ALLSPARK: WEAK TO STRONG TRANSFER VIA ALTERNATING CHAIN OF THOUGHT

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent progress in frontier models has renewed interest in large-scale reinforcement learning (RL), but the cost of generating large-model rollouts makes even testing RL recipes expensive. We ask whether reasoning improvements learned by a small, weak model can benefit a larger, stronger model without using the strong model's rollouts during training. We introduce Allspark, a training and inference framework for weak-to-strong transfer through alternating chains of thought. A weak teacher is trained alongside a frozen copy of the same model; the two alternate reasoning segments, and the frozen model produces the final answer. At inference time, a stronger student replaces the frozen training partner, while both models remain fixed. Because they communicate through text, the teacher can steer students from different model families and with different tokenizers. We study Allspark at two scales: controlled Qwen experiments across math and reasoning, and larger-scale Inkling experiments on ARC-AGI-2. The Inkling experiments show accuracy gains in within-family and cross-family settings, including transfer to Kimi and Nemotron, with benefits that vary across inference settings. These findings motivate reusing a trained weak teacher across strong students and examining the resulting accuracy–token tradeoff.

1 INTRODUCTION

> 弟子不必不如师，师不必贤于弟子。
>
> *A student is not necessarily inferior to the teacher, nor is a teacher necessarily superior to the student.*
>
> —Han Yu, *Shishuo* ("On Teachers")

Recent progress in frontier models has renewed interest in large-scale reinforcement learning (RL) (Kimi Team, 2026; DeepSeek-AI, 2026; LLM-Core Xiaomi, 2026; Microsoft AI, 2026; OpenAI, 2026; Anthropic, 2026). Yet on testing an RL recipe on a large model requires repeated sampling and optimization, making even preliminary experimentation expensive. Rollout generation, particularly for long reasoning traces, can be a major bottleneck (Zhou et al., 2025). This cost limits how many training recipes researchers can test and makes it difficult for laboratories with modest compute budgets to explore improvements for frontier models. A compelling alternative is to develop reasoning capabilities in smaller models and transfer their benefits to stronger ones.

One approach is to train specialist teachers and consolidate their capabilities into a single student through on-policy distillation (OPD), as in Kimi K3 and DeepSeek-V4 (Kimi Team, 2026; DeepSeek-AI, 2026). However, OPD still samples from and updates the student, retaining the expense of generating large-model rollouts when the student is large. Direct token-level distribution matching also assumes aligned tokenizations, although recent methods explicitly support OPD across different tokenizers (Niu et al., 2026). Our aim is to remove the need for the target strong student during training and transfer the learned behavior through text at inference time. This leads to the central question:

> *Can a weak teacher trained with RL improve a strong student without using any rollouts from that strong student during training?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We propose Allspark, a training and inference framework for weak-to-strong transfer through alternating chains of thought (Figure 1). During training, two copies of a small model take turns extending a shared reasoning trace. One copy, the *weak teacher*, is trained with RL; the other remains frozen and produces the final answer. At inference time, we replace the frozen training partner with a stronger student and reuse the trained teacher without further optimization. Teacher and student continue each other's reasoning, and the student produces the final answer. The transfer interface is the shared text, allowing the same teacher to steer students with different architectures and tokenizers. The teacher therefore need not be a better standalone problem solver: its role is to contribute reasoning that helps the student reach a better answer.

We study Allspark at two complementary scales. The controlled Qwen3 study uses 1.7B and 4B models (Yang et al., 2025) across math and reasoning to examine teacher-training choices and transfer to larger students under fixed inference limits. The larger-scale study trains an Inkling-Small teacher (Thinking Machines Lab, 2026b) and evaluates it on 96 development problems drawn from the public training split of ARC-AGI-2 (Chollet et al., 2025). This study examines transfer within the Inkling family and across families to Kimi K2.6 (Moonshot AI, 2026a) and Nemotron 3 Ultra (NVIDIA, 2026). It shows accuracy gains in various settings, providing evidence that a trained teacher can be reused across strong students. Using Inkling's thinking-effort controls (Thinking Machines Lab, n.d.b), we also examine how teacher steering changes accuracy as reasoning effort varies (Figure 4). We report accuracy against retained completion tokens, making the scope of the token–performance comparison explicit.

Our contributions are threefold:

- We introduce Allspark, which trains a weak teacher using only small-model rollouts and reuses it to steer frozen strong students through alternating chains of thought.
- We study teacher training and transfer at two scales: controlled Qwen experiments across two domains, and Inkling experiments with evidence of transfer within and across model families.
- We analyze how teacher steering interacts with test-time scaling, identifying settings in which it improves accuracy and characterizing the accompanying retained-token tradeoff.

2 Background and Related Works

Weak supervision for a trainable strong student. Weak-to-strong generalization studies whether a stronger model can benefit from a weaker supervisor without inheriting its performance ceiling. Burns et al. (2023) finetune strong models on weak labels, and Yang et al. (2024) extend this approach to selected weak-model reasoning traces. These methods update the strong model, but supervised finetuning need not generate its rollouts. On-policy distillation instead gives teacher feedback on sequences sampled from the student, reducing the mismatch between training prefixes and those encountered at inference (Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025). Cross-tokenizer variants can provide token-level supervision even when teacher and student vocabularies differ (Niu et al., 2026).

Recent weak-to-strong distillation methods often transfer a change learned by the weak model rather than its entire policy. Direct-OPD uses the post-RL versus pre-RL log-ratio as an implicit reward on strong-student rollouts (Feng et al., 2026). W2S-OPD adds a weak-model logit contrast to the strong student's base logits and distills the resulting proxy teacher (Yu et al., 2026). OPRD uses the weak model's policy change to rescale the strong student's verifier-driven policy gradient (Park et al., 2026). These approaches obtain strong-student rollouts and update its parameters during transfer. They address a different resource setting from training a weak teacher once and applying it to frozen strong students.

Guidance for a frozen strong student. Other methods keep the strong model fixed and alter how it is guided at inference. Weak-to-strong search steers a frozen large model with likelihood differences from tuned and untuned weak models (Zhou et al., 2024). CoWeST trains a weak collaborator using preferences derived from its interactions with a strong model; the strong model then refines the weak model's drafts and background information (Jiao et al., 2025). Co-LLM learns when to defer to a

frozen assistant (Shen et al., 2024), while weak critics can prompt a strong model to revise an answer (Jin et al., 2026).

3 ALLSPARK: TRANSFER WITHOUT STRONG-MODEL TRAINING ROLLOUTS

Allspark asks whether a weak model can learn to improve another model's reasoning without using the target strong model during training. We train a weak teacher alongside a frozen *weak student*, initialized from the same base model. They alternate reasoning chunks, and the weak student produces the final answer. Its task reward updates only the teacher's reasoning contributions. At inference, we freeze the teacher and replace the weak student with a frozen strong student, which now continues the shared reasoning and produces the answer (Figure 1). The teacher is thus trained entirely with weak-model rollouts and transferred through text, without updating or training on rollouts from the strong student.

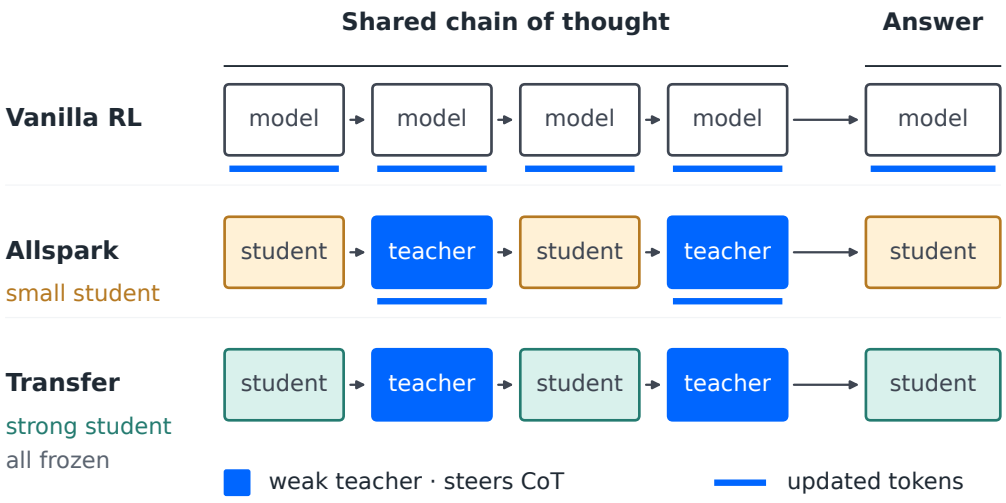


Figure 1: Vanilla RL updates the full reasoning and answer. Allspark updates only the weak teacher's reasoning beside a frozen small student. Transfer freezes the teacher and replaces the student with a stronger model. Blocks show authorship; underlines mark direct token loss.

3.1 ALTERNATING REASONING

Let π be the weak teacher and ρ the student. Given a problem x, both models extend the same reasoning prefix h, initially containing x. The student starts, and the two models take turns generating a chunk z:

$$z \sim \begin{cases} \rho(\cdot \mid h), & \text{student turn,} \\ \pi(\cdot \mid h), & \text{teacher turn.} \end{cases} \tag{1}$$

Each chunk is appended to h before the next turn. When reasoning stops, the student produces the final answer $y \sim \rho(\cdot \mid h)$. Let $R(x, y) \in \{0, 1\}$ indicate whether the final answer is correct.

3.2 TRAINING THE WEAK TEACHER

During training, ρ is a frozen copy of the teacher's initial model. Only π is optimized, with the task objective

$$\max_{\pi} \; J(\pi), \qquad J(\pi) = \mathbb{E}_{\pi,\rho}\big[R(x, y)\big], \qquad \rho \text{ fixed.} \tag{2}$$

The expectation averages over training problems and the alternating rollouts above. Only sampled teacher reasoning tokens receive direct loss; student tokens and inserted boundary markers are masked out. The experiment-specific RL losses and training rewards are given in Appendices A and B.1.

The usefulness of a teacher chunk can be expressed as its continuation value:

$$Q(h, z) = \mathbb{E}\big[R(x, y) \mid h, z\big]. \tag{3}$$

Here the teacher has emitted z after prefix h; the expectation covers the remaining rollout under the current π and ρ. A useful chunk therefore need not solve the problem by itself: its value comes from the student's subsequent continuation.

3.3 Transfer across model size and family

At transfer, freeze the selected teacher π and replace ρ with a stronger student. The same alternating procedure applies, with both models fixed. Transfer changes the student's reasoning context through teacher-generated text; it does not update the student's weights. Each writer renders and tokenizes the shared text using its own model interface, so the models need not share a vocabulary.

4 Experiment

We evaluate Allspark at two scales: controlled experiments with Qwen models and larger-scale experiments with Inkling models. The Qwen experiments examine weak-to-strong transfer through targeted ablations, while the Inkling experiments demonstrate improvements in the accuracy–token tradeoff and weak-to-strong transfer across model families.

4.1 Controlled Qwen experiments

Setup. The controlled study uses two Qwen3-1.7B (Yang et al., 2025) training runs: ordinary RL and Allspark, each with seed 0. Both train on the same mixed-domain pool, with equal numbers of accepted prompts from each domain per update. Allspark trains alongside a frozen Qwen3-1.7B student. The resulting 1.7B teachers are reused with frozen Qwen3-1.7B and Qwen3-4B students, without additional training. Table 1 summarizes the data; Appendix A gives the loss, training recipe, and checkpoint selection.

Table 1: Qwen training and development data for the two reported domains. Both methods share the screened RL pool. Development questions are disjoint from this pool but were used for checkpoint selection and subsequent evaluation. All samples come from the datasets' original training splits.

Domain	Dataset	RL pool	Development
Math	MATH (Hendrycks et al., 2021)	142	128
Reasoning	LogiQA 2.0 (Liu et al., 2023)	162	128
Total		304	256

Counts are distinct prompts. Each domain starts with 512 training candidates; initial-policy screening retains prompts with 1–7 correct answers in eight samples. Each update uses two accepted prompts per domain, despite the unequal pool sizes.

Controls and evaluation. Table 2 compares each frozen student alone and with the checkpoint-94 Allspark teacher, an ordinary-RL teacher, and (for the 4B student) the untrained teacher. We also include the ordinary-RL-trained 1.7B policy on its own. Every paired condition uses a Qwen3-1.7B teacher; the student remains frozen and produces the final answer. We report the math and reasoning portions of the development panel in Table 1, with 128 questions per domain and one sampled answer per question. We use final-answer accuracy for both domains. Within each comparison, we match the evaluation questions, student decoding settings, and reasoning and answer limits.

Table 2: Qwen development accuracy (%) on math and reasoning, with 128 questions per domain. The Allspark teacher is checkpoint 94; Ordinary RL is the separately trained Qwen3-1.7B policy. Dashes mark ongoing evaluations.

Student	Teacher	Math	Reasoning
Qwen3-1.7B	None	81.3	73.4
	Ordinary RL	88.2	71.8
	Allspark	89.8	75.0
Qwen3-1.7B (RL-tuned)	None	89.8	70.3
Qwen3-4B	None	**96.9**	81.2
	Untrained	93.0	79.7
	Ordinary RL	89.8	79.6
	Allspark	96.1	**82.8**

Takeaway. Table 2 provides a sign of life for weak-to-strong capability transfer from an Allspark-trained teacher across math and reasoning tasks. In the 4B ablation, using either an untrained or an ordinary-RL weak teacher to steer the stronger student lowers accuracy in both domains. Using the Allspark-trained teacher instead yields approximately neutral transfer in math and positive transfer in reasoning.

Case studies. Figure 2 shows two math recoveries and two reasoning answer changes on paired Qwen3-4B development questions. In the octagon problem (M3), the student alone settles on $1053/2$; in the alternating trace, the teacher returns to corner subtraction and the student next computes the correct area, 567. In the peg-board problem (P1), the solo final never settles on a total, whereas the alternating final states the correct answer, 1. Both P1 finals reach the answer-token limit, so this pair identifies an outcome difference, not a single decisive handoff. In R1 and R2, the teacher reframes the relevant claim and the student's answer changes from incorrect to correct. The R2 teacher also overstates the passage. These selected traces show what an exchange can look like; Table 2 gives the aggregate context. Appendix C includes fuller excerpts and other cases.

Figure 3 shows two complementary behaviors. In M2, the teacher supplies an algebraic substitution that the student uses in its next turn; both conditions reach the correct answer, 12. In R5, the student alone selects B, while the alternating rollout selects C after the teacher rules out B's stronger claim about ferns in thick litter. The student had considered C earlier, so this excerpt shows a useful continuation rather than an isolated moment of discovery.

4.2 Inkling experiments

Setup. We train the weak teacher, Inkling-Small (276B total parameters, 12B active per token) (Thinking Machines Lab, 2026b), alongside a frozen copy of the same model. At evaluation, we replace the frozen partner with Inkling (975B total, 41B active) (Thinking Machines Lab, 2026a) as the strong student. The strong student contributes no training rollouts, and both models remain fixed during transfer. We use the 1,000 public training tasks of ARC-AGI-2 (Chollet et al., 2025), split into 904 training-pool tasks and 96 development tasks. Appendix B.1 provides the training and rollout settings.

Evaluation. Using Rl'ed teacher, we compare Inkling alone, Inkling-Small with the teacher, and Inkling with the teacher (Allspark). Each configuration is evaluated at thinking efforts 0.40, 0.55, 0.70, and 0.85, with three attempts per development task and shared task–attempt seed assignments. We report mean pass@1 over all 288 attempts and the mean retained output tokens per attempt, including both writers and the final answer.

Takeaway. Allspark achieves the highest observed accuracy in Figure 4, reaching 81.6% compared with 78.1% for Inkling alone. It also improves the accuracy–token tradeoff at selected operating points: Allspark reaches 79.2% with 10.8k retained tokens, exceeding the strong student's best

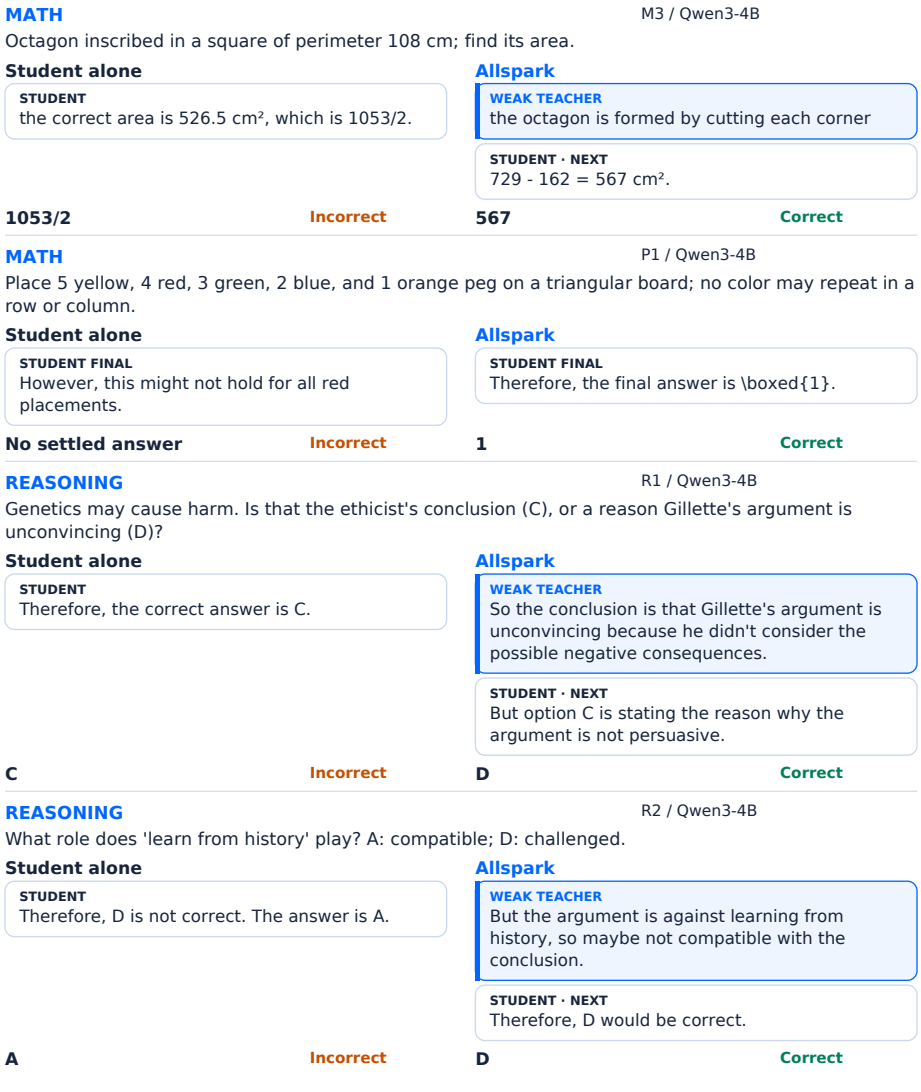


Figure 2: Selected paired math and reasoning rollouts with a Qwen3-4B student and a Qwen3-1.7B weak teacher. Blue bubbles are teacher-authored; white bubbles are student-authored. M3, R1, and R2 show teacher and next-student excerpts; P1 compares final-answer excerpts. Both P1 finals hit the answer-token limit. Quotes are verbatim; task summaries are abbreviated.

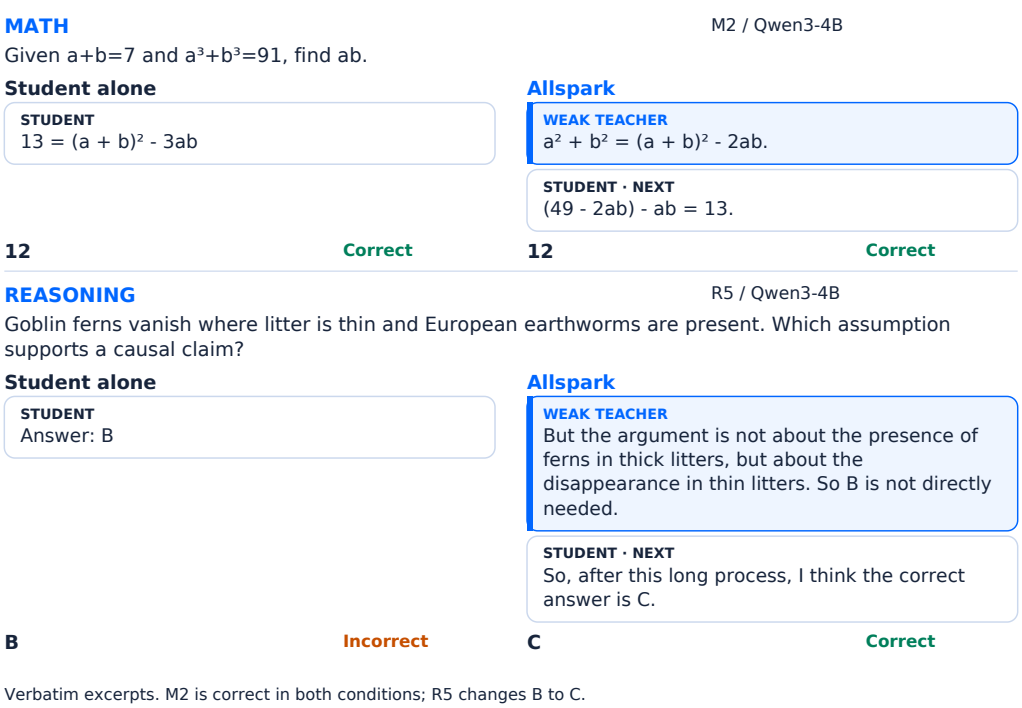


Figure 3: Two further recorded handoffs with Qwen3-4B students and the same Qwen3-1.7B weak teacher. Blue bubbles are teacher-authored; other excerpts are student-authored. Both conditions solve M2; R5 changes from B (incorrect) to C (correct). Quotes are verbatim and task summaries are abbreviated.

observed accuracy of 78.1% at 12.3k tokens. These results support weak-to-strong transfer without updating the strong student.

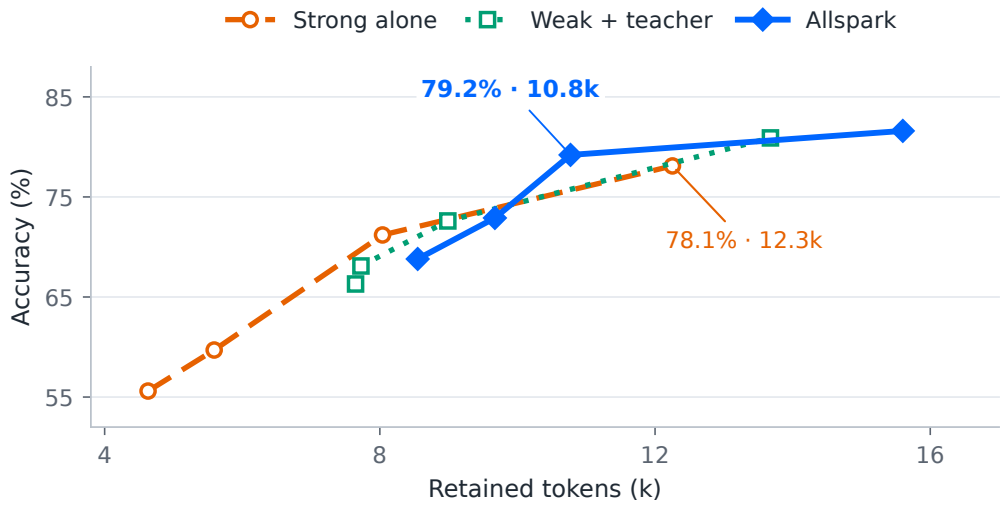


Figure 4: Accuracy–token tradeoff on 96 ARC development tasks, with three attempts per task. Paired runs reuse the rl'ed teacher with an Inkling-Small or Inkling student. Points from left to right use thinking efforts 0.40, 0.55, 0.70, and 0.85. Tokens include retained output from both writers and the final answer.

4.3 CROSS-FAMILY TRANSFER

Setup. We reuse the same Inkling-Small weak teacher with two strong students from other model families: Nemotron-3-Ultra (550B total, 55B active) (NVIDIA, 2026), evaluated in medium- and full-thinking modes, and Kimi-K2.6 (1T total, 32B active) (Moonshot AI, 2026b) on its 128K-context serving tier. Both teacher and student remain frozen; no target-family retraining is performed. Because the models use different tokenizers and reasoning formats, each handoff decodes the retained reasoning into text and re-encodes it in the next model's native reasoning channel. The student starts the trace and always produces the final answer.

Each setting compares the student alone with Allspark on the same 96 ARC development tasks, with three attempts per task and shared task–attempt seed assignments. We report mean pass@1 and task-bootstrap intervals. The teacher uses effort 0.40 with Nemotron medium and 0.85 with Nemotron full and Kimi; their reasoning ceilings are 24,576 and 49,152 tokens, respectively.

The alternating evaluator explicitly closes reasoning and requests a separate student answer, including after a reasoning-ceiling stop; the student-alone baseline must transition to its answer naturally within one completion. Thus, the comparison measures the complete inference procedure. Appendix B.2 specifies the text interface, stopping rules, and token accounting.

Transfer results. We evaluate the target models at native thinking settings rather than tracing an Inkling-style effort curve. Figure 5 shows accuracy gains with different effects on retained output length. For Kimi, Allspark improves accuracy by 11.5 percentage points while using 7.7% fewer retained tokens. For Nemotron, mean accuracy increases by 17.7 and 5.2 percentage points in medium- and full-thinking modes, respectively, with increased token use. These results support reuse of the same teacher across model families, although accuracy and token benefits depends on the target student's inference settings.

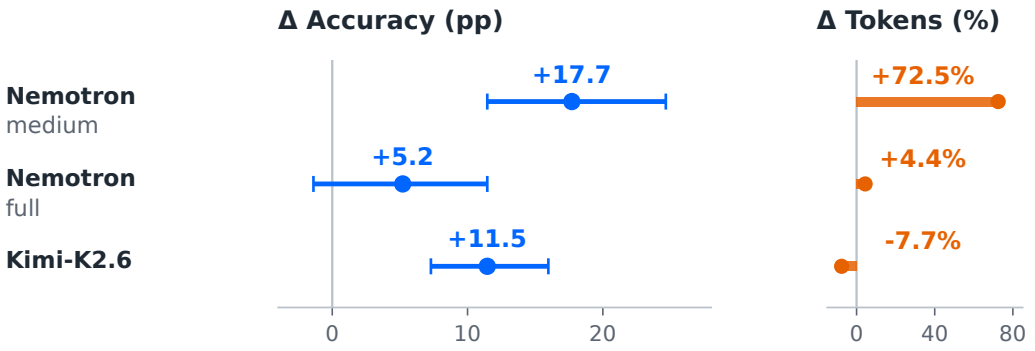


Figure 5: Changes versus each student alone: Nemotron-3-Ultra (medium/full) and Kimi-K2.6 (128K). Same weak teacher; 96 ARC development tasks, three attempts each. Bars: archived 95% task-bootstrap intervals. Tokens: retained output from both models, in native tokenizer units.

5 CONCLUSION

We introduced Allspark, a training and inference framework for weak-to-strong transfer through alternating chains of thought. A weak teacher learns alongside a frozen copy of the same model, then steers a stronger, frozen student at inference. This avoids strong-model rollouts during training, while the shared text interface enables transfer across model sizes and families. Our study spans math and reasoning with Qwen models and ARC-AGI-2 with larger models. The Inkling results demonstrate cross-family transfer and improved accuracy–token tradeoffs. These findings suggest a practical route to exploring RL recipes and training data with smaller models and reusing the resulting teachers with stronger students. We hope Allspark lowers the barrier to RL research and inspires extensions to model alignment and AI safety research.

6 LIMITATIONS

Allspark avoids generating strong-model rollouts during teacher training, but inference requires serving both the teacher and the student. The additional model calls and handoffs may increase inference cost and latency, so reductions in retained tokens do not necessarily translate into lower deployment cost. Our experiments provide initial evidence under limited compute resources. Evaluation covers a restricted set of models, tasks, and training runs; broader studies are needed to establish the robustness and generality of the observed gains. Finally, our implementation relies on the ability to interrupt and resume a shared reasoning stream. Its applicability to non-thinking models, or to interfaces that do not expose a suitable reasoning stream, remains untested.

AI USE STATEMENT

In this work, we used generative AI tools for polishing writing, drafting experiment setting since it's mostly summarization and organization of documented experiments, and discovery of related works. We have not used generative AI tools for research ideation, generating synthetic datasets, provide core analysis in paper and proving mathematical claims. Additionally, we used generative AI tools for creating or modifying scientific figures or images and editing software code. We have reviewed all AI-assisted work. For example, we have gone through the citations found by AI and try our best to make sure they truly exist and they are truly relevant instead of hallucination. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5be69a584901a26c521c2b51e40a4c20-Paper-Conference.pdf.

Anthropic. Introducing Claude Opus 5.5. https://www.anthropic.com/claude-opus-5-5, September 2026. Accessed: 2026-09-24.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023. URL https://arxiv.org/abs/2312.09390.

François Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. ARC-AGI-2: A new challenge for frontier AI reasoning systems. *arXiv preprint arXiv:2505.11831*, 2025. URL https://arxiv.org/abs/2505.11831.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Shiyuan Feng, Huan-ang Gao, Haohan Chi, Hanlin Wu, Zhilong Zhang, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Weak-to-strong generalization via direct on-policy distillation. *arXiv preprint arXiv:2607.05394*, 2026. URL https://arxiv.org/abs/2607.05394v2. Version 2.

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. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yizhu Jiao, Xuchao Zhang, Zhaoyang Wang, Yubo Ma, Zhun Deng, Rujia Wang, Chetan Bansal, Saravan Rajmohan, Jiawei Han, and Huaxiu Yao. Synergistic weak-strong collaboration by aligning preferences. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20355–20371, 2025. doi: 10.18653/v1/2025.acl-long.995. URL https://aclanthology.org/2025.acl-long.995/.

Can Jin, Jiakang Li, Rui Wu, Eddy Z. Zhang, and Dimitris N. Metaxas. Weak critics make strong learners: On-policy critique distillation for scalable oversight. *arXiv preprint arXiv:2606.00424*, 2026. URL https://arxiv.org/abs/2606.00424v2. Version 2.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. URL https://arxiv.org/abs/2607.24653.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. LogiQA 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023. doi: 10.1109/TASLP.2023.3293046.

LLM-Core Xiaomi. MiMo-V2.6: Scaling reinforcement learning towards self-improvement. Technical report, 2026. URL https://huggingface.co/XiaomiMiMo/MiMo-V2.6-Pro-RL/blob/main/MiMo_V2_6_technical_report.pdf. Accessed 23 September 2026.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. URL https://thinkingmachines.ai/blog/on-policy-distillation/.

Martin Marek and Max Ryabinin. Score centering stabilizes off-policy reinforcement learning. *arXiv preprint arXiv:2609.20807*, 2026. URL https://arxiv.org/abs/2609.20807.

Microsoft AI. Introducing MAI-Thinking-1. https://microsoft.ai/news/introducing-mai-thinking-1/, June 2026. Accessed: 2026-09-24.

Moonshot AI. Kimi K2.6: Advancing open-source coding. Official technical blog, apr 2026a. URL https://www.kimi.ai/blog/kimi-k2-6. Published 20 April 2026; accessed 23 September 2026.

Moonshot AI. Kimi K2.6 model card. Hugging Face model card, 2026b. URL https://huggingface.co/moonshotai/Kimi-K2.6. Accessed 25 September 2026.

Yifan Niu, Han Xiao, Dongyi Liu, Zelong Wang, Dihong Gong, Yasheng Wang, and Jia Li. Breaking the tokenizer barrier: On-policy distillation across model families. *arXiv preprint arXiv:2606.09456*, 2026. URL https://arxiv.org/abs/2606.09456.

NVIDIA. NVIDIA Nemotron 3 Ultra. Official research project page, jun 2026. URL https://research.nvidia.com/labs/nemotron/Nemotron-3-Ultra/. Published 4 June 2026; accessed 23 September 2026.

OpenAI. GPT-6 Astra: A new generation of intelligence. https://openai.com/index/gpt-6-astra/, September 2026. Accessed: 2026-09-24.

Youngrok Park, Sangmin Bae, Hojung Jung, Jongwoo Ko, Yunseon Choi, Young Jin Kim, Pashmina Cameron, Aaron Courville, and Se-Young Yun. Eliciting weak-to-strong generalization with on-policy reverse distillation. *arXiv preprint arXiv:2609.08798*, 2026. URL https://arxiv.org/abs/2609.08798.

Shannon Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. Learning to decode collaboratively with multiple language models. *arXiv preprint arXiv:2403.03870*, 2024. URL https://arxiv.org/abs/2403.03870.

Thinking Machines Lab. Inkling: Our open-weights model, jul 2026a. URL https://thinkingmachines.ai/news/introducing-inkling/. Published 15 July 2026; accessed 24 September 2026.

Thinking Machines Lab. Introducing Inkling-Small, jul 2026b. URL https://thinkingmachines.ai/news/inkling-small/. Published 30 July 2026; accessed 21 September 2026.

Thinking Machines Lab. Importance sampling. Tinker Documentation, n.d.a. URL https://tinker-docs.thinkingmachines.ai/tinker/losses/importance-sampling/. Accessed 24 September 2026.

Thinking Machines Lab. Thinking effort. Tinker Documentation, n.d.b. URL https://tinker-docs.thinkingmachines.ai/cookbook/inkling/thinking-effort/. Accessed 21 September 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Yuqing Yang, Yan Ma, and Pengfei Liu. Weak-to-strong reasoning. *arXiv preprint arXiv:2407.13647*, 2024. URL https://arxiv.org/abs/2407.13647.

Fangxu Yu, Weijia Xu, Michael Xu, Tianyi Zhou, and Zinan Lin. Weak-to-strong on-policy distillation. *arXiv preprint arXiv:2607.26246*, 2026. URL https://arxiv.org/abs/2607.26246v2. Version 2.

Yuzhen Zhou, Jiajun Li, Yusheng Su, Gowtham Ramesh, Zilin Zhu, Xiang Long, Chenyang Zhao, Jin Pan, Xiaodong Yu, Ze Wang, Kangrui Du, Jialian Wu, Ximeng Sun, Jiang Liu, Qiaolin Yu, Hao Chen, Zicheng Liu, and Emad Barsoum. APRIL: Active partial rollouts in reinforcement learning to tame long-tail generation. *arXiv preprint arXiv:2509.18521*, 2025. URL https://arxiv.org/abs/2509.18521.

Zhanhui Zhou, Zhixuan Liu, Jie Liu, Zhichen Dong, Chao Yang, and Yu Qiao. Weak-to-strong search: Align large language models via searching over small language models. *arXiv preprint arXiv:2405.19262*, 2024. URL https://arxiv.org/abs/2405.19262.

A QWEN RL IMPLEMENTATION AND TRAINING PROTOCOL

A.1 MODELS AND ROLLOUT MASKS

Both arms initialize from `Qwen/Qwen3-1.7B` with training seed 0. Only LoRA parameters are optimized (Hu et al., 2022); the pretrained weights remain fixed. In Allspark, the student uses the same initial checkpoint without an adapter.

Rollouts and update masks. Ordinary RL samples the entire reasoning and answer from the adapted model. Allspark starts with at most 128 student reasoning tokens, then alternates teacher and student chunks of at most 256 tokens until reasoning ends or reaches its shared token ceiling. Each writer conditions on the prompt and the complete preceding stream. The frozen student generates the final answer. The loss mask m_t is:

Token source	Ordinary RL	Allspark
Sampled adapted-model reasoning	1	1 (teacher)
Sampled student reasoning	—	0
Sampled final answer	1	0 (student)
Forced closing token / separator	0	0

A sampled `</think>` inherits its writer's mask. A closing token inserted at the reasoning ceiling has mask zero. Student tokens remain in the causal context of subsequent teacher predictions. A single binary verifier reward $r \in \{0, 1\}$ for the final answer supplies the training signal; there is no intermediate reasoning reward.

A.2 RL LOSS: MASKED SAMPLER-HEAD CENTERING

Both arms use the same loss. Ordinary RL applies it to the adapted model's reasoning and answer tokens; Allspark applies it only to the teacher's reasoning tokens. The calculation has three steps.

Step 1: solution advantage. For one prompt, sample G trajectories and grade their final answers with binary rewards $r_1, \ldots, r_G$. Compute the group's mean μ and population standard deviation σ, then assign each trajectory its standardized reward:

$$\mu = \frac{1}{G}\sum_{j=1}^{G} r_j, \qquad \sigma^2 = \frac{1}{G}\sum_{j=1}^{G}(r_j - \mu)^2,$$
$$A_i = \begin{cases} (r_i - \mu)/\sigma, & \sigma > 0, \\ 0, & \sigma = 0. \end{cases} \tag{4}$$

We use $G = 32$ trajectories per prompt. The same A_i is used at every trainable position in trajectory i. A better-than-group-average answer has positive advantage; a worse answer has negative advantage. No epsilon is added to the standard deviation.

Step 2: loss at one trainable token. Let h be the full prefix and y the sampled next token. The current trainable model assigns probability $p_\theta(v \mid h)$ to token v. The stored distribution $q(v \mid h)$ comes from the checkpoint that sampled y, evaluated at this same prefix h. It is held fixed and may come from an older checkpoint. Let H contain its K highest-probability tokens, and let A be this trajectory's advantage from Step 1. For this position, minimize

$$\ell(\theta) = -A\left[\underbrace{\mathbf{1}\{y \in H\}\log p_\theta(y \mid h)}_{\text{sampled-token score}} - \underbrace{\sum_{v \in H} q(v \mid h)\log p_\theta(v \mid h)}_{\text{sampler-weighted center}}\right]. \tag{5}$$

The first term scores the token that was sampled. The second is a weighted sum of the learner's log probabilities over the saved sampler head. For an in-head sample, positive advantage favors the

sampled-token score relative to this center; negative advantage reverses that direction. If $y \notin H$, the first term is zero and only the centering term remains.

Only p_θ receives gradients. The advantage, sampler probabilities, and membership in H are fixed. The probabilities in H retain their original mass: they are not renormalized to sum to one. The head is used to compute the loss; generation itself does not use top-K truncation. We use $K = 128$. This is a head-only variant of score centering (Marek & Ryabinin, 2026). Unlike the published implementation, which retains the sampled-token log probability outside the head and models the tail with the learner distribution rescaled to the sampler's tail mass, our loss sets the sampled-token term to zero when $y \notin H$ and omits the tail.

Step 3: loss for one optimizer update. An update contains M prompt groups of G trajectories each. Let $\mathcal{B}$ be this batch of MG trajectories and $\mathcal{T}(\tau)$ the trainable positions in trajectory τ. The update loss is

$$\mathcal{L}(\theta) = \frac{1}{MG} \sum_{\tau \in \mathcal{B}} \sum_{t \in \mathcal{T}(\tau)} \ell_{\tau,t}(\theta). \tag{6}$$

We use $M = 6$ prompt groups (two per domain), giving 192 trajectories per update. Thus token losses are *summed*, and trajectories are *averaged*. There is no division by a trajectory's token count. The set $\mathcal{T}$ implements the mask in Appendix A.1: student and forced tokens are excluded from the Allspark loss. A trajectory with no trainable positions contributes zero loss but still enters its group's advantage calculation and the denominator MG.

A.3 Asynchronous RL and filtering

Three rollout actors generate trajectories while one learner updates the adapter. Each actor holds one immutable adapter snapshot for a prompt group, including screening and the fresh training rollouts. The learner pools six accepted groups (two per domain), accumulates their gradients, performs one optimizer step, and publishes a new adapter checkpoint. A batch can therefore contain groups sampled by different behavior-policy versions.

Immediately before consumption, each group's version lag must satisfy $0 \leq v_{\text{learner}} - v_{\text{sampler}} \leq \Delta_{\max}$, where $\Delta_{\max}$ is the maximum permitted lag, set to two updates. Older queued groups are discarded. Each loss-bearing token uses its own saved sampler probabilities in Equation 5. This is asynchronous off-policy optimization with bounded lag, without an importance-sampling ratio, PPO clipping, reference-policy KL term, or entropy bonus. The sampler-head term is not an importance-weight correction for stale rollouts.

Prompt and group filtering. Both arms use the same frozen training pool drawn from MATH and LogiQA2. Initial-policy solo screening retains a prompt when its first complete eight-sample group has 1–7 successes. Online, each actor pins a sampling snapshot, screens a candidate with eight fresh trajectories, and requires 1–7 successes. It then draws 32 fresh trajectories using a separate random-seed namespace; groups with 1–31 successes are retained in full. Screening trajectories and tied training groups do not enter updates. Each batch contains six distinct prompts, two per domain; prompts may recur in later updates. Allspark shares student random-seed keys within a group, including the answer stage, while teacher keys vary across trajectories.

A.4 Solution-length controls and termination

Length is controlled by separate hard generation ceilings for reasoning and the final answer. The reasoning ceiling is shared across all chunks from both writers, not allocated afresh to each writer or turn. Each call is limited by the remaining budget; Allspark additionally limits its first student chunk to 128 tokens and later chunks to 256. All returned raw tokens are retained in the shared stream, without paragraph trimming or prefix cropping.

Limit (tokens)	Training / historical evaluation	Revised evaluation
Total context	8,192	16,384
Shared reasoning	4,096	8,192
Final answer	1,536	4,096

A sampled `</think>` ends reasoning naturally. If the reasoning ceiling is reached first, the generator inserts `</think>` with zero loss mask and starts the answer phase. An EOS or empty generation during reasoning instead invalidates the answer stage. The answer stops at its own ceiling or native EOS; unused reasoning budget is not transferred to the answer allowance. Before generation, the full prompt, both allowances, and required boundary tokens must fit the context window; an oversized request is rejected rather than silently cropped.

There is no minimum generation length, length-dependent reward penalty, or per-sequence length normalization in the loss. Equation 6 sums trainable-token terms, so different trajectory lengths contribute different numbers of terms. A cap hit is recorded, but does not itself force zero reward: the saved final answer is scored by the verifier. Report actual reasoning, answer, and trainable-token counts and both cap rates; identical ceilings do not imply identical realized length or compute. The longer evaluation protocol changes inference only and cannot repair truncated training rewards.

A.5 Optimization settings

Gradients accumulate one trajectory at a time over the accepted batch, followed by one clipped optimizer step. Log probabilities and loss arithmetic use FP32; the base model is loaded in BF16. Causal forward passes retain all prefixes preceding loss-bearing tokens, with gradient checkpointing. Vocabulary projections are evaluated in chunks of 32 loss-bearing positions.

Table 3: Shared settings for the local Qwen training runs.

Setting	Value
Optimizer	AdamW; constant learning rate 10^{-5}
Adam moments / epsilon	$(0.9, 0.999)$ / 10^{-8}
Weight decay / gradient-norm ceiling	0 / 1
LoRA rank r / α / dropout	16 / 32 / 0
LoRA projections	q, k, v, o, gate, up, down; no biases
Training context ceiling	8,192 tokens
Reasoning / answer ceilings	4,096 / 1,536 generated tokens
Sampling	Temperature 1; top-$p = 1$; top-k disabled
Sampler head used in the loss	128 tokens; no generation truncation
Queue capacity	12 prompt groups, four per domain

A.6 Development evaluation and checkpoint selection

The frozen development panel contains 384 questions, with 128 each from the MATH, TACO, and LogiQA 2.0 training splits, disjoint from the local training pool. Each candidate is evaluated as a teacher alternating with the same frozen, unadapted Qwen3-1.7B student; the student writes the final answer. There is one sampled answer per question (seed 0), using the revised evaluation limits in Appendix A.4 and the same sampling settings and chunk schedule. The training limits remain unchanged.

Checkpoint selection rule. All 384 development questions are evaluated for each candidate. Let N_d be the number of development questions in domain d and $c_d(s)$ the number correct at checkpoint s. Selection is

$$s^* = \min \operatorname*{argmax}_{s \in \mathcal{C}} \frac{c_{\text{math}}(s) + c_{\text{reasoning}}(s)}{N_{\text{math}} + N_{\text{reasoning}}}. \tag{7}$$

Here $N_{\text{math}} = N_{\text{reasoning}} = 128$. Exact ties choose the earlier update. Reasoning accuracy and the three-domain mean are reported separately and do not choose the checkpoint. Ordinary-RL update

92 is the nominated comparison checkpoint. . The completed review selects checkpoint 94 (Table 4);

Table 4: Finalized Allspark checkpoints under the revised evaluation protocol. Entries are correct answers on the two reported domains; each has 128 development questions. Checkpoint 94 is the fixed selection from the original three-domain protocol. Bold marks the selected checkpoint. These are development results, not public-test scores.

Checkpoint	Math	Reasoning
55	116	93
91	114	96
94	119	96

it is held fixed for subsequent evaluations and transfer to the 4B student. Ordinary-RL checkpoint 92 retains its earlier nomination.

A.7 Development learning curves

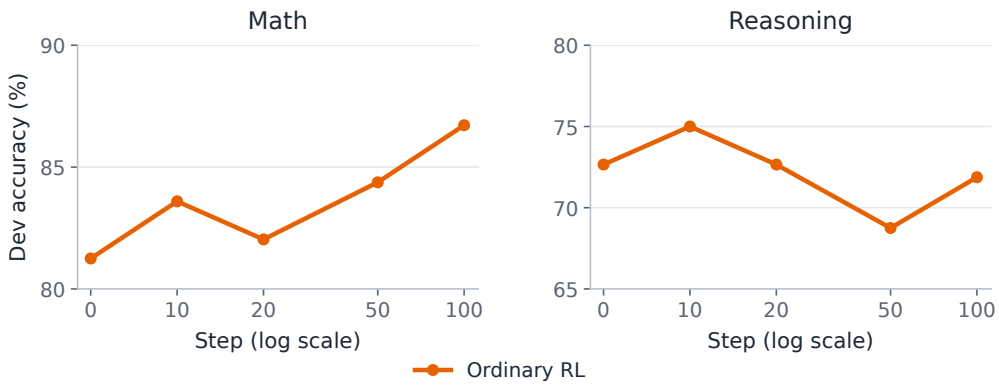


Figure 6: Development accuracy of the ordinary-RL-trained Qwen3-1.7B teacher alternating with the frozen student, under the historical 8K protocol (4,096 reasoning and 1,536 answer tokens). Each domain has 128 fixed questions; the figure reports math and reasoning only. Filled circles connect the initial checkpoint (step 0) to checkpoints 10, 20, 50, and 100 without smoothing.

A.8 TRAINING DIAGNOSTICS

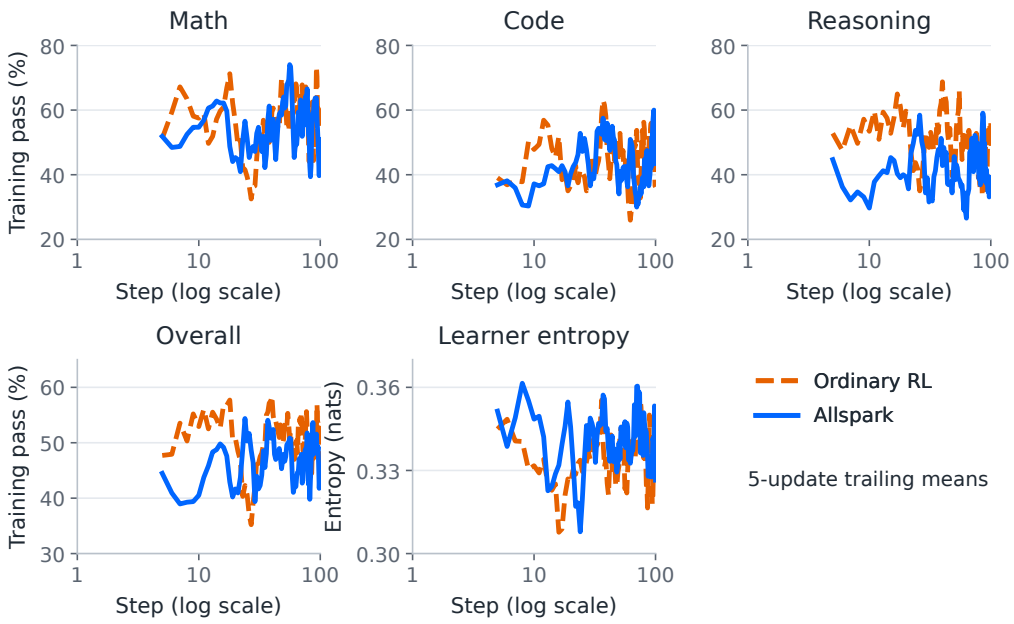


Figure 7: Qwen3-1.7B training diagnostics over the first 100 committed updates of each method on a logarithmic step axis (one seed per method). Lines show full trailing-five-update means, starting at update 5. Pass rates are measured on accepted, filtered training rollouts; overall is the equal-domain mean. Learner entropy is measured on the respective trainable-token contexts. The accepted prompts and rollout policies differ between methods, so these curves are descriptive training diagnostics, not held-out accuracy or paired treatment effects.

Allspark completed 100 updates; the ordinary-RL run stopped at 110. Figure 7 shows the common 1–100 range. The additional ten ordinary-RL updates remain part of its training expenditure. Update counts do not equate generated tokens, optimized tokens, or GPU time; screening and rejected rollouts also contribute to training cost.

B INKLING TRAINING PROTOCOL

B.1 INKLING TRAINING AND INFERENCE

Training. We trained a rank-32 LoRA adapter (Hu et al., 2022) on Inkling-Small for 72 rollout steps (indexed 0–71), paired with a frozen copy of the same base model. The frozen copy opened each reasoning trace and generated the visible answer. Each step sampled 16 tasks at all three thinking efforts, $\{0.4, 0.55, 0.7\}$, with 32 rollouts per task–effort group: nominally 1,536 training rollouts. We used four minibatches, keeping each group together, with one optimizer update per nonempty minibatch. The learning rate was 2×10^{-5} through step 24 and 5×10^{-5} thereafter.

RL loss. Because training used the Tinker API, we chose its built-in `importance_sampling` policy-gradient objective (Thinking Machines Lab, n.d.a). Each trajectory contributes

$$\ell(\pi) = -\frac{\bar{n}}{n} A \sum_{t \in T} \frac{\pi(y_t \mid h_t)}{\pi_{\text{old}}(y_t \mid h_t)}, \qquad A = r - \bar{r}. \tag{8}$$

Here π is the current teacher, π_{old} is the teacher that sampled the rollout, and h_t is the full shared prefix before token y_t. T contains only teacher-generated reasoning positions, $n = |T|$, and $\bar{n}$ is the mean teacher-token count across retained trajectories in the rollout batch. The factor $\bar{n}/n$ normalizes each trajectory's contribution by its trainable length. The baseline $\bar{r}$ is the mean reward within the same task–effort group; we do not divide by the group standard deviation. We sum these losses within each minibatch. The prompt, frozen-model chunks, and final answer receive no policy-gradient loss.

Reward and length control. Let s indicate exact task success, c denote cell accuracy, and L count retained reasoning tokens from both writers, excluding the final answer. We use

$$r = ws + (1 - w)c - \min\left\{p_{\max}, \lambda \max\left(0, \frac{L}{L_0} - 1\right)\right\}. \tag{9}$$

We set $w = 0.85$, $L_0 = 5{,}000$, and $p_{\max} = 0.30$. Steps 0–35 use $\lambda = 0$; steps 36–71 continue the same adapter with $\lambda = 0.15$. Thus the length penalty is zero through 5,000 reasoning tokens, then increases by 0.15 per additional 5,000 tokens, up to 0.30. Training also imposes a hard reasoning ceiling of 24,576 tokens; on reaching it, the reasoning channel is closed and the frozen model generates the final answer.

Models and rollout settings. All Inkling models are served through Tinker. The teacher and its frozen training partner use `thinkingmachines/Inkling-Small`; the strong transfer student is `thinkingmachines/Inkling`. A deterministic task-ID hash defines the 904/96 training-pool/development split. The student opens each trace with at most 256 tokens, followed by alternating teacher and student chunks of at most 2,048 tokens. Handoffs are trimmed to the last paragraph boundary when at least half the sampled chunk can be retained. The frozen student produces the final answer as a JSON object containing the predicted output grids. Both writers use temperature 1.0 and top-p 1.0, with a final-answer allowance of 4,096 tokens. In the checkpoint-63 effort sweep, the reasoning ceiling is 24,576 tokens at efforts 0.40, 0.55, and 0.70, and 49,152 tokens at effort 0.85.

B.2 CROSS-FAMILY INFERENCE PROTOCOL

Models and operating points. The cross-family evaluation pairs the Inkling-Small teacher from the within-family study with frozen Nemotron-3-Ultra or Kimi-K2.6 students. Neither the teacher nor the target student is updated during these evaluations. Table 5 lists the three operating points; the two Nemotron rows are settings of one model family. The numerical effort parameter applies only to Inkling-Small. The target students instead use their native thinking renderers, so equal numerical effort or equal reasoning compute across families is not assumed.

Text handoffs. Each model receives the same ARC task messages, rendered in its own format. Every reasoning chunk is decoded with its writer's tokenizer, stripped of model-specific control markers, and appended to a shared text trace. Before the next turn, the complete trace is re-encoded with the receiving model's tokenizer and placed in its native hidden-reasoning channel. This preserves

Table 5: Cross-family inference settings. Context and reasoning limits are in tokens. The context limit applies to the student; the reasoning guard is checked in the receiving model's tokenizer at each handoff.

Student setting	Teacher effort	Student context	Reasoning limit
Nemotron-3-Ultra, medium	0.40	65,536	24,576
Nemotron-3-Ultra, full	0.85	65,536	49,152
Kimi-K2.6, 128K	0.85	131,072	49,152

the retained reasoning text while flattening model-specific message boundaries; token IDs are never passed directly between families. The student starts the trace, subsequent turns alternate between teacher and student, and the student always generates the final answer. The evaluation configuration specifies a 256-token first chunk, 2,048-token later chunks, paragraph trimming that retains at least half of a sampled chunk, temperature 1.0, top-p 1.0, and a final-answer allowance of 4,096 tokens.

Stopping and grading. At each handoff, the evaluator recomputes the trace length in the receiving model's tokenizer and limits the next chunk by both the reasoning ceiling and the remaining context, reserving space for the answer. Either model can end reasoning with its native closing marker. Alternation also stops when the reasoning allowance is exhausted. The evaluator then closes the hidden channel in the student's format and requests a separate final answer, including after a reasoning-ceiling stop. By contrast, the student-alone baseline generates one uninterrupted completion with a total allowance equal to its reasoning room plus the answer allowance; it must emit its own hidden-to-visible boundary. Only the visible final answer is graded against the exact ARC output grids.

Aggregation and scope. Each condition contains 288 attempts over the same 96 development tasks. Mean pass@1 averages exact success over attempts; it is not best-of-three accuracy. For student-alone comparisons, task–attempt seed assignments are paired. The archived 95% intervals use 5,000 paired bootstrap resamples of task-level mean differences, with bootstrap seed 17. The Inkling-Small–teacher reference is reused from the within-family scan; its medium-effort comparison is task matched but not seed matched. Token totals sum retained teacher and student output, including the final answer, in each writer's native tokenizer units; discarded chunk suffixes are excluded. These totals are neither a common-tokenizer budget nor total billed generation. A separate 16-task target-model pilot examined thinking modes and context limits, including an earlier 32K Kimi configuration.

B.3 Earlier within-family control

This evaluation uses checkpoint 35 at effort 0.7, with one attempt on each of the 96 development tasks. Each frozen student (Inkling-Small or Inkling) is evaluated alone, with an untrained Inkling-Small teacher, and with the checkpoint-35 Allspark teacher. All model weights are fixed during evaluation.

Table 6: Step 35, effort 0.7, one attempt on each of 96 ARC development tasks.

Student	Teacher	Accuracy (%)
Inkling-Small	None	60.4
Inkling-Small	Untrained	60.4
Inkling-Small	Allspark-trained	70.8
Inkling	None	75.0
Inkling	Untrained	66.7
Inkling	Allspark-trained	76.0

B.4 Dollar cost and latency

Cost model. We report the inference-cost estimates preserved in the Tinker experiment archive. They use the rates in Table 7, which apply the archive's recorded 50% account discount to its list-price

snapshot. These are historical analysis assumptions, not current price quotes. The calculation assumes a separate prefix cache for each writer: each model prefills the prompt on its first turn, pays the cached-read rate for its existing prefix on later turns, and pays fresh-prefill rates for newly received text. Retained output, including the final answer, is charged at the generating model's sampling rate.

Table 7: Historical rate assumptions, in USD per million tokens.

Model / serving tier	Fresh prefill	Cached read	Sampling
Inkling-Small	0.580	0.058	1.440
Inkling	1.870	0.187	4.680
Nemotron-3-Ultra, 64K	2.490	0.249	6.225
Kimi-K2.6, 128K	2.575	0.515	6.405

Table 8: Archived modeled inference cost, in USD per attempt, over 96 development tasks with three attempts each. Weak + teacher always uses an Inkling-Small student and the trained teacher; the cross-family rows reuse that reference. For the Inkling rows, effort applies to both writers; for cross-family rows it denotes the teacher's effort, with the target student using its native thinking setting.

Student setting	Effort	Student alone	Weak + teacher	Allspark
Inkling	0.40	0.0266	0.0198	0.0483
	0.55	0.0311	0.0199	0.0546
	0.70	0.0426	0.0232	0.0606
	0.85	0.0623	0.0354	0.0895
Nemotron, medium	0.40	0.0459	0.0196	0.0757
Nemotron, full	0.85	0.1230	0.0352	0.1407
Kimi, 128K	0.85	0.1466	0.0352	0.1774

At the same Inkling effort, Allspark's modeled cost per attempt is 1.42–$1.82\times$ that of the student alone. Across different efforts, the highlighted Inkling comparison uses 12.1% fewer retained tokens (10,771 versus 12,253), but only about 2.7% less modeled cost ($0.0606 versus $0.0623). The cross-family estimates are higher for Allspark in all three settings. In particular, Kimi's 7.7% reduction in retained tokens accompanies a 21.1% increase in modeled cost, from approximately $0.1466 to $0.1774 per attempt. Prefix processing, cached rereads, and the mix of teacher and student tokens therefore affect the dollar comparison.

C ROLLOUT CASE STUDIES

The cases below are selected paired development rollouts. They illustrate the content exchanged between a frozen student and the weak teacher; they do not establish the causal effect of an individual teacher span. We include successful, unchanged, ambiguous, and unsuccessful pairs.

Selection context. The cases include the two domains reported in Table 2 and two supplementary coding rollouts. They cover unchanged, recovered, and unsuccessful outcomes. Full task identities, excerpt role offsets, and source hashes are retained in the accompanying case manifest; complete traces remain in the run archive.

Case	Domain	Student	Native → Allspark
M1	Math	4B	correct → correct
M2	Math	4B	correct → correct
M3	Math	4B	incorrect → correct
R1	Reasoning	4B	incorrect → correct
R2	Reasoning	4B	incorrect → correct
C1	Coding	4B	incorrect → correct
C4	Coding	4B	correct → incorrect

M1 | Math | Qwen3-4B

The unpaired square root

TASK SUMMARY

Name the greatest whole number less than 100 that has an odd number of positive factors.

Student alone

STUDENT · EXCERPT

Oh! I think it's when the number is a perfect square. Because in a perfect square, one of the factors is repeated.

81 Correct

Allspark · alternating reasoning

STUDENT · TURN 1

I remember that numbers usually have factors in pairs, right? Like for a number n, if d is a factor, then n/d is also a factor.

WEAK TEACHER · TURN 2

If a number is a perfect square, then one of its factors is the square root, and that factor pairs with itself.

STUDENT · TURN 3

Wait, according to the logic, if all factors come in pairs, then the number of factors is even unless there is a square root factor. Therefore, only perfect squares have an odd number of factors.

81 Correct

READING THE EXCHANGE

The teacher names the self-pairing square root and the student carries the criterion forward to 81.

math:EleutherAI/hendrycks_math:number_theory:261

Exact saved excerpts · teacher 1.7B / step 94 · selected development pair

Figure 8: Case M1: The unpaired square root. Recorded excerpts; task text is summarized above.

M2 | Math | Qwen3-4B

Eliminate the square terms

TASK SUMMARY

Two reals a and b are such that a+b=7 and $a^3+b^3=91$. Compute ab.

Student alone

STUDENT · EXCERPT

$13 = (a + b)^2 - 3ab$

12 Correct

Allspark · alternating reasoning

STUDENT · TURN 1

Oh yeah, the formula is $a^3 + b^3 = (a + b)(a^2 - ab + b^2)$. Right? So maybe I can use that here.

WEAK TEACHER · TURN 2

$a^2 + b^2 = (a + b)^2 - 2ab$.

Since a + b is 7, that would be $7^2 - 2ab$, which is 49 - 2ab.

STUDENT · TURN 3

(49 - 2ab) - ab = 13. So combining like terms:

49 - 3ab = 13.

12 Correct

READING THE EXCHANGE

The teacher substitutes for the squared terms, and the student combines them to obtain ab=12.

math:EleutherAI/hendrycks_math:intermediate_algebra:42

Exact saved excerpts · teacher 1.7B / step 94 · selected development pair

Figure 9: Case M2: Eliminate the square terms. Recorded excerpts; task text is summarized above.

M3 | Math | Qwen3-4B

Recovery after uncertain geometry

TASK SUMMARY

An octagon is inscribed in a square so that the vertices of the octagon trisect the sides of the square. The perimeter of the square is 108 centimeters. What is the number of square centimeters in the area of the octagon?

Student alone

STUDENT · EXCERPT

Given that the shoelace formula is reliable and I've double-checked it, I think the correct area is 526.5 cm², which is 1053/2.

1053/2 Incorrect

Allspark · alternating reasoning

STUDENT · TURN 23

Wait, maybe each triangle has legs of 9 cm? Let me think.

WEAK TEACHER · TURN 24

Alternatively, maybe the octagon is formed by cutting each corner

STUDENT · TURN 25

The area of this triangle is (9 * 9)/2 = 40.5 cm². Similarly, each corner has such a triangle. There are four corners, so total area cut off is 4 * 40.5 = 162 cm². Then, the area of the square is 27*27 = 729 cm². Therefore, the area of the octagon would be 729 - 162 = 567 cm². But wait, is that correct?

567 Correct

READING THE EXCHANGE

The teacher returns to corner subtraction and the student computes 567, which matches the eventual final answer.

math:EleutherAI/hendrycks_math:geometry:89

Exact saved excerpts · teacher 1.7B / step 94 · selected development pair

Figure 10: Case M3: Recovery after uncertain geometry. Recorded excerpts; task text is summarized above.

R1 | Reasoning | Qwen3-4B

Separate a reason from the conclusion

TASK SUMMARY

An ethicist rejects Gillette's promise of genetic research benefits because it ignores possible harms. What is the conclusion? C: research might harm more than help. D: Gillette's argument is unconvincing because it ignores possible consequences.

Student alone

STUDENT · EXCERPT

Therefore, the correct answer is C. Because the ethicist is saying that the harm is a possibility, which is the main point. The answer is C.

C Incorrect

Allspark · alternating reasoning

STUDENT · TURN 1

So the question is asking for the conclusion of the ethicist's argument. The conclusion is the main point the ethicist is trying to make.

WEAK TEACHER · TURN 2

The ethicist is saying that Gillette's argument isn't good because he ignores possible harms. So the conclusion is that Gillette's argument is unconvincing because he didn't consider the possible negative consequences.

STUDENT · TURN 3

But option C is stating the reason why the argument is not persuasive. However, the conclusion of the ethicist's argument is that Gillette's argument is not persuasive.

D Correct

READING THE EXCHANGE

The teacher identifies the criticism as the conclusion, and the next student turn distinguishes it from the supporting reason.

reasoning:csitfun/LogiQA2.0:13825

Exact saved excerpts · teacher 1.7B / step 94 · selected development pair

Figure 11: Case R1: Separate a reason from the conclusion. Recorded excerpts; task text is summarized above.

R2 | Reasoning | Qwen3-4B

Find the claim being challenged

TASK SUMMARY

The passage questions whether history's lessons can be discovered or applied. What role does 'we should learn the lessons of history' play? A: compatible with accepting or denying the conclusion. D: the position the argument aims to discredit.

Student alone

STUDENT · EXCERPT

Therefore, D is not correct. The answer is A. Because the position is compatible with the conclusion.

A Incorrect

Allspark · alternating reasoning

STUDENT · TURN 1

The question is about how the idea that we should learn from history is used in the argument. The options are A to D. Let me break down each part.

WEAK TEACHER · TURN 2

But the argument is against learning from history, so maybe not compatible with the conclusion. Wait, the conclusion is that we shouldn't learn from history, so if the argument's conclusion is that we shouldn't learn, then the idea of learning is being challenged. So maybe A is not correct.

STUDENT · TURN 3

So the author is arguing against the idea that learning from history is useful. So the argument is directed towards discrediting the idea that we should learn from history. Therefore, D would be correct.

D Correct

READING THE EXCHANGE

The teacher frames the historical-lessons claim as the target of criticism, and the next student turn selects D.

reasoning:csitfun/LogiQA2.0:10768:row:train:855c7899398a7d2c3d14b2e6c26679fc2def1e70703ffa71ca3fb9ff3df406ca
Exact saved excerpts · teacher 1.7B / step 94 · selected development pair

Figure 12: Case R2: Find the claim being challenged. Recorded excerpts; task text is summarized above.

C1 | Coding | Qwen3-4B

Parity before an exception

TASK SUMMARY

Remove as many tree edges as possible while leaving every connected component even-sized; return -1 if impossible.

Student alone

STUDENT · EXCERPT

```
if n == 1:
        print(0)
        return
```

0 for n=1 (incorrect) Incorrect

Allspark · alternating reasoning

STUDENT · TURN 1

The key is to find how many such splits are possible. Let's think about the conditions.

WEAK TEACHER · TURN 2

An important observation is that for the entire tree to be split into components with even sizes, the total number of vertices must be even.

STUDENT · TURN 3

Wait, but the original tree has n vertices. So, if n is odd, then it's impossible. Because even if you split into multiple components, each of even size, the sum would be even.

FINAL CODE · EXCERPT

```
if n % 2 != 0:
        print(-1)
        return
```

-1 for n=1; 96/96 supplied tests Correct

READING THE EXCHANGE

The teacher introduces the even-total invariant, the student adopts it, and the final code rejects the one-node case.

code:BAAI/TACO:train:16753

Exact saved excerpts · teacher 1.7B / step 94 · selected development pair

Figure 13: Case C1: Parity before an exception. Recorded excerpts; task text is summarized above.

C4 | Coding | Qwen3-4B

A shared indexing error

TASK SUMMARY

Convert an integer from 0 through 99 to its lowercase English spelling.

Student alone

STUDENT · EXCERPT

```
print(f"{tens_words[tens_digit]}-{o
nes_words[ones_digit]}")
```

ninety-nine for 99; 200/200 supplied tests Correct

Allspark · alternating reasoning

STUDENT · TURN 3

So the code needs to have two dictionaries or lists: one for the ones (0-19), and one for the tens (2-9).

WEAK TEACHER · TURN 4

tens = ["", "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety"]

STUDENT · TURN 5

For example, 20: s//10 is 2, so index 0 in the tens list.

eighty-nine for 99 (incorrect) Incorrect

READING THE EXCHANGE

The teacher supplies an inconsistent tens table and the student repeats the indexing mistake in the final code.

code:BAAI/TACO:train:17287

Exact saved excerpts · teacher 1.7B / step 94 · selected development pair

Figure 14: Case C4: A shared indexing error. Recorded excerpts; task text is summarized above.