advancing Small Language Models for Complex Reasoning Tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23): 24567–24575.

Liu, J.; Cui, L.; Liu, H.; Huang, D.; Wang, Y.; and Zhang, Y. 2020. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. arXiv:2007.08124.

Liu, J.; Huang, Z.; Dai, W.; Cheng, C.; Wu, J.; Sha, J.; Li, S.; Liu, Q.; Wang, S.; and Chen, E. 2025a. CogMath: Assessing LLMs' Authentic Mathematical Ability from a Human Cognitive Perspective. arXiv:2506.04481.

Liu, Z.; Chen, C.; Li, W.; Qi, P.; Pang, T.; Du, C.; Lee, W. S.; and Lin, M. 2025b. Understanding R1-Zero-Like Training: A Critical Perspective. arXiv:2503.20783.

Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; Tang, Y.; and Zhang, D. 2025a. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. arXiv:2308.09583.

Luo, M.; Tan, S.; Wong, J.; Shi, X.; Tang, W.; Roongta, M.; Cai, C.; Luo, J.; Zhang, T.; Li, E.; Popa, R. A.; and Stoica, I. 2025b. DeepScaleR: Surpassing O1-Preview with a 1.5B Model by Scaling RL. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2. Notion Blog.

Ma, R.; Wang, P.; Liu, C.; Liu, X.; Chen, J.; Zhang, B.; Zhou, X.; Du, N.; and Li, J. 2025. S^2R: Teaching LLMs to Self-verify and Self-correct via Reinforcement Learning. arXiv:2502.12853.

Magister, L. C.; Mallinson, J.; Adamek, J.; Malmi, E.; and Severyn, A. 2023. Teaching Small Language Models to Reason. 1773–1781. Toronto, Canada: Association for Computational Linguistics.

Mirzadeh, I.; Alizadeh, K.; Shahrokhi, H.; Tuzel, O.; Bengio, S.; and Farajtabar, M. 2024. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. arXiv:2410.05229.

Muralidharan, S.; Turuvekere Sreenivas, S.; Joshi, R.; Chochowski, M.; Patwary, M.; Shoeybi, M.; Catanzaro, B.; Kautz, J.; and Molchanov, P. 2024. Compact Language Models via Pruning and Knowledge Distillation. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 41076–41102. Curran Associates, Inc.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; 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. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 27730–27744. Curran Associates, Inc.

Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP Models really able to Solve Simple Math Word Problems? arXiv:2103.07191.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Sessa, P. G.; Dadashi, R.; Hussenot, L.; Ferret, J.; Vieillard, N.; Ramé, A.; Shariari, B.; Perrin, S.; Friesen, A.; Cideron, G.; et al. 2024. Bond: Aligning llms with best-of-n distillation. *arXiv preprint arXiv:2407.14622*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300.

Shen, M.; Zeng, G.; Qi, Z.; Hong, Z.-W.; Chen, Z.; Lu, W.; Wornell, G.; Das, S.; Cox, D.; and Gan, C. 2025. Satori: Reinforcement Learning with Chain-of-Action-Thought Enhances LLM Reasoning via Autoregressive Search. arXiv:2502.02508.

Sheng, Y.; Li, L.; and Zeng, D. D. 2025. Learning Theorem Rationale for Improving the Mathematical Reasoning Capability of Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(14): 15151–15159.

Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Papernot, N.; Anderson, R.; and Gal, Y. 2024. AI models collapse when trained on recursively generated data. *Nature*, 631(8022): 755–759.

Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. arXiv:2004.02984.

Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Vendrow, J.; Vendrow, E.; Beery, S.; and Madry, A. 2025. Do Large Language Model Benchmarks Test Reliability? arXiv:2502.03461.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wen, J.; Guan, J.; Wang, H.; Wu, W.; and Huang, M. 2024. Codeplan: Unlocking reasoning potential in large language models by scaling code-form planning. In *The Thirteenth International Conference on Learning Representations*.

Xu, J.; Li, Z.; Chen, W.; Wang, Q.; Gao, X.; Cai, Q.; and Ling, Z. 2024. On-Device Language Models: A Comprehensive Review. arXiv:2409.00088.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, L.; Yu, Z.; Zhang, T.; Xu, M.; Gonzalez, J. E.; Cui, B.; and Yan, S. 2025b. SuperCorrect: Advancing Small LLM Reasoning with Thought Template Distillation and Self-Correction. arXiv:2410.09008.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yuan, W.; Neubig, G.; and Liu, P. 2021. BARTScore: Evaluating Generated Text as Text Generation. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 27263–27277. Curran Associates, Inc.

Zelikman, E.; Wu, Y.; Mu, J.; and Goodman, N. D. 2024. Star: Self-taught reasoner bootstrapping reasoning with reasoning. In *Proc. the 36th International Conference on Neural Information Processing Systems*, volume 1126.

Zhang, T.; Wang, X.; Zhou, D.; Schuurmans, D.; and Gonzalez, J. E. 2022. TEMPERA: Test-Time Prompting via Reinforcement Learning. arXiv:2211.11890.

Zhang, X.; Li, C.; Zong, Y.; Ying, Z.; He, L.; and Qiu, X. 2024. Evaluating the Performance of Large Language Models on GAOKAO Benchmark. arXiv:2305.12474.