Beyond 'Aha!': Toward Systematic Meta-Abilities Alignment in Large Reasoning Models

Zhiyuan Hu[1*] **Yibo Wang**[2] **Hanze Dong**[3] **Yuhui Xu**[3]
Amrita Saha[3] **Caiming Xiong**[3] **Bryan Hooi**[1†] **Junnan Li**[3†]
[1] National University of Singapore [2] Tsinghua University
[3] Salesforce AI Research

Abstract

Large reasoning models (LRMs) already possess a latent capacity for long chain-of-thought reasoning. Prior work has shown that outcome-based reinforcement learning (RL) can incidentally elicit advanced reasoning behaviors such as self-correction, backtracking, and verification–phenomena often referred to as the model's "aha moment." However, the timing and consistency of these emergent behaviors remain unpredictable and uncontrollable, limiting the scalability and reliability of LRMs' reasoning capabilities. To address these limitations, we move beyond reliance on prompts and coincidental 'aha moments'. Instead, we explicitly align models with three meta-abilities—**deduction, induction, and abduction**, using automatically generated, self-verifiable tasks. Our three-stage pipeline (individual alignment, parameter-space merging, domain-specific reinforcement learning) boosts performance by over 10% relative to instruction-tuned baselines. Furthermore, domain-specific RL from the aligned checkpoint yields an additional 2% average gain in original performance ceiling across math, coding, and science benchmarks, demonstrating that explicit meta-ability alignment offers a scalable and dependable foundation for reasoning. Our code are released here.[3]

1 Introduction

Large reasoning models, including OpenAI-o1 [11], o3 [17], DeepSeek-R1 [8], Grok 3.5 [27], and Gemini 2.5 Pro [3], have demonstrated remarkable capabilities. These models excel at generating long Chain-of-Thought (CoT) [24] responses when tackling complex tasks and exhibit advanced, reflection-like reasoning behaviors. Recently, DeepSeek-R1 has shown that, starting from pretrained base or instruction-tuned models, pure reinforcement learning (RL) with rule-based rewards can spontaneously lead to the emergence of long CoT reasoning, self-correction, self-reflection, and other advanced behaviors, collectively referred to as the "aha moment". Other open-source works, such as SimpleRL-Zoo [31], tinyzero [18], and Logic-RL [28], which attempt to reproduce R1's performance and technical details, have also observed similar aha moments. These behaviors—such as self-correction, self-verification, and backtracking, signal the model's internal experience of strong reasoning ability.

However, relying solely on emergent behaviors is inherently unreliable and difficult to control. Models may fail to consistently manifest these advanced reasoning schemes, which limits both the predictability and scalability of LLM-based reasoning. To overcome this, we propose to explicitly align LLMs with three domain-general reasoning meta-abilities—deduction, induction, and abduction—drawn from Peirce's classical inference triad [19].

*Zhiyuan Hu. Email: zhiyuan_hu@u.nus.edu
†Corresponding Authors.
[3]https://github.com/zhiyuanhubj/Meta-Ability-Alignment

Preprint. Under review.

Deduction infers specific outcomes from general rules and hypotheses ($H + R \to O$), enabling rigorous prediction and verification. Induction abstracts rules from repeated co-occurrences ($H + O \to R$), facilitating pattern discovery and generalization. Abduction infers the most plausible explanation for surprising observations ($O + R \to H$), promoting creative and backward reasoning.

Together, they form a closed inferential loop for hypothesis generation, testing, and revision, mirroring the scientific method and supporting robust and interpretable reasoning.

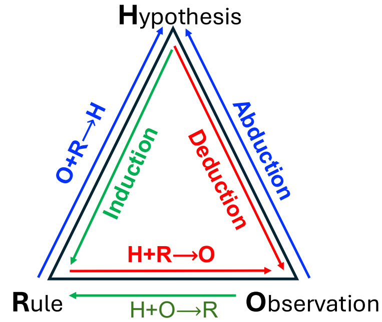

Figure 1: These meta-abilities form a unified reasoning framework.

To operationalize these meta-abilities, we construct a task suite with programmatically generated instances and automatic verifiability. Each task targets one core reasoning mode: Deduction: Propositional satisfiability tasks use rule sets R and candidate hypotheses H to test if all premises entail the observation O. Induction: Masked-sequence completion requires models to infer latent rules R from partial inputs H, O. Abduction: Inverse rule-graph search backchains from observed consequences O through a rule graph R to infer the minimal explanatory H. These tasks are constructed from synthetic distributions that lie out-of-distribution relative to common pretraining corpora, ensuring that performance improvements reflect genuine meta-ability acquisition rather than memorization or shortcut exploitation.

We observe that models aligned to individual meta-abilities make complementary errors. Aggregating their predictions raises overall accuracy by more than 10% relative to a vanilla instruction-tuned baseline. To incorporate the three competencies into a single network, we compared two approaches: training on a mixed task corpus and parameter-space model merging. Parameter-space merging improves average accuracy across math, coding, and science by 2% on a 7B model and 4% on a 32B model over the instruction-tuned baseline, demonstrating the strong generalization of merged meta-abilities.

Furthermore, to evaluate whether meta-ability alignment offers a stronger foundation for subsequent learning, we resumed domain-specific RL training from a checkpoint that have already been aligned and compared it with the same procedure applied to an instruction-tuned model. Starting from the meta-ability checkpoint raises the attainable performance ceiling: after identical continual domain-specific RL training, the model achieves an average gain of about 2% over its instruction-only counterpart. Our key contributions are as follows:

- **Task suite for meta-abilities.** We introduce a novel RL task suite aligned with three classical meta-abilities—deduction, induction, and abduction—each constructed to train and validate domain-general reasoning skills in large models.
- **Recipe for Reasoning Mastery.** We propose a three-stage recipe (1) independently align models to each meta-ability; (2) merge them via parameter-space integration; and (3) fine-tune with domain-specific RL. This leads to improved generalization and downstream task accuracy.
- **Upper-bound boost and scalability.** We empirically demonstrate that meta-ability alignment raises the performance ceiling: our 7B and 32B models show consistent gains over instruction-tuned baselines, across math, coding, and science benchmarks.

2 Related Work

RL-Driven Emergence of Reasoning Abilities Recent studies show that direct RL post-training can unlock long chain-of-thought reasoning beyond what supervised fine-tuning achieves [31, 29, 8]. *SimpleRL-Zoo* [31] proposes a zero-RL recipe using rule-based rewards, boosting math reasoning accuracy and inducing cognitive behaviors like self-verification across diverse base models. *DeepSeek-R1* [8] extends this idea to large-scale training; its public replications—*Light-R1* [25], *Open-R1* [4], and the minimal-cost *TinyZero* [18] —confirm that curriculum schedules, DPO warm-up, and carefully shaped length rewards together yield stronger logical accuracy while keeping compute affordable.

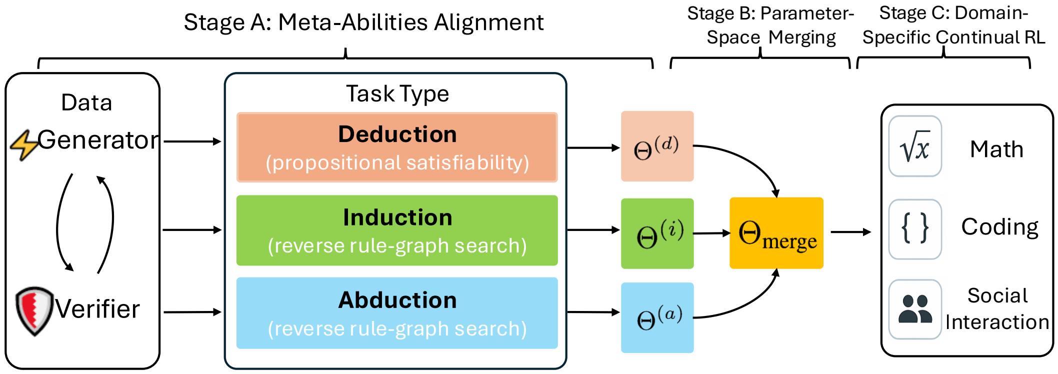

Figure 2: Overview of the three-stage pipeline: align deduction, induction, and abduction specialists, merge them in parameter space, and continually RL-adapt the unified model to downstream domains.

Complementary to these general pipelines, *Logic-RL* [28] applies rule-conditioned reinforcement learning to synthetic Knights-and-Knaves puzzles, Enabling transferable logical reasoning for math tasks. Together, these works establish RL as a viable path to large reasoning models.

Advanced reasoning ability In addition to enhancing long-chain reasoning through RL, recent work investigates specific reasoning skills such as self-correction, counterfactual inference, self-verification and others. Chen et al. [2] enhances overall reasoning performance by training models to reason both forward and backward, demonstrating that reverse-thinking objectives can improve forward reasoning as well. Kumar et al. [13] proposes SCoRe, an online RL method where a model iteratively critiques and improves its own answers, bridging the offline self-correction gap. Several recent studies also focus on equipping LLMs with self-verification and self-correction abilities, including ProCo [26], S^2R [14], and SETS [1].

Investigation of Aha-moment RL pipelines often show sudden accuracy jumps. Some concurrent analyses aim to uncover the internal "aha" moments that precede them. Gandhi et al. [5] introduces four reasoning behaviors as diagnostic tools to explain and engineer models' capacity for self-improvement under reinforcement learning. Yang et al. [30] shows that "aha moments" emerge through anthropomorphic language, uncertainty adjustment, and latent-space shifts, helping models avoid reasoning collapse and adapt to problem difficulty. Additionally, Zhou et al. [32] shows that similar emergence occurs in a 2B vision–language model without supervised warm-up.

3 Methodology

3.1 Task Design for Meta-Abilities Alignment

We design three reasoning tasks by by systematically instantiating the element triad (H, R, O) into a "given two, infer the third" framework, each corresponding to a distinct reasoning mode. In *deduction* ($H + R \Rightarrow O$), the model is given a set of logical rules R and a candidate truth assignment H as hypothesis, and must verify whether the overall observation O (i.e., all formulas being true) follows—formulated as a **propositional satisfiability** task. In *induction* ($H + O \Rightarrow R$), the model is provided with observable items O and incomplete inputs H (e.g., masked tokens or implied guesses), and must abstract the underlying generative rule R to correctly complete the sequence—framed as a **masked-sequence completion** task. In *abduction* ($O + R \Rightarrow H$), the model is given observations O and a rule graph R, and must trace backward to recover the minimal set of hidden assumptions H that can logically explain the conclusion—posed as a **reverse rule-graph search** task. This design follows a strict two-known-one-infer schema, clearly ensuring a clean separation of reasoning types, and reformulates all tasks into a unified (H,R,O) triplet format. This enables consistent, comparable, and complementary training signals, systematically equipping the model with a full range of meta-reasoning capabilities. As illustrated in Figure 2, each instance is produced by an automated *Generator* and screened by a *Verifier*, yielding large-scale, self-checked training data entirely free of manual annotation.

Deduction To instill deductive reasoning, the model proceeds from a conjectured hypothesis and specific conditions to derive rigorous predictions, thereby enabling systematic hypothesis proposal, empirical testing, and self-correction. We pose task that present a concise cluster of nested propositional clauses involving the standard Boolean operators NOT, AND, OR, IMPLIES, IFF, and XOR; the model must either return a satisfying truth assignment or report that the clauses are unsatisfiable. The task poses a combinatorial explosion of possible variable assignments, with tightly coupled logical formulas such that the value of one variable may indirectly constrain or determine the values of many others through chains of logical dependencies. This interdependence renders purely enumerative or heuristically guessed assignments overwhelmingly unlikely to satisfy all constraints. The only tractable approach is to begin with a provisional assignment, treat each formula as a logical premise, systematically derive its consequences, identify any resulting contradictions, and revise the assignment accordingly. This iterative loop—of hypothesis generation, logical consequence propagation, empirical inconsistency detection, and corrective refinement—directly instantiates the core structure of deductive reasoning.

Induction To develop inductive reasoning capabilities in models, we design a task for automatically-generated sequence with hidden terms. Each instance presents a series of elements following an undisclosed pattern (including numeric reasoning, symbolic patterns, and multi-step operation cycles) and requires identification of a missing element. This methodology specifically targets induction as the model must extract the underlying regularity governing the visible sequence and apply it to predict unseen values. Inductive learning through such structured sequences enhances the model's fundamental capabilities in abstraction and generalization, which are essential components for robust reasoning across domains.

Abduction To cultivate backward reasoning ability, we introduce a reverse rule-graph search task in which forward inference is deliberately obstructed while backward inference remains efficient. Each instance is formulated as a directed rule graph, with atoms as nodes and implications encoded as hyperedges from premise sets to conclusions. Observed facts activate source nodes, while target hypotheses correspond to sink nodes with unknown truth values. By inflating the branching factor in forward chaining, exhaustive exploration becomes computationally infeasible. In contrast, a backward strategy starts from a goal, hypothesizes minimal supporting premises, and verifies them against known facts. This approach can efficiently isolate relevant subgraphs. The design induces repeated cycles of goal-directed hypothesis formation, verification, and revision, thereby fostering the core mechanism of abductive reasoning.

3.2 Training Recipe for Reasoning

Figure 2 sketches how we transform the emergent "aha" moment into *controllable, composable* meta-abilities: we first carry out **Meta-Abilities Alignment**, independently training deduction, induction, and abduction specialists on synthetic diagnostics; we then fuse these specialists through **Parameter-Space Merging** to obtain a single checkpoint that retains their complementary strengths; finally, **Domain-Specific Reinforcement Learning Training** further refines the merged model on domain-specific data such as math, coding, and social dialogue.

3.2.1 Stage A: Meta-Abilities Alignment

We curate three synthetic but diagnostic datasets: *Deduction* (propositional satisfiability), *Induction* (masked-sequence completion), and *Abduction* (reverse rule-graph search). For a policy π_θ, we adopt the critic-free *REINFORCE++* loss [10], along with several improvements proposed in the Logic-RL framework [28].:

$$\mathcal{J}_{\text{R++}}(\theta) = \frac{1}{|O|}\sum_{i=1}^{|O|}\Big[r_i\,\hat{A}_i - \beta\, D_{\text{KL}}\big(\pi_\theta \| \pi_{\text{ref}}\big)\Big], \quad \hat{A}_i = \frac{r_i - \mu_r}{\sigma_r}, \tag{1}$$

where O is the response group, π_{ref} is the frozen instruction model, r_i the scalar reward, and $\{\mu_r, \sigma_r\}$ are group statistics.

Each reward r_i is computed via a rule-based scheme combining **Format Reward** and **Answer Reward**. The Format Reward checks structural compliance using regex-based rules: the model must place reasoning in <think> tags and the final answer in <answer> tags. A correct format yields $+1$, while any deviation gives -1. The Answer Reward evaluates correctness relative to the task-specific ground truth: a fully correct answer receives $+2$, and an unparseable or missing answer -2. Task-specific criteria guide this evaluation. A deduction (Propositional satisfiability) output is correct only if it satisfies all Boolean formulas; an induction (masked-sequence completion) prediction is valid if the predicted term fits the sequence pattern; and an abduction (inverse rule-graph search) answer is accepted when its premises form the minimal consistent causal path from evidence to target. The total reward is then normalized across the group to produce $\hat{A}_i$.

3.2.2 Stage B: Parameter-Space Merging for Meta-Ability Integration

To unify the strengths of models specialized in distinct meta-abilities, we adopt *parameter-space merging*, which enables: (i) a cost-efficient combination of complementary competencies without additional training, and (ii) a high-quality initialization for domain-specific fine-tuning in Stage C.

We denote the parameters of the deduction-, induction-, and abduction-aligned specialists as $\Theta^{(d)}$, $\Theta^{(i)}$, and $\Theta^{(a)}$, respectively. These models, trained separately on their respective meta-abilities, demonstrate highly complementary behaviors—aggregating their predictions. We construct the merged model Θ_{merge} by linearly interpolating the weights of the three specialists:

$$\Theta_{\text{merge}} = \lambda_d \Theta^{(d)} + \lambda_i \Theta^{(i)} + \lambda_a \Theta^{(a)} \tag{2}$$

We control the contribution of each specialist model via scalar weights λ_d, λ_i, and λ_a. These coefficients determine the relative influence of each meta-ability in the merged model. Notably, uniform weighting is not assumed—unequal allocation may better reflect the asymmetry in task difficulty or generalization benefit across reasoning modes. Optimal weights are selected empirically based on the performance.

3.2.3 Stage C: Domain-Specific Reinforcement Learning Training

To evaluate whether meta-ability alignment provides a stronger foundation for downstream learning, we apply reinforcement learning to the aligned checkpoints using domain-specific data, specifically math tasks. For a fair and controlled comparison with instruction-tuned baselines, we follow the experimental settings of SimpleRL-Zoo [31]. Specifically, we adopt a rule-based reward function that assigns +1 to correct completions and 0 otherwise, and use the Group Relative Policy Optimization (GRPO) objective [21] in place of more complex objectives such as REINFORCE++. These choices match SimpleRL-Zoo and help isolate the impact of initialization, ensuring performance gains arise from meta-ability alignment rather than optimization differences.

$$\mathcal{J}\text{GRPO}(\theta) = \frac{1}{\sum g|o_g|} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \min\Big[r_{i,t}\hat{A}i, ; \text{clip}(ri,t; 1{-}\epsilon, 1{+}\epsilon)\hat{A}i\Big] - \beta, D_{\text{KL}}(\pi_\theta|\pi_{\text{ref}}), \tag{3}$$

where $r_{i,t}$ is the per-token importance weight and π_{ref} is a fixed reference model used to regularize deviations.

4 Experimental Performance

4.1 Experimental Setup

Dataset For each meta-ability task we introduce a specific difficulty-controlling parameter. Thus, we generate multiple difficulty levels for every task and adopt the curriculum learning strategy that trains the model level by level from easy to hard. With this schedule, the 7B model converges by Level 2, and its reward does not improve further at higher levels, so we restrict training to Levels 1–2. The 32B model occasionally benefits from Level 3 but shows unstable reward curves. Therefore, we also use only the first two levels for it. We sample 200 instances per task per level for the 7B model and 2000 instances per task per level for the 32B model. For further domain-specific RL training, we adopt the same dataset as SimpleRL-Zoo [31].

Evaluation Setup To validate the generalization of these meta-ability, we select 7 benchmarks from math, coding and science domain. In math tasks, we utilize MATH-500 [9], the full AIME 1983–2024 corpus [23], the recent AMC 2023 [15] and AIME 2024 [16] sets and the Olympiad-level OmniMath subset [6] as the evaluation benchmark. LiveCodeBench is designed for code generation [12], and GPQA [20] is aimed for graduate-level science QA. For most benchmarks, we report pass@1 results using temperature 0.0 and top-p 1.0. While, for AIME 2024 and AMC 2023—which contain fewer problems—we report average accuracy (avg@32), computed over 32 samples per problem using temperature 1.0 and top-p 0.95.

Hyperparameter for Three Stage Training We utilize VERL [22] for meta-abilities alignment and continual reinforcement learning, and adopt MERGEKIT [7] for parameter-space merging to integrate distinct meta-abilities. The optimal weighting coefficients are set to $\lambda_d = 1.0$, $\lambda_i = 0.2$, and $\lambda_a = 0.1$.

Table 1: Performance of meta-ability–aligned models, merged ensembles, and oracle upper bounds on 7 benchmarks at both 7B and 32B parameter scales, illustrating consistent gains from scaling

Size	Model	Math					Code	Science	Average	
		Math500	AIME	AIME24 (Avg@32)	AMC23 (Avg@32)	Olympic	LCB	GPQA	Math	Overall
7B	Qwen2.5-7B-Instruct	73.0	22.4	10.7	50.8	37.3	25.7	27.2	38.8	35.3
	Deduction-Aligned	75.8	22.6	10.2	51.4	39.3	25.8	31.5	39.9(+1.1)	36.7(+1.4)
	Induction-Aligned	75.0	22.5	**11.8**	52.3	37.5	**27.0**	33.0	39.8(+1.0)	37.0(+1.7)
	Abduction-Aligned	75.2	22.7	11.4	49.1	38.4	26.8	31.9	39.4(+0.8)	36.5(+1.2)
	Merged Model	**77.8**	**22.9**	11.5	**52.3**	**40.4**	26.0	**33.5**	**41.0(+2.2)**	**37.8(+2.5)**
	Oracle Ensemble	85.5	32.1	18.0	67.1	46.7	–	–	49.9(+11.1)	–
32B	Qwen2.5-32B-Instruct	79.8	31.2	15.3	62.7	45.6	39.5	38.0	46.9	44.6
	Deduction-Aligned	83.8	**36.9**	19.4	68.5	**47.4**	**42.1**	**38.6**	51.2(+4.3)	**48.1(+3.5)**
	Induction-Aligned	82.6	34.8	18.6	66.2	45.8	41.7	38.4	49.6(+2.7)	46.9(+2.3)
	Abduction-Aligned	83.8	33.3	17.5	65.9	46.2	41.1	38.6	49.3(+2.4)	46.6(+2.0)
	Merged Model	**84.2**	36.0	**19.7**	**69.5**	46.9	41.8	38.4	**51.3(+4.4)**	**48.1(+3.5)**
	Oracle Ensemble	88.1	43.2	27.3	76.8	53.0	–	–	57.7 (+10.8)	–

Table 2: Comparison of 7B- and 32B-scale baseline instruction models and our continual domain-specific RL variants across math, code, and science benchmarks. *Domain-RL-Ins* denotes continual domain-specific RL starting from instruction model; *Domain-RL-Meta* applies the same RL schedule but from a meta-ability–merged initialization, yielding a higher attainable performance ceiling.

Size	Model	Math					Code	Science	Average	
		Math500	AIME	AIME24 (Avg@32)	AMC23 (Avg@32)	Olympic	LCB	GPQA	Math	Overall
7B	Qwen2.5-7B-Instruct	73.0	22.4	10.7	50.8	37.3	**25.7**	27.2	38.8	35.3
	Domain-RL-Ins	78.2	23.6	11.9	53.2	39.3	25.1	33.0	41.2(+2.4)	37.8(+2.5)
	Domain-RL-Meta	**78.8**	**27.7**	**12.6**	**54.7**	**41.0**	25.4	**33.1**	**43.0(+4.2)**	**39.0(+3.7)**
32B	Qwen2.5-32B-Instruct	79.8	31.2	15.3	62.7	45.6	37.5	38.0	46.9	44.6
	Domain-RL-Ins	83.0	36.5	18.6	67.5	46.1	**41.8**	38.2	50.3(+3.4)	47.4(+2.8)
	Domain-RL-Meta	**84.6**	**38.2**	**19.8**	**70.4**	**48.7**	41.6	**38.6**	**52.3(+5.4)**	**48.8(+4.2)**

4.2 Out-of-Domain Generalization of Meta-Abilities

Table 1 shows that meta-ability alignment, trained solely on synthetic diagnostic tasks, already transfers to seven unseen benchmarks. At the 7B scale, the induction-aligned model provides the largest average improvement, lifting the mean score by 1.7%, whereas the deduction-aligned model yields the largest single math task gain with a 2.8% increase on MATH500. Integrating the three meta-abilities in the **Merged** model further raises the overall score by 2.5%, confirming that the abilities combine constructively. An **Oracle Ensemble** that marks a problem correct if *any* aligned model succeeds boosts the Math-average by 11.1% , underlining the strong complementarity still to be tapped by better fusion methods.

Scaling to the 32B model amplifies the pattern: each aligned model surpasses the *Qwen2.5-32B-Instruct* baseline, yielding a mean gain of 3.1% on the Math-overall metric and 2.6% on the overall average. These results further confirm that the proposed alignment strategy instills reasoning skills that generalize reliably beyond their training domain. Additionally, the **Merged** checkpoint improves the overall average by 3.5%, with a standout 4.4% gain on the Math-average. A full Oracle Ensemble run shows an additional 10.8% lift in the math average, indicating that the three reasoning modes remain highly complementary even at larger scale.

4.3 Scalable Gains from Meta-Abilities Alignment

Table 2 shows that resuming domain-specific RL from a meta-ability–merged checkpoint (Domain-RL-Meta) consistently outperforms the same schedule applied to an instruction-tuned model (Domain-RL-Ins). At 7B, math rises from 38.8 (instruction baseline) to 41.2 with Domain-RL-Ins and to 43.0 with Domain-RL-Meta, while the overall average climbs from 35.3→37.8→39.0; the largest jumps appear on the compositional AIME (+5.3) and Olympic (+3.7) subsets, with code and science remaining steady. Scaling to 32B amplifies the pattern: math improves 46.9→50.3→52.3 and the overall average 44.6→47.4→48.8, translating to relative math gains of 7% (Domain-RL-Ins) and 12% (Domain-RL-Meta). Thus, embedding general deductive, inductive, and abductive routines before task-specific RL raises the attainable performance ceiling, and the advantage widens as model capacity grows.

5 Conclusion

This work demonstrates that large reasoning models need not rely on unpredictable ‘aha moments’ to acquire advanced problem-solving skills. By explicitly aligning deduction, induction, and abduction through automatically generated, self-verifiable tasks, we create specialist agents whose complementary strengths can be merged—without extra compute—into a single checkpoint that outperforms an instruction-tuned baseline by more than 10% on purpose-built diagnostics and up to 2% on seven diverse math, code, and science benchmarks. When this meta-ability-aligned model is used as the starting point for domain-specific reinforcement learning, it lifts the attainable performance ceiling by a further 4% and widens the gap as model capacity scales from 7B to 32B parameters. These results confirm that systematic, modular training of fundamental reasoning modes provides a controllable and scalable foundation for downstream capability composition. Future work will explore richer fusion strategies, extend the task suite to multimodal settings, and investigate how explicit meta-ability control can improve interpretability and safety in large-scale reasoning systems.

References

[1] Jiefeng Chen, Jie Ren, Xinyun Chen, Chengrun Yang, Ruoxi Sun, and SercanÖ. Arık. Sets: Leveraging self-verification and self-correction for improved test-time scaling. *arXiv preprint arXiv:2501.19306*, 2025. URL `https://arxiv.org/abs/2501.19306`.

[2] Justin Chih-Yao Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, et al. Reverse thinking makes llms stronger reasoners. *arXiv preprint arXiv:2411.19865*, 2024.

[3] Google DeepMind. Gemini 2.5 pro: The latest gemini multimodal model. `https://deepmind.google/technologies/gemini/`, May 2024. Accessed: 2025-05-15.

[4] Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL `https://github.com/huggingface/open-r1`.

[5] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

[6] Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omnimath: A universal olympiad level mathematical benchmark for large language models. In *Proc. of ICLR*, 2025.

[7] Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee's MergeKit: A toolkit for merging large language models. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.36. URL `https://aclanthology.org/2024.emnlp-industry.36`.

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[9] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[10] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

[11] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[12] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida I. Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[13] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

[14] Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. S^2r: Teaching llms to self-verify and self-correct via reinforcement learning. *arXiv preprint arXiv:2502.12853*, 2025. URL `https://arxiv.org/abs/2502.12853`.

[15] Mathematical Association of America. American Mathematics Competitions (AMC) 2023, 2023.

[16] Mathematical Association of America. American Invitational Mathematics Examination (AIME) 2024, 2024.

[17] OpenAI. Openai o3 and o4-mini system card. 2025.

[18] Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025. Accessed: 2025-01-24.

[19] Charles Sanders Peirce. Collected papers of charles sanders peirce, volume 2: Elements of logic. pages Chapter 7, especially paragraphs 619–645. Harvard University Press, 1931. Peirce's formulation of deduction, induction, and abduction as distinct forms of inference.

[20] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[21] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[22] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[23] Hemish Veeraboina. Aime problem set 1983-2024, 2023. URL `https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024`.

[24] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[25] Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*, 2025.

[26] Zhenyu Wu, Qingkai Zeng, Zhihan Zhang, Zhaoxuan Tan, Chao Shen, and Meng Jiang. Large language models can self-correct with key condition verification. *arXiv preprint arXiv:2405.14092*, 2024. URL `https://arxiv.org/abs/2405.14092`. EMNLP 2024.

[27] xAI. Grok 3.5: Advanced reasoning ai model by xai. `https://grok.x.ai/`, February 2025. Accessed: 2025-05-15.

[28] Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025.

[29] Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

[30] Shu Yang, Junchao Wu, Xin Chen, Yunze Xiao, Xinyi Yang, Derek F Wong, and Di Wang. Understanding aha moments: from external observations to internal mechanisms. *arXiv preprint arXiv:2504.02956*, 2025.

[31] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

[32] Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. R1-zero's" aha moment" in visual reasoning on a 2b non-sft model. *arXiv preprint arXiv:2503.05132*, 2025.