

References

- [1] L. AI. Litgpt. <https://github.com/Lightning-AI/litgpt>, 2023. 1
- [2] L. B. Allal, A. Lozhkov, E. Bakouch, G. M. Blázquez, G. Penedo, L. Tunstall, A. Marafioti, H. Kydlíček, A. P. Lajarín, V. Srivastav, J. Lochner, C. Fahlgren, X.-S. Nguyen, C. Fourier, B. Burtenshaw, H. Larcher, H. Zhao, C. Zakka, M. Morlon, C. Raffel, L. von Werra, and T. Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. 2
- [3] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 22
- [4] L. Bethune, D. Grangier, D. Busbridge, E. Gualdoni, M. Cuturi, and P. Ablin. Scaling laws for forgetting during finetuning with pretraining data injection. *arXiv preprint arXiv:2502.06042*, 2025. 2
- [5] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O’Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023. 22
- [6] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi. Piqa: Reasoning about physical common-sense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020. 2
- [7] D. Brandfonbrener, N. Anand, N. Vyas, E. Malach, and S. Kakade. Loss-to-loss prediction: Scaling laws for all datasets. *arXiv preprint arXiv:2411.12925*, 2024. 22
- [8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1
- [9] D. Busbridge, A. Shidani, F. Weers, J. Ramapuram, E. Littwin, and R. Webb. Distillation scaling laws. *arXiv preprint arXiv:2502.08606*, 2025. 22
- [10] T. Chu, Y. Zhai, J. Yang, S. Tong, S. Xie, D. Schuurmans, Q. V. Le, S. Levine, and Y. Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025. 1, 22
- [11] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018. 2
- [12] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 2, 3, 26
- [13] Z. Du, A. Zeng, Y. Dong, and J. Tang. Understanding emergent abilities of language models from the loss perspective. *arXiv preprint arXiv:2403.15796*, 2024. 22
- [14] R. M. French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 4
- [15] S. Y. Gadre, G. Smyrnis, V. Shankar, S. Gururangan, M. Wortsman, R. Shao, J. Mercat, A. Fang, J. Li, S. Keh, et al. Language models scale reliably with over-training and on downstream tasks. *arXiv preprint arXiv:2403.08540*, 2024. 1, 9, 22
- [16] K. Gandhi, A. Chakravarthy, A. Singh, N. Lile, and N. D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. 1
- [17] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant. Did aristotle use a lap-top? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. 3, 26

- [18] A. Gu, B. Rozière, H. Leather, A. Solar-Lezama, G. Synnaeve, and S. I. Wang. Cruxeval: A benchmark for code reasoning, understanding and execution. *arXiv preprint arXiv:2401.03065*, 2024. 3, 26
- [19] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021. 2, 3, 26
- [20] D. Hernandez, T. Brown, T. Conerly, N. DasSarma, D. Drain, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, T. Henighan, T. Hume, et al. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022. 1
- [21] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017. 1
- [22] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 1, 2, 22
- [23] S. Hu, Y. Tu, X. Han, C. He, G. Cui, X. Long, Z. Zheng, Y. Fang, Y. Huang, W. Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. 1
- [24] A. Ibrahim, B. Thérien, K. Gupta, M. L. Richter, Q. Anthony, T. Lesort, E. Belilovsky, and I. Rish. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*, 2024. 2, 4
- [25] J. Jin, V. Syrgkanis, S. Kakade, and H. Zhang. Discovering hierarchical latent capabilities of language models via causal representation learning, 2025. Manuscript. 9
- [26] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1, 22
- [27] M. Kazemi, Q. Yuan, D. Bhatia, N. Kim, X. Xu, V. Imbrasaite, and D. Ramachandran. Boardgameqa: A dataset for natural language reasoning with contradictory information. *Advances in Neural Information Processing Systems*, 36:39052–39074, 2023. 3, 26
- [28] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. 26
- [29] H. Li, W. Zheng, J. Hu, Q. Wang, H. Zhang, Z. Wang, S. Xuyang, Y. Fan, S. Zhou, X. Zhang, et al. Predictable scale: Part i—optimal hyperparameter scaling law in large language model pretraining. *arXiv preprint arXiv:2503.04715*, 2025. 1
- [30] J. LI, E. Beeching, L. Tunstall, B. Lipkin, R. Soletskyi, S. C. Huang, K. Rasul, L. Yu, A. Jiang, Z. Shen, Z. Qin, B. Dong, L. Zhou, Y. Fleureau, G. Lample, and S. Polu. Numina-math. [<https://huggingface.co/AI-MQ/NuminaMath-CoT>] (https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024. 1, 2, 26
- [31] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 1
- [32] C. Y. Liu, L. Zeng, J. Liu, R. Yan, J. He, C. Wang, S. Yan, Y. Liu, and Y. Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024. 3
- [33] E. Liu, A. Bertsch, L. Sutawika, L. Tjauatja, P. Fernandes, L. Marinov, M. Chen, S. Singhal, C. Lawrence, A. Raghunathan, et al. Not-just-scaling laws: Towards a better understanding of the downstream impact of language model design decisions. *arXiv preprint arXiv:2503.03862*, 2025. 1

- [34] P. Lu, L. Qiu, K.-W. Chang, Y. N. Wu, S.-C. Zhu, T. Rajpurohit, P. Clark, and A. Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*, 2022. 3, 26
- [35] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018. 2
- [36] N. Muennighoff, A. Rush, B. Barak, T. Le Scao, N. Tazi, A. Piktus, S. Pyysalo, T. Wolf, and C. A. Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023. 22
- [37] G. Penedo, H. Kydliček, A. Lozhkov, M. Mitchell, C. A. Raffel, L. Von Werra, T. Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024. 1, 2, 25
- [38] Z. Qi, M. Ma, J. Xu, L. L. Zhang, F. Yang, and M. Yang. Mutual reasoning makes smaller llms stronger problem-solvers. *arXiv preprint arXiv:2408.06195*, 2024. 2
- [39] Z. Qin, Q. Dong, X. Zhang, L. Dong, X. Huang, Z. Yang, M. Khademi, D. Zhang, H. H. Awadalla, Y. R. Fung, et al. Scaling laws of synthetic data for language models. *arXiv preprint arXiv:2503.19551*, 2025. 22
- [40] H. Que, J. Liu, G. Zhang, C. Zhang, X. Qu, Y. Ma, F. Duan, Z. Bai, J. Wang, Y. Zhang, et al. D-cpt law: Domain-specific continual pre-training scaling law for large language models. *Advances in Neural Information Processing Systems*, 37:90318–90354, 2024. 2, 22
- [41] M. Raghavendra, V. Nath, and S. Hendryx. Revisiting the superficial alignment hypothesis. *arXiv preprint arXiv:2410.03717*, 2024. 1, 6, 22
- [42] Y. Ren and D. J. Sutherland. Learning dynamics of llm finetuning. *arXiv preprint arXiv:2407.10490*, 2024. 1
- [43] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021. 2
- [44] R. Schaeffer, B. Miranda, and S. Koyejo. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36:55565–55581, 2023. 1
- [45] F. Schmidt. Generalization in generation: A closer look at exposure bias. *arXiv preprint arXiv:1910.00292*, 2019. 1
- [46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2
- [47] Y. Shen, M. Stallone, M. Mishra, G. Zhang, S. Tan, A. Prasad, A. M. Soria, D. D. Cox, and R. Panda. Power scheduler: A batch size and token number agnostic learning rate scheduler. *arXiv preprint arXiv:2408.13359*, 2024. 1
- [48] G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024. 1
- [49] C. Snell, E. Wallace, D. Klein, and S. Levine. Predicting emergent capabilities by finetuning. *arXiv preprint arXiv:2411.16035*, 2024. 22
- [50] J. M. Springer, S. Goyal, K. Wen, T. Kumar, X. Yue, S. Malladi, G. Neubig, and A. Raghunathan. Overtrained language models are harder to fine-tune. *arXiv preprint arXiv:2503.19206*, 2025. 1, 3, 9
- [51] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1
- [52] K. Tirumala, A. Markosyan, L. Zettlemoyer, and A. Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022. 1

- [53] S. Toshniwal, W. Du, I. Moshkov, B. Kisacanin, A. Ayrapetyan, and I. Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*, 2024. 1, 2, 26
- [54] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 22
- [55] J. Vendrow, E. Vendrow, S. Beery, and A. Madry. Do large language model benchmarks test reliability? *arXiv preprint arXiv:2502.03461*, 2025. 3, 26
- [56] Y. Wang, Q. Yang, Z. Zeng, L. Ren, L. Liu, B. Peng, H. Cheng, X. He, K. Wang, J. Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025. 23
- [57] M. Xia, M. Artetxe, C. Zhou, X. V. Lin, R. Pasunuru, D. Chen, L. Zettlemoyer, and V. Stoyanov. Training trajectories of language models across scales. *arXiv preprint arXiv:2212.09803*, 2022. 1
- [58] A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024. 26
- [59] E. Yeo, Y. Tong, M. Niu, G. Neubig, and X. Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025. 1, 22
- [60] Ç. Yıldız, N. K. Ravichandran, N. Sharma, M. Bethge, and B. Ermiş. Investigating continual pre-training in large language models: Insights and implications. *arXiv preprint arXiv:2402.17400*, 2024. 2
- [61] L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023. 1, 2, 26
- [62] Y. Yue, Z. Chen, R. Lu, A. Zhao, Z. Wang, S. Song, and G. Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. 7, 9, 22
- [63] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019. 2
- [64] B. Zhang, Z. Liu, C. Cherry, and O. Firat. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*, 2024. 1, 9
- [65] H. Zhang, D. Morwani, N. Vyas, J. Wu, D. Zou, U. Ghai, D. Foster, and S. M. Kakade. How does critical batch size scale in pre-training? In *The Thirteenth International Conference on Learning Representations*, 2025. 1
- [66] P. Zhang, G. Zeng, T. Wang, and W. Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024. 22
- [67] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 22
- [68] R. Zhao, A. Meterez, S. Kakade, C. Pehlevan, S. Jelassi, and E. Malach. Echo chamber: RL post-training amplifies behaviors learned in pretraining. *arXiv preprint arXiv:2504.07912*, 2025. 1, 9, 22
- [69] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, Z. Feng, and Y. Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. 1
- [70] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023. 22

Appendices

Table of Contents

A Additional Related Work	22
----------------------------------	-----------

A.1 Scaling Laws for Language Models	22
--	----

A.2 Studies on Post-training for Reasoning	22
--	----

B Additional Experiment Results	22
--	-----------

B.1 Comparison of Pretrained Models	22
---	----

B.2 Scaling Up RL compute	22
-------------------------------------	----

B.3 Post-trained Models are Miscalibrated for Language Modeling Tasks	23
---	----

C Reproducibility	24
--------------------------	-----------

C.1 Model Architectures	24
-----------------------------------	----

C.2 Training Details	24
--------------------------------	----

C.3 Evaluation Details	26
----------------------------------	----

A Additional Related Work

A.1 Scaling Laws for Language Models

Early scaling work [22, 26] established fundamental relationships linking training loss to model size, data quantity, and compute. Recent studies have extended this framework in several ways. A dual-axis scaling law has shown reliable loss predictions even in highly over-trained regimes, significantly beyond traditional optimal compute points [15]. Additionally, new quantitative models predict emergent behaviors in model accuracy either through explicit loss thresholds or by probing with targeted finetuning [49, 13]. Cross-distribution transferability has also been modeled, allowing accurate extrapolations of loss curves between different datasets from minimal pilot data [7].

Further refinements address data-limited contexts, deriving optimal epoch allocation when unique training data is scarce [36], and revealing similar scaling patterns for synthetic data with clear diminishing returns [39]. Moreover, scaling laws now capture continual pre-training dynamics, providing guidance on mixing domain-specific and general data, and quantifying forgetting effects during domain adaptation with replay data [40]. Finally, research into compute allocation has developed scaling relationships specifically for distillation, determining precisely when distillation methods surpass direct pre-training efficiency [9].

A.2 Studies on Post-training for Reasoning

Recent research has explored how post-training strategies influence the reasoning capabilities of LLMs. One study challenges the “Superficial Alignment Hypothesis” [70], demonstrating that SFT post-training performance scales with the number of fine-tuning examples, akin to pre-training scaling laws [41]. Moreover, RL post-training has been shown to amplify behaviors acquired during pre-training, particularly in tasks requiring advanced mathematical reasoning and coding [68]. A comparative study indicates that while SFT tends to memorize training data, RL foster better generalization [10].

Investigations into the mechanics of reasoning have demystified long chain-of-thought learned through RL, identifying factors that enable the generation of extended reasoning trajectories [59]. Conversely, a critical examination questions whether RL truly incentivizes reasoning capacities beyond what is already learned during pre-training, suggesting that RL may not elicit fundamentally new reasoning patterns [62].

B Additional Experiment Results

B.1 Comparison of Pretrained Models

Table 1 compares our pretrained models against several open-weight models including OPT [67], Pythia [5], TinyLlama [66], Llama [54], and Qwen [3]. Our models, pretrained on a significantly smaller number of tokens (320B tokens for our 1B and 4B models), demonstrate competitive performance with other state-of-the-art small models such as TinyLlama-1B (trained on 2T tokens) and Qwen1.5-4B (trained on 3T tokens).

Specifically, despite TinyLlama-1B and Qwen1.5-4B models being trained with 6.25x and 9.38x more tokens respectively, our 1B and 4B models achieve similar or slightly better results across standard benchmarks like HellaSwag (H/S), Winogrande (W/G), PIQA, OBQA, ARC-Easy (ARC-E), and ARC-Challenge (ARC-C). This empirical observation is consistent with our experimental findings in Section 3.1, highlighting diminishing returns from excessive pretraining: beyond a certain optimal compute threshold, additional pretraining leads to minimal incremental gains in general domain upstream task performance.

B.2 Scaling Up RL compute

To further look into effective practice for scaling up RL compute, we plot results in “example-epochs” units ($\#examples \times \#epochs$, in 10^5) in Figure 10. We use the exact same configurations as Section 3.4. Under a fixed compute budget, allocating more epochs on a moderate dataset (e.g., $100K \times 8 = 800K$ example-epochs) typically yields higher ID and OOD performance than spreading

Name	Tokens	H/S	W/G	PIQA	OBQA	ARC-E	ARC-C	Avg.
OPT 1.3B	300B	53.65	59.59	72.36	33.40	50.80	29.44	49.87
Pythia 1.0B	300B	47.16	53.43	69.21	31.40	48.99	27.05	46.21
Pythia 1.4B	300B	52.01	57.38	70.95	33.20	54.00	28.50	49.34
TinyLlama 1B	2T	61.47	59.43	73.56	36.80	55.47	32.68	53.23
Llama3.2 1B	9T	63.66	60.46	74.54	37.00	60.48	35.75	55.31
Qwen3 1.7B	36T	60.46	61.01	72.36	36.80	69.91	43.26	57.30
1B (ours)	20B	42.25	51.30	67.85	32.80	54.80	29.61	46.44
	40B	47.53	54.62	69.59	36.20	58.08	30.29	49.38
	80B	51.05	53.59	70.78	37.20	62.71	35.92	51.88
	160B	52.30	53.99	71.71	36.60	63.09	36.09	52.30
	320B	53.86	53.51	71.93	37.20	62.29	36.18	52.49
Pythia 6.9B	300B	63.89	61.17	76.39	37.20	61.07	35.15	55.81
OPT 6.7B	300B	67.18	65.35	76.50	37.40	60.06	34.73	56.87
Qwen1.5 4B	3T	71.45	64.09	77.10	39.60	61.41	39.51	58.86
Qwen2.5 3B	18T	73.61	68.51	78.89	42.00	73.23	47.18	63.90
Qwen3 4B	36T	73.71	70.64	77.75	41.00	76.22	51.88	65.20
Llama 3.2 3B	9T	73.63	69.69	77.53	43.20	71.76	45.90	63.62
4B (ours)	80B	48.84	54.38	69.91	35.80	59.68	32.68	50.22
	160B	56.49	55.88	72.63	40.20	66.67	39.93	55.30
	320B	61.38	57.46	74.27	41.80	67.55	39.16	56.94

Table 4: **Upstream** benchmark comparison across various small-size LMs. All scores are percentages. We highlight our base model performance in **bold font**, models with performance at a comparable scale in **red**, and excessively over-trained models with similar performance in **green**.

compute over a larger dataset with fewer epochs, and RL with excessive training examples could sometimes lead to collapsed performance due to overly long and unfinished responses (shown by the crosses in Figure 10 and response length in Figure 11), while we do not observe such problems when conducting RL with excessive training epochs (shown in Figure 12). This demonstrates that deeper policy optimization per sample is more cost-effective than broader data coverage for RL scaling, which is consistent with findings proposed by [56] showing that RL using even only one training example could be effective in incentivizing the mathematical reasoning capabilities of LLMs.

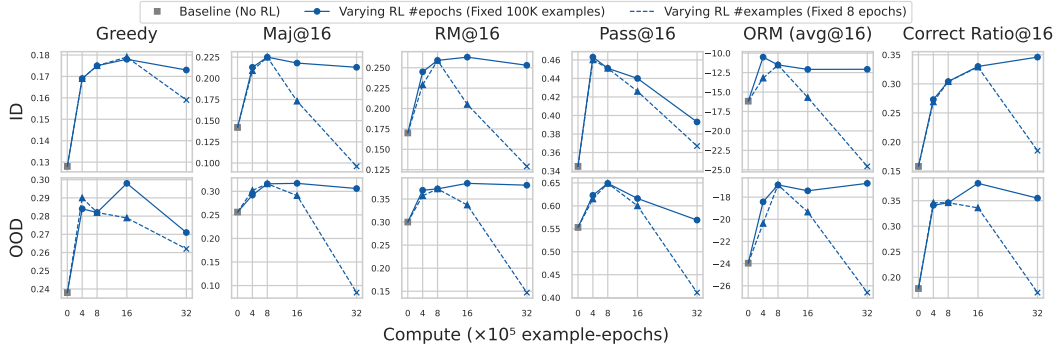


Figure 10: **Downstream** task performance vs. RL compute. A cross mark indicates models that tend to generate responses longer than their context window limits.

925 B.3 Post-trained Models are Miscalibrated for Language Modeling Tasks

926 Our upstream evaluations indicate that post-trained LMs exhibit significant miscalibration when
 927 assessed through validation PPL. We evaluate PPL on the validation set (disjoint from the training
 928 set) for each post-trained model. As illustrated in Figure 13, we observe negligible correlations
 929 between validation perplexity and downstream task accuracy across various datasets. Specifically, the

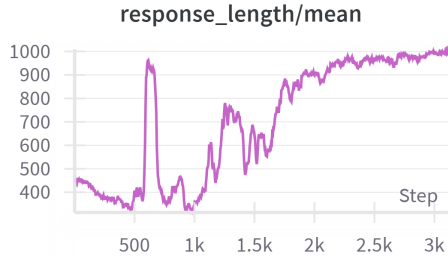


Figure 11: Response length versus training step when tuning 1B-160BT-8+42BT-100Kep1-400Kep8.

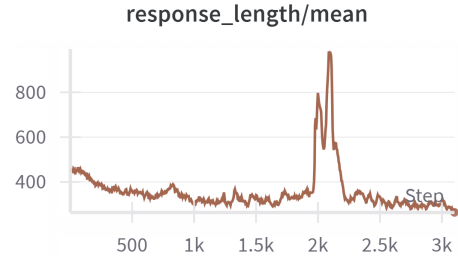


Figure 12: Response length versus training step when tuning 1B-160BT-8+42BT-100Kep1-100Kep32.

Pearson correlation coefficients remain close to zero, reinforcing that low perplexity does not reliably predict enhanced generative reasoning performance. This contrasts sharply with the strong predictive capability exhibited by ORM scores, as discussed in Section 4.2. While validation perplexity is conventionally used to monitor model quality, it is insufficient for post-training phases, particularly when evaluating generative reasoning tasks. In practice, relying solely on perplexity as a validation metric could misguide resource allocation decisions during training.

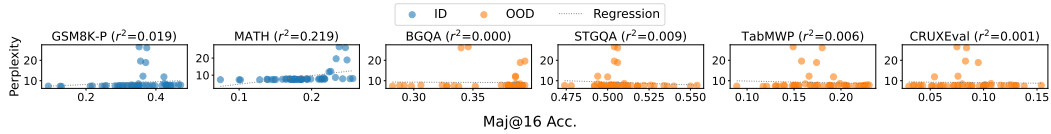


Figure 13: Correlation between accuracy and validation PPL across different tasks. Each subplot represents one dataset, where each point corresponds to a post-trained model variant. A dashed line indicates the linear trend, and the Pearson correlation coefficient is reported in each title.

C Reproducibility

C.1 Model Architectures

We show model architecture details for 0.5B, 1B and 4B models in Table 5.

Model Size	Hidden Size	Intermediate Size	Vocab Size	Context Length	# Heads	# Layers	# Query Groups
0.5B	1536	3216	32000	2048	32	20	4
1B	2048	4896	32000	2048	32	22	4
4B	4096	7792	32000	2048	32	28	4

Table 5: Model architecture details.

C.2 Training Details

C.2.1 Hyperparameters

Hyperparameters for pretraining/continued pretraining, SFT, and RL are shown in Table 6, Table 7, Table 8, respectively. We use the AdamW optimizer and up to 32 NVIDIA H100 80GB HBM3 GPUs for all training stages.

0.5B		1B		4B	
precision	bf16-mixed	precision	bf16-mixed	precision	bf16-mixed
global_batch_size	512	global_batch_size	512	global_batch_size	1024
max_seq_length	2048	max_seq_length	2048	max_seq_length	2048
lr_warmup_ratio	0.1	lr_warmup_ratio	0.1	lr_warmup_ratio	0.1
max_norm	1	max_norm	1	max_norm	1
lr	0.00025	lr	0.0002	lr	0.00015
min_lr	0.000025	min_lr	0.00002	min_lr	0.000015
weight_decay	0.1	weight_decay	0.1	weight_decay	0.1
beta1	0.9	beta1	0.9	beta1	0.9
beta2	0.95	beta2	0.95	beta2	0.95
epoch	1	epoch	1	epoch	1

Table 6: Hyperparameters for pretraining/continued pretraining.

1B		4B	
cutoff_len	2048	cutoff_len	2048
batch_size	128	batch_size	256
learning_rate	0.00001	learning_rate	0.0000075
lr_scheduler_type	cosine	lr_scheduler_type	cosine
warmup_ratio	0.1	warmup_ratio	0.1

Table 7: Hyperparameters for supervised finetuning.

1B		4B	
actor_lr	2.00E-06	actor_lr	1.00E-06
critic_lr	2.00E-05	critic_lr	1.00E-05
kl	0.0001	kl	0.0001
train_batch_size	1024	train_batch_size	2048
max_prompt_length	1024	max_prompt_length	1024
max_response_length	1024	max_response_length	1024
ppo_mini_batch_size	1024	ppo_mini_batch_size	2048
ppo_micro_batch_size_per_gpu	32	ppo_micro_batch_size_per_gpu	16
log_prob_micro_batch_size_per_gpu	64	log_prob_micro_batch_size_per_gpu	32
warmup_steps_ratio	0.1	warmup_steps_ratio	0.1

Table 8: Hyperparameters for reinforcement learning (PPO).

C.2.2 SFT/RL Template

We use the following template for SFT and RL tuning:

Human: {query}
Assistant: {response}

C.2.3 Training Data

FineWeb-Edu [37]: An extensive educational dataset sourced from web content, specifically designed for pretraining language models on high-quality academic and educational text. There are ~ 1.3 trillion tokens in total.

FineMath [37]: A curated dataset of mathematical texts, problems, and solutions, intended to enhance language models’ mathematical knowledge. There are ~ 50 billion tokens in total.

953 **OpenMathInstruct2** [53], **MetaMathQA** [61], **NuminaMath** [30]: Instruction-tuning datasets
954 containing mathematical questions paired with step-by-step solutions and explanations, designed
955 to improve the mathematical reasoning capabilities of LLMs. The responses corresponding to the
956 prompts from these datasets are collected by prompting the Qwen2.5-7B-Math-Instruct model [58].

957 C.3 Evaluation Details

958 C.3.1 Benchmarks and Sampling Parameters

959 For all test datasets and all models, we directly ask the models the corresponding questions applying
960 the same prompt template used for SFT/RL. We set the temperature to 0 for greedy decoding and 1
961 for decoding with randomness (the number of generations being 16), and set the repetition penalty to
962 1.1. We use the vLLM framework [28] for inference. Details of each test dataset are as follows.

963 **MATH** [19] is a large-scale benchmark designed to evaluate mathematical reasoning. It contains
964 12,500 challenging problems sourced from math competitions, categorized into seven topics including
965 Algebra, Geometry, Calculus, and Number Theory, and divided into 5 difficulty levels. Each
966 problem requires generating detailed, step-by-step solutions rather than simple numerical answers,
967 emphasizing comprehensive reasoning skills and logical deduction.

968 **GSM8K-Platinum** [55] is a manually cleaned and denoised version of **GSM8K** [12] which is a
969 math benchmark that consists of 8.5K high-quality, linguistically diverse grade-school math word
970 problems designed for multi-step reasoning (2 to 8 steps). Solutions involve elementary arithmetic
971 operations and require no concepts beyond early algebra. Its test set contains 1319 unique problems.

972 **BoardgameQA** [27] is a logical reasoning benchmark designed to evaluate language models’ ability
973 to reason with contradictory information using defeasible reasoning, where conflicts are resolved
974 based on source preferences (e.g., credibility or recency). Its test set contains 15K unique problems.

975 **CRUXEval** [18] is a benchmark for evaluating code reasoning, understanding, and execution,
976 featuring 800 Python functions (3-13 lines) with input-output pairs for input and output prediction
977 tasks. Given a function snippet and an input example, LLMs are tasked to generate the corresponding
978 outputs. Its test set contains 800 unique problems.

979 **StrategyQA** [17] is a commonsense question-answering benchmark designed for multi-hop reasoning
980 where the necessary reasoning steps are implicit and must be inferred using a strategy. Each of
981 the 2,780 examples includes a strategy question, its step-by-step decomposition, and supporting
982 Wikipedia evidence.

983 **TabMWP** [34] is a benchmark introduced to evaluate mathematical reasoning over tabular data. It
984 contains around 38,000 math word problems, each associated with relevant tables, spanning diverse
985 mathematical reasoning types like arithmetic operations, comparisons, and aggregation tasks.

986 C.3.2 Statistical Significance

987 In Appendix C.3.2, we show the standard errors of model performance on ID and OOD tasks for
988 all main results in Section 3, encompassing Maj@16, Pass@16, RM@16, ORM scores, and correct
989 ratios. The results are obtained from four experiments with different random seeds.

990 **C.3.3 Example Model Outputs**

991 The following shows an example input-output pair of the problem “GSM8KPlatinum-0772” using
992 the model 1B-160BT-8+42BT-100Kep1-100Kep8:

[input starts from here]Human: While on vacation in Bali, Thea bought a hat from a craftsman worth \$70. If she gave the craftsman four \$20 bills, how much change did she get?

Assistant:**[output starts from here]** To determine how much change Thea got, we need to follow these steps: 1. Calculate the total cost of the hat. 2. Subtract the amount she paid for the hat from the amount she received.

First, the cost of the hat is \$70. She gave the craftsman four \$20 bills. Therefore, the total amount she gave is: $4 \times 20 = 80$ dollars.

Next, we subtract the amount she paid for the hat from the total cost of the hat: $70 - 80$

Performing the subtraction: $70 - 80 = -10$ So, Thea got dollars in change. The final answer is: dollars. The change Thea got is dollars. Therefore, the answer is: dollars. The final answer is: dollars.

993

994