Train-before-Test Harmonizes Language Model Rankings

Anonymous Author(s)
Affiliation
Address
email

Abstract

Existing language model benchmarks provide contradictory model rankings, even for benchmarks capturing similar skills. This hampers model selection and adds confusion to the growing ecosystem of competing models. We propose a fundamental shift in evaluation methodology: rather than measuring out-of-the-box performance, we assess model potential—achievable performance after task-specific fine-tuning. Our *train-before-test* approach provides each model with identical benchmark-specific fine-tuning prior to evaluation. Our primary contribution is a comprehensive empirical evaluation of model potential across 24 benchmarks and 61 models. First, we demonstrate that model potential rankings through train-before-test exhibit remarkable consistency across all benchmarks. While traditional rankings show little external validity under direct evaluation, they enjoy significant external validity with train-before-test: model potential rankings transfer gracefully between benchmarks. Second, train-before-test restores the connection between perplexity and downstream task performance. For base models, even pre-fine-tuning perplexity predicts post-fine-tuning downstream performance, suggesting ranking consistency reflects inherent model potential rather than fine-tuning artifacts. Finally, train-before-test reduces the model-score matrix to essentially rank one, indicating model potential is dominated by one latent factor.

1 Introduction

Existing language model benchmarks provide contradictory model rankings, even for benchmarks capturing similar skills [43, 6, 21]. This inconsistency poses a fundamental challenge: how can we reliably compare and select models when different benchmarks yield conflicting assessments? While this ranking disagreement is often attributed to the diverse capability profiles of large language models [63], it creates practical confusion that hampers model development decisions [88].

The root problem lies in how we evaluate language models. Current practice follows *direct evaluation* measuring out-of-the-box performance. However, modern language models are pre-trained on diverse, often proprietary data mixtures [28, 58, 74, 29]. Recent work showed this leads to *training on the test task [19]:* the extent a model has encountered similar data during training confounds comparisons and rankings [37]. An otherwise worse model may have simply prepared better for a specific task.

We propose a fundamental shift: rather than measuring out-of-the-box performance, we assess *model potential*—achievable performance after task-specific fine-tuning. Our approach, *train-before-test*, provides each model with identical benchmark-specific fine-tuning prior to evaluation, leveling the playing field by ensuring equal task-specific preparation. The distinction between performance and potential is crucial. Direct evaluation measures immediate capabilities, while train-before-test measures what a model would achieve given equal preparation opportunity. This shift is particularly valuable for model development and adaptation scenarios. When practitioners select a base model

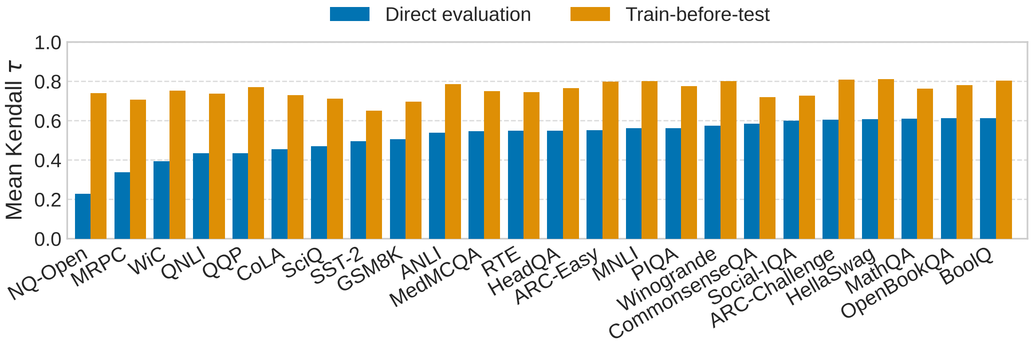


Figure 1: Mean ranking agreement between each benchmark and all others. We calculate Kendall's τ between each benchmark and every other benchmark, and then average the results. Compared to direct evaluation, train-before-test consistently improves ranking agreement, often by a large margin. A detailed comparison of Kendall's τ values for every benchmark pair is provided in Appendix C.1. On average, the overall average Kendall's τ is 0.52 for direct evaluation and 0.76 for train-before-test.

for fine-tuning or organizations make long-term infrastructure investments, understanding model potential becomes more informative than out-of-the-box performance. These stakeholders care less about current capabilities and more about future achievement with appropriate adaptation. See the discussion of related work in Appendix A.

Direct evaluation leads to ranking disagreement even between related tasks. We demonstrate that direct evaluation results in strong ranking disagreement across benchmarks, persisting even when restricting to similar tasks or models from the same family. This presents a serious conundrum: Under direct evaluation, benchmarks fail to give reliable insights for model selection.

Train-before-test leads to consistent model potential rankings. We comprehensively evaluate train-before-test across 24 benchmarks and 61 models. By fine-tuning each model on identical task-relevant data before evaluation, we uncover remarkably consistent model potential rankings. Ranking agreement between benchmarks improves for 274 out of 276 benchmark pairs, with average Kendall's τ increasing from 0.52 to 0.76. Figure 3 in Appendix illustrates an example. This consistency suggests model potential has external validity [65] and transfers across tasks.

Model potential aligns perplexity rankings with downstream tasks. Perplexity benchmarks fell out of fashion due to apparent disconnect with downstream performance [79, 24, 46, 49, 47]. We validate this disconnect under direct evaluation. However, our train-before-test approach restores the connection: post-fine-tuning perplexity rankings match post-fine-tuning downstream task rankings. For base models, even pre-fine-tuning perplexity predicts post-fine-tuning downstream performance, indicating ranking consistency reflects inherent model potential rather than fine-tuning artifacts.

Train-before-test sheds light on latent factors of benchmark scores. We show that the benchmark-model score matrix becomes essentially rank one under train-before-test. The first principal component accounts for 86% of explained variance across all models, and 93% for single model families. This suggests model potential is dominated by a single latent factor, while additional components in direct evaluation may reflect task-specific training exposure.

2 Experiments

Experiment setting. We begin our study with the `lm-eval-harness` package [25], which offers a comprehensive suite of language model benchmarks. We select 24 benchmarks covering diverse domains and task types. See Appendix B.1 for details. We consider 61 language models across six model families: Llama [28], Qwen [58], Gemma [74], Pythia [8], GPT-2 [59] and Yi [85]. Due to computational constraints, we select models with no more than 14B parameters. See Table 2 for the full list. We include both base and instruction-tuned models.

We evaluate 61 models across all 24 benchmarks using both direct evaluation and train-before-test evaluation. For direct evaluation, we evaluate models zero-shot as-is [11]. For train-before-test, we

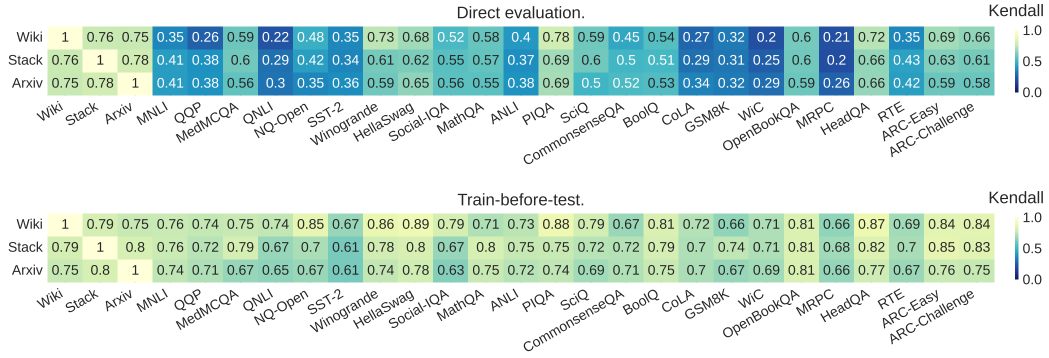


Figure 2: Ranking agreement between perplexity and downstream benchmarks under direct evaluation (top) and train-before-test (bottom). Perplexity rankings show strong internal consistency under both evaluation (avg. τ = 0.76 and 0.78). However, direct evaluation yields poor perplexity-downstream agreement (avg. τ = 0.48), but train-before-test dramatically improves it (avg. τ = 0.74).

fine-tune models using parameter-efficient fine-tuning (PEFT) [35, 50] and select the best checkpoint based on validation performance, yielding $61 \times 24 = 1,464$ fine-tuned models in total. Each fine-tuned model is then evaluated on the corresponding benchmark's test set. See more details in Appendix B.3. We rank models by performance on each benchmark and measure ranking correlation across benchmark pairs using Kendall's τ [38].

Downstream ranking agreement. As depicted in Figure 1, direct evaluation shows only modest ranking agreement between the 24 benchmarks, with an average Kendall's τ ranking correlation of 0.52. This lack of agreement across benchmarks complicates model assessment and makes it challenging to aggregate results into a meaningful overall ranking [88]. In contrast, the train-before-test methodology leads to a substantial improvement in ranking agreement. Under this approach, 274 out of 276 benchmark pairs show higher Kendall's τ scores, with the average τ rising from 0.52 to 0.76. This stronger consistency suggests that model potential measured on one benchmark is likely to generalize to others, including practitioners' own cases, which simplifies model comparison and selection. We further show that direct evaluation yields poor ranking consistency both within and across benchmark categories, while train-before-test significantly improves both intra- and inter-category agreement in Appendix C.1.

Perplexity agreement. We now compare downstream benchmark rankings with perplexity rankings on three general-domain corpora. We collect three corpora from `Wikipedia`, `Stack`Exchange, and `arXiv`, retaining only contents from 2025 to ensure models could not have seen these texts during pretraining. We measure perplexity in bits per byte with `lm-eval-harness`, and compare the perplexity rankings with the downstream benchmark rankings considered earlier. See Appendix C.2.

Figure 2 presents our main results. Perplexity rankings demonstrate strong internal consistency under both evaluation schemes (average Kendall's τ of 0.76 and 0.78), likely due to the smooth relationship between perplexity evaluations [10, 51]. However, agreement between perplexity and downstream benchmarks is poor under direct evaluation (τ = 0.48), signaling a disconnect between the language modeling objective and benchmark performance.

Crucially, train-before-test substantially improves ranking agreement, raising the mean Kendall's τ to 0.74—comparable to agreement across downstream evaluations themselves. This suggests that light fine-tuning on task data effectively aligns the language modeling objective with downstream performance, making perplexity as effective for ranking as traditional benchmarks.

Figure 3 examines whether pre-fine-tuning perplexity predicts post-fine-tuning downstream performance. For base models, the correlation is strong (average τ = 0.78), indicating that direct perplexity evaluation reliably ranks model potential. However, instruction-tuned models show much weaker correlation (τ = 0.51), as instruction-tuning tends to increase both benchmark performance (↑) and perplexity (↓) on general text corpora, clouding their relationship. Fortunately, train-before-test restores high ranking agreement for these models as well as shown earlier.

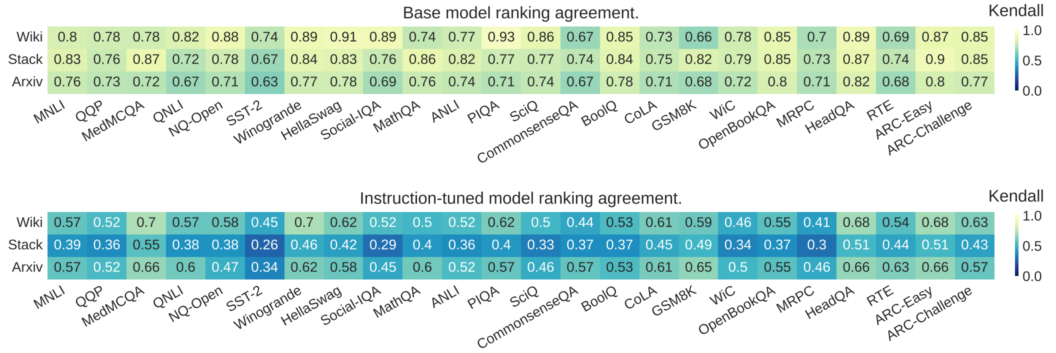


Figure 3: Ranking agreement between perplexity rankings **before fine-tuning** (direct evaluation) and downstream benchmark rankings **after fine-tuning** (train-before-test) for base models (top) and instruction-tuned models (bottom). Base models show strong correlation (average Kendall's $\tau = 0.78$), while instruction-tuned models show much weaker correlation (average Kendall's $\tau = 0.51$).

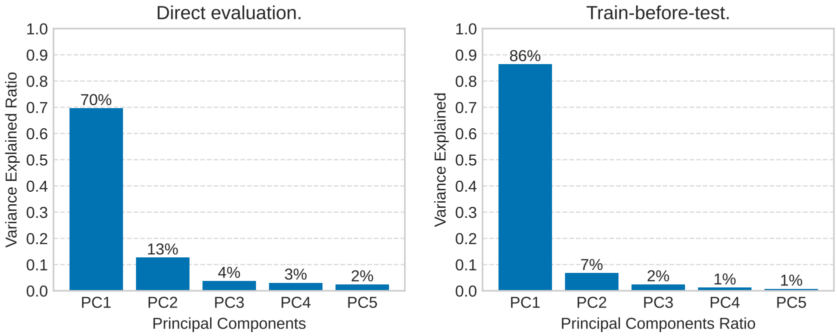


Figure 4: Explained variance ratios of the top five principal components of the benchmark score matrix, under direct evaluation (left) and train-before-test (right). Train-before-test substantially increases the explained variance by the first principal component, from 70% to 86%.

Low-ranked model score matrix. So far, we have shown that evaluating model potential using the train-before-test methodology yields consistent rankings across benchmarks. We now examine the implications of this finding by analyzing the resulting matrix of model scores, where each entry (i, j) corresponds to the performance of model j on a benchmark i. We use Principal Component Analysis (PCA) to examine the structure of the matrix of model scores.

Figure 4 shows the explained variance ratios of the first five principal components. These results support previous findings that the score matrix is of low rank [63]. Under direct evaluation, the first five components account for 91% of the total variance. A similar trend is observed for train-before-test scores, where the first five components explain 97% of the variance. Notably, under train-before-test, the first principal component (PC1) captures a significantly larger share of the variance: 86%, compared to 70% for direct evaluation. This shows that the model potential is dominated by one single principal axis. In Appendix C.3 we show that PC1 correlates positively with pre-training compute [37, 34], and in Appendix C.4 we show that conducting PC1 only on QWEN models increases PC1's explained variance to 93%, making the score matrix essentially rank one.

3 Conclusion

We proposed evaluating model potential through train-before-test, addressing the fundamental problem of contradictory rankings across benchmarks. Our work recommends making train-before-test a default component of LLM benchmarking. Train-before-test complements direct evaluation: direct evaluation gauges deployment readiness, while train-before-test reveals adaptability. Together they provide a complete view of model capabilities. See more discussion in Appendix E.

References

[1] Ehab A AlBadawy, Ashirbani Saha, and Maciej A Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical physics*, 45(3):1150–1158, 2018.

[2] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *Transactions on Machine Learning Research*, 2025.

[3] Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms, 2019.

[4] Kenneth J. Arrow. *Social Choice and Individual Values*. Wiley, 1951.

[5] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019.

[6] Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023.

[7] Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. The fifth PASCAL recognizing textual entailment challenge. In *Text Analysis Conference (TAC)*, 2009.

[8] Stella Biderman, Hailey Schoelkopf, Quentin G. Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. *ArXiv*, abs/2304.01373, 2023.

[9] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019.

[10] David Brandfonbrener, Nikhil Anand, Nikhil Vyas, Eran Malach, and Sham Kakade. Loss-to-loss prediction: Scaling laws for all datasets. *arXiv preprint arXiv:2411.12925*, 2024.

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

[12] J Quiñonero Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. Dataset shift in machine learning. *The MIT Press*, 1:5, 2009.

[13] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019.

[14] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

[15] Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021.

[16] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer, 2006.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[18] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*, 2005.

[19] Ricardo Dominguez-Olmedo, Florian E Dorner, and Moritz Hardt. Training on the test task confounds evaluation and emergence. *arXiv preprint arXiv:2407.07890*, 2024.

[20] Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. Understanding emergent abilities of language models from the loss perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[21] Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open llm leaderboard v2. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`, 2024.

[22] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander J. Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alexandros G. Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. *ArXiv*, abs/2304.14108, 2023.

[23] Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean-Pierre Mercat, Alex Fang, Jeffrey Li, Sedrick Scott Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Jenia Jitsev, Alexandros G. Dimakis, Gabriel Ilharco, Shuran Song, Thomas Kollar, Yair Carmon, Achal Dave, Reinhard Heckel, Niklas Muennighoff, and Ludwig Schmidt. Language models scale reliably with over-training and on downstream tasks. *ArXiv*, abs/2403.08540, 2024.

[24] Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, et al. Predictability and surprise in large generative models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1747–1764, 2022.

[25] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023.

[26] Adhiraj Ghosh, Sebastian Dziadzio, Ameya Prabhu, Vishaal Udandarao, Samuel Albanie, and Matthias Bethge. Onebench to test them all: Sample-level benchmarking over open-ended capabilities. *arXiv preprint arXiv:2412.06745*, 2024.

[27] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics, 2007.

[28] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[29] Etash Kumar Guha, Ryan Marten, Sedrick Scott Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean-Pierre Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Ben Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models. *ArXiv*, abs/2506.04178, 2025.

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

[31] Moritz Hardt. The emerging science of machine learning benchmarks. Online at `https://mlbenchmarks.org`, 2025. Manuscript.

[32] Moritz Hardt and Benjamin Recht. *Patterns, predictions, and actions: Foundations of machine learning*. Princeton University Press, 2022.

[33] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300, 2020.

[34] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. Training compute-optimal large language models. *ArXiv*, abs/2203.15556, 2022.

[35] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021.

[36] Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

[37] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[38] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[39] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

[40] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019.

[41] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[42] Hector J Levesque, Ernest Davis, and Leora Morgenstern. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46, page 47, 2011.

[43] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525:140 – 146, 2023.

[44] Thomas Liao, Rohan Taori, Inioluwa Deborah Raji, and Ludwig Schmidt. Are we learning yet? a meta review of evaluation failures across machine learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[45] Mark Liberman. Obituary: Fred jelinek. *Computational Linguistics*, 36(4):595–599, 2010.

[46] Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models. In *International Conference on Machine Learning*, pages 22188–22214. PMLR, 2023.

[47] Nicholas Lourie, Michael Y. Hu, and Kyunghyun Cho. Scaling laws are unreliable for downstream tasks: A reality check. *ArXiv*, abs/2507.00885, 2025.

[48] Nicholas Lourie, Michael Y Hu, and Kyunghyun Cho. Scaling laws are unreliable for downstream tasks: A reality check. *arXiv preprint arXiv:2507.00885*, 2025.

[49] Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, A. Jha, Oyvind Tafjord, Dustin Schwenk, Pete Walsh, Yanai Elazar, Kyle Lo, Dirk Groeneveld, Iz Beltagy, Hanna Hajishirzi, Noah A. Smith, Kyle Richardson, and Jesse Dodge. Paloma: A benchmark for evaluating language model fit. *ArXiv*, abs/2312.10523, 2023.

[50] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

[51] Prasanna Mayilvahanan, Thaddäus Wiedemer, Sayak Mallick, Matthias Bethge, and Wieland Brendel. Llms on the line: Data determines loss-to-loss scaling laws. In *Forty-second International Conference on Machine Learning*, 2025.

[52] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[53] John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution shift on question answering models. In *International conference on machine learning*, pages 6905–6916. PMLR, 2020.

[54] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding, 2020.

[55] OpenAI. Gpt-4 technical report. *arXiv*, 2023.

[56] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering, 2022.

[57] Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*, 2018.

[58] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[59] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. In *OpenAI Technical Report*, 2019. OpenAI technical report.

[60] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*, pages 2383–2392. Association for Computational Linguistics, 2016.

[61] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ArXiv*, abs/2102.12092, 2021.

[62] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet?, 2019.

[63] Yangjun Ruan, Chris J. Maddison, and Tatsunori B. Hashimoto. Observational scaling laws and the predictability of language model performance. *ArXiv*, abs/2405.10938, 2024.

[64] Olawale Salaudeen and Moritz Hardt. Imagenot: A contrast with imagenet preserves model rankings. *arXiv preprint arXiv:2404.02112*, 2024.

[65] Olawale Salaudeen, Anka Reuel, Ahmed M. Ahmed, Suhana Bedi, Zachary Robertson, Sudharsan Sundar, Ben Domingue, Angelina Wang, and Oluwasanmi Koyejo. Measurement to meaning: A validity-centered framework for ai evaluation. *ArXiv*, abs/2505.10573, 2025.

[66] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions, 2019.

[67] Tal Shnitzer, Anthony Ou, M'irian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *ArXiv*, abs/2309.15789, 2023.

[68] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642, 2013.

[69] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv*, abs/2206.04615, 2022.

[70] Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Annual Meeting of the Association for Computational Linguistics*, 2022.

[71] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019.

[72] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

[73] Gemini Team. Gemini: A family of highly capable multimodal models. *arXiv*, 2023.

[74] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[75] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[76] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From imagenet to image classification: Contextualizing progress on benchmarks. In *International Conference on Machine Learning*, 2020.

[77] David Vilares and Carlos Gómez-Rodríguez. HEAD-QA: A healthcare dataset for complex reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 960–966, Florence, Italy, July 2019. Association for Computational Linguistics.

[78] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *arXiv preprint 1805.12471*, 2018.

[79] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

[80] Laura Weidinger, Deborah Raji, Hanna Wallach, Margaret Mitchell, Angelina Wang, Olawale Salaudeen, Rishi Bommasani, Sayash Kapoor, Deep Ganguli, Sanmi Koyejo, and William Isaac. Toward an evaluation science for generative ai systems. *ArXiv*, abs/2503.05336, 2025.

[81] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions, 2017.

[82] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, 2018.

[83] Mengzhou Xia, Mikel Artetxe, Chunting Zhou, Xi Victoria Lin, Ramakanth Pasunuru, Danqi Chen, Luke Zettlemoyer, and Veselin Stoyanov. Training trajectories of language models across scales. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[84] Chhavi Yadav and Léon Bottou. Cold case: The lost mnist digits. In *Neural Information Processing Systems*, 2019.

[85] 01.AI Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.ai. *ArXiv*, abs/2403.04652, 2024.

[86] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019.

[87] Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets llm finetuning: The effect of data, model and finetuning method. *ArXiv*, abs/2402.17193, 2024.

[88] Guanhua Zhang and Moritz Hardt. Inherent trade-offs between diversity and stability in multi-task benchmarks. *arXiv preprint arXiv:2405.01719*, 2024.

[89] Jieyu Zhang, Weikai Huang, Zixian Ma, Oscar Michel, Dong He, Tanmay Gupta, Wei-Chiu Ma, Ali Farhadi, Aniruddha Kembhavi, and Ranjay Krishna. Task me anything. *ArXiv*, abs/2406.11775, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper is mainly an empirical work and doesn't provide many new theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Appendix B.1 and B.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release our codes in the supplemental materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Appendix B.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix B.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Authors have reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix F.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper doesn't release any new data or model.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All used models and datasets are well cited in Section 2.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper doesn't provide new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper doesn't involve crowd-sourcing experiments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper doesn't involve crowd-sourcing experiments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Related Work

Benchmarking has played a central role in the advancement of machine learning [45, 32]. While absolute model performance is often fragile to even seemingly minor changes in evaluation data [12, 75, 1, 72, 76, 53], relative model performance—that is, model rankings—tends to transfer surprisingly well across classical benchmarks [84, 62, 53]. For instance, prior work [40, 5] has shown that model rankings on ImageNet [17] also transfer to other image classification and object recognition benchmarks. Moreover, Salaudeen and Hardt (2024, [64]) demonstrated that ImageNet rankings remain robust even under major dataset variations [64]. This transferability of model rankings is highly desirable, as it indicates that progress on specific benchmarks reliably reflects broader scientific advancements [44, 31].

However, the emergence of foundation models has dramatically transformed the benchmarking landscape compared to the ImageNet era [43, 69, 80]. With huge training costs and much improved capabilities [58, 74, 28, 61, 73, 55], practitioners now lean towards directly evaluating LLMs across a wide range of different benchmarks, in the hope of obtaining a more comprehensive assessment of their capabilities [43, 70, 33, 6, 21]. This shift introduces new challenges, as model rankings across different tasks may vary significantly [43, 36, 48]. Zhang and Hardt (2024, [88]) draw an analogy between multi-task benchmarks and voting systems [4], revealing that a multi-benchmarking approach with diverse rankings inherently lacks robustness to minor changes and thus cannot provide a stable unified ranking.

This lack of unified ranking is sometimes seen as a desirable feature within the community [43]. Some argue that variability reflects the multifaceted strengths and weaknesses of LLMs, suggesting that users should select the best model tailored to their specific needs [26, 89, 67]. For example, a user who focuses on mathematical tasks could prioritize the math benchmark to choose the optimum model. However, there are two significant concerns regarding this approach: First, the user-driven selection strategy poses challenges for model developers. Given the resource-intensive nature of LLM development [30], it is impractical to release a different model for every potential use case. Moreover, developers typically aim to create a general-purpose model [58, 28, 74]; however, such a desideratum is often difficult to reliably measure due to the inconsistent rankings observed across benchmarks. Second, we demonstrate in this paper that benchmarks within the same task category can still exhibit substantial discrepancies in model rankings.

One potential reason for the observed inconsistencies in model rankings is that models vary substantially in their training data [22, 2]. In particular, Dominguez et al. (2024, [19]) show that models vary in their degree of preparedness for popular benchmarks. Building on this idea, we introduce the notion of train-before-test, wherein we fine-tune each model on the corresponding training set to get every model well-prepared. We then investigate how this approach improves the consistency of rankings across benchmarks and discuss its implications for future benchmarking practices.

Table 1: We categorize benchmarks into language understanding (LU), commonsense reasoning (CR), question answering (QA), physics/biology/chemistry (PBC), math (Math), and medicine (Med).

Category	Benchmarks
LU	MNLI [82], QNLI [60], RTE [16, 27, 7], CoLA [78], SST-2 [68], MRPC [18], QQP, WiC [57], ANLI [54]
CR	Winogrande [42], CommonsenseQA [71], Hellaswag [86], Social-IQA [66]
QA	OpenBookQA [52], NQ-Open [41], BoolQ [13], ARC-Easy, ARC-Challenge [14]
PBC	SciQ [81], PIQA [9]
Math	MathQA [3], GSM8K [15]
Med	MedMCQA [56], HeadQA [77]

B Additional Experiment Setting

B.1 Benchmark Selection

We begin our study with the lm-eval-harness package [25], which offers a comprehensive suite of language model benchmarks. To accommodate the train-before-test methodology which requires a dedicated training set for fine-tuning, we first identify benchmarks that provide at least 1,000 training examples. This yields a total of 37 benchmarks, which we broadly categorize into 28 likelihood-based and 9 generation-based benchmarks.

Likelihood-based evaluations test for the likelihood of different completions given some input string; for example, different answer choices given a multiple-choice input question. Since the number of completions is usually small, likelihood-based evaluations are generally compute-efficient. Generation-based evaluations, in contrast, generate some output response given an input query. If responses tend to be long, then generation-based evaluations naturally become compute-intensive. This is particularly true for base models, which are usually not trained for instruction following and therefore continue to generate tokens until hitting their maximum token limit. These generation-based benchmarks are also over-challenging for smaller models with limited parameters, such as GPT-2 [59]. Therefore, we exclude seven generation-based benchmarks, Drop, CoQa, ReCoRD, bAbi, WebQA, TriviaQA and Fld-Default. Nevertheless, we retain two widely used generation-based benchmarks, GSM8K and NQ-Open, in our experiments.

We additionally excluded five benchmarks due to anomalies observed during fine-tuning: MedQA-4Options, LogiQA, Mutual, Mela-EN, and Swag. For these benchmarks, more than 20% of models showed no performance improvement after fine-tuning. We also excluded Paws-EN, as its corresponding model ranking under direct evaluation was negatively correlated (Kendall's τ less than zero) with 23 out of 24 other benchmarks. We attribute this anomaly to the unusual prompting template used by lm-eval-harness.

Our final selection consists of 24 benchmarks covering diverse domains and task types. These benchmarks are primarily multiple-choice question answering benchmarks, with accuracy as the task metric. We categorize all benchmarks by their descriptions, see Table 1.

If a benchmark does not come with a validation split, we randomly allocate 20% of the training data as the validation set. To save computational resources, we cap the number of training data at 50,000, validation data at 1,000, and testing data at 10,000.

B.2 Model Selection

See Table 2 for the complete list of models used in our experiments.

B.3 Evaluation Setup

For our train-before-test evaluations, we fine-tune each model for five epochs and select the best-performing checkpoint based on evaluations on a separate validation set. We use the AdamW optimizer with a weight decay of 0.01. For each model-benchmark combination, we perform a hyperparameter search over three learning rates $\{1e-5, 2e-5, 5e-5\}$ and select the optimal one based on validation performance. To reduce memory consumption, we employ parameter-efficient

Table 2: Models considered, categorized by model family.

Family	Model Name Suffix
Llama- [28]	3-8B, 3.1-8B, 3.2-1B, 3.2-3B, 3-8B-IT, 3.1-8B-IT, 3.2-1B-IT, 3.2-3B-IT
Qwen- [58]	1.5-0.5B, 1.5-1.8B, 1.5-4B, 1.5-7B, 1.5-14B, 2-0.5B, 2-1.5B, 2-7B, 2.5-0.5B, 2.5-1.5B, 2.5-3B, 2.5-7B, 2.5-14B, 1.5-0.5B-IT, 1.5-1.8B-IT, 1.5-4B-IT, 1.5-7B-IT, 1.5-14B-IT, 2-0.5B-IT, 2-1.5B-IT, 2-7B-IT, 2.5-0.5B-IT, 2.5-1.5B-IT, 2.5-3B-IT, 2.5-7B-IT, 2.5-14B-IT
Gemma- [74]	2B, 7B, 2-2B, 2-9B, 2B-IT, 7B-IT, 2-2B-IT, 2-9B-IT
GPT2- [59]	124M, 335M, 774M, 1.5B
Pythia- [8]	70M, 160M, 410M, 1B, 1.4B, 2.8B, 6.9B, 12B
Yi- [85]	6B, 9B, 6B-IT, 1.5-6B, 1.5-9B, 1.5-6B-IT, 1.5-9B-IT

fine-tuning (PEFT) [35, 50], We use a LoRA configuration with rank 8, $\alpha = 32$, and dropout 0.1. Most of our experiments are conducted on Quadro RTX 6000, Tesla V100-SXM2-32GB and NVIDIA A100-SXM4-80GB GPUs.

In cases where models show no performance improvement after fine-tuning, we report their pre-fine-tuning results. This scenario is rare and typically occur with smaller models (less than 500M parameters) that lack the capacity to perform certain tasks, resulting in near-random performance both before and after fine-tuning. Additionally, since all training datasets in our study are publicly available, some models may have encountered this data during pre-training, potentially limiting the benefits of additional fine-tuning.

For instruction-tuned models, we evaluate performance both with and without chat templates, selecting the configuration that yields better results. Specifically, during direct evaluation, we assess model performance on the validation set under both conditions and apply the better-performing configuration to the test set. In the train-before-test setting, we similarly fine-tune two variants: one with training data formatted using chat templates and one without. We then select the approach that achieves the best performance on the validation set for final evaluation.

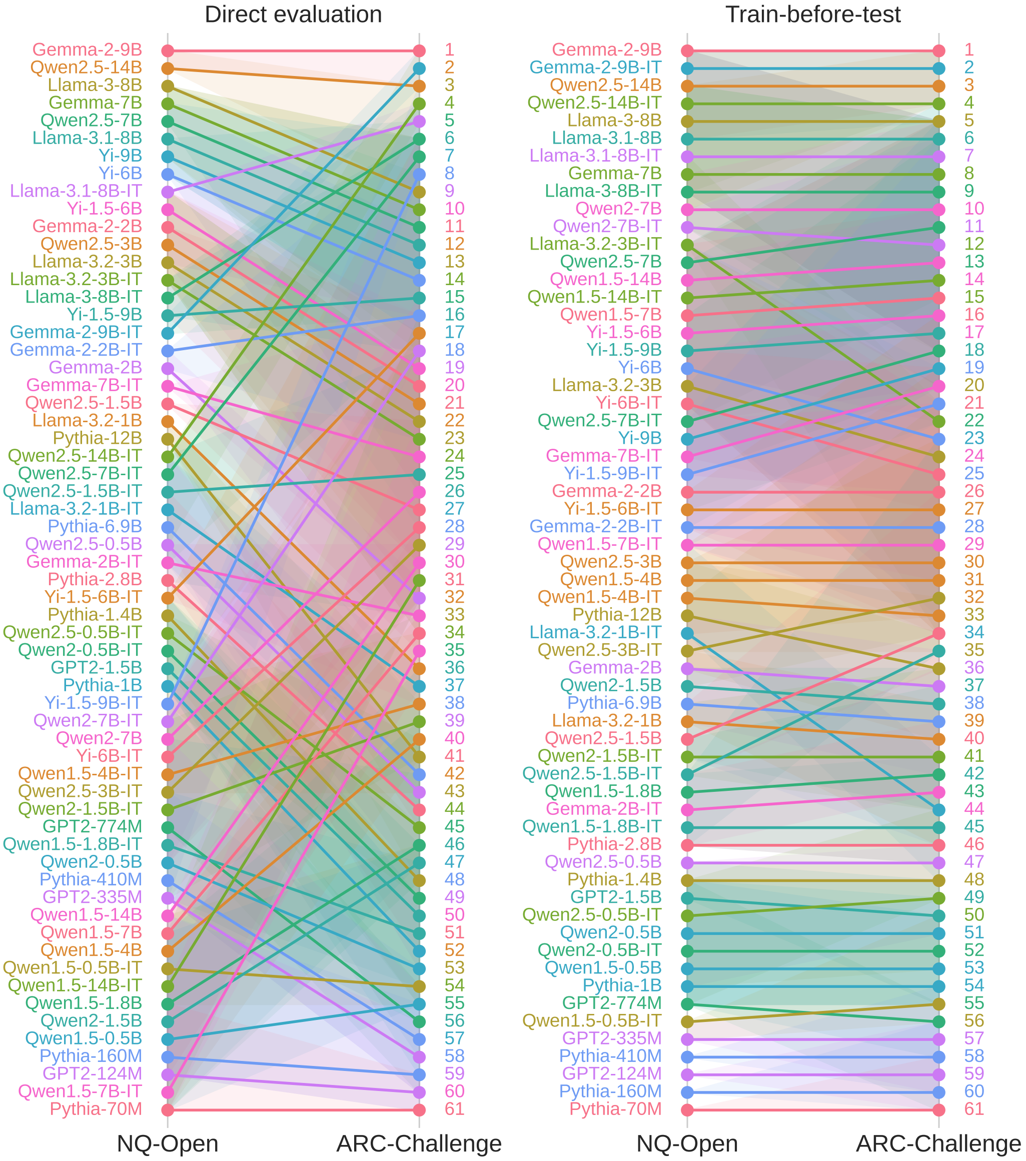
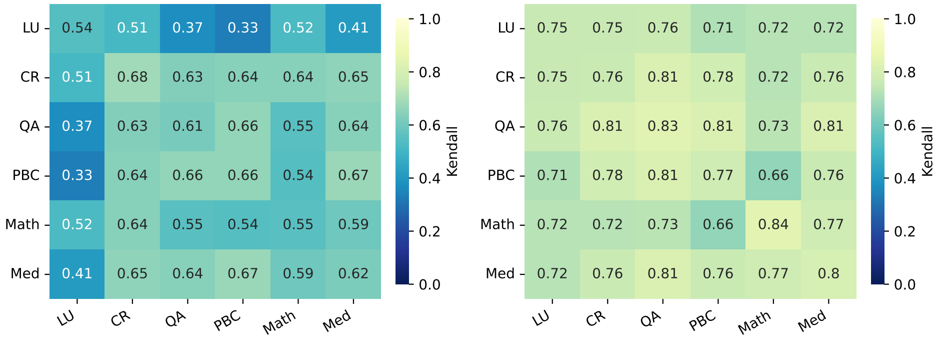


Figure 5: Rankings of 61 language models on two question-answering benchmarks: Natural Questions Open and ARC Challenge. **Left:** Direct evaluation leads to inconsistent rankings. Although both benchmarks test for question-answering ability, the resulting model rankings show substantial disagreement. **Right:** Train-before-test aligns model rankings. **Note:** For each of the two plots, we greedily align model rankings as much as possible without violating confidence intervals, thus revealing only those ranking changes that are statistically significant. See Appendix D.1 for details.



(a) Direct evaluation.

(b) Train-before-test.

Figure 6: Cross-category ranking agreement for direct evaluation (left) and train-before-test (right). We categorize benchmarks into language understanding (LU), commonsense reasoning (CR), question answering (QA), physics/biology/chemistry (PBC), math (Math), and medicine (Med), see Table 1. Kendall's τ is averaged across all pairs of benchmarks that belong to two given categories. The diagonal entries represent intra-category agreement and the other entries represent inter-category agreement. Train-before-test improves both intra- and inter-category ranking agreement in all instances.

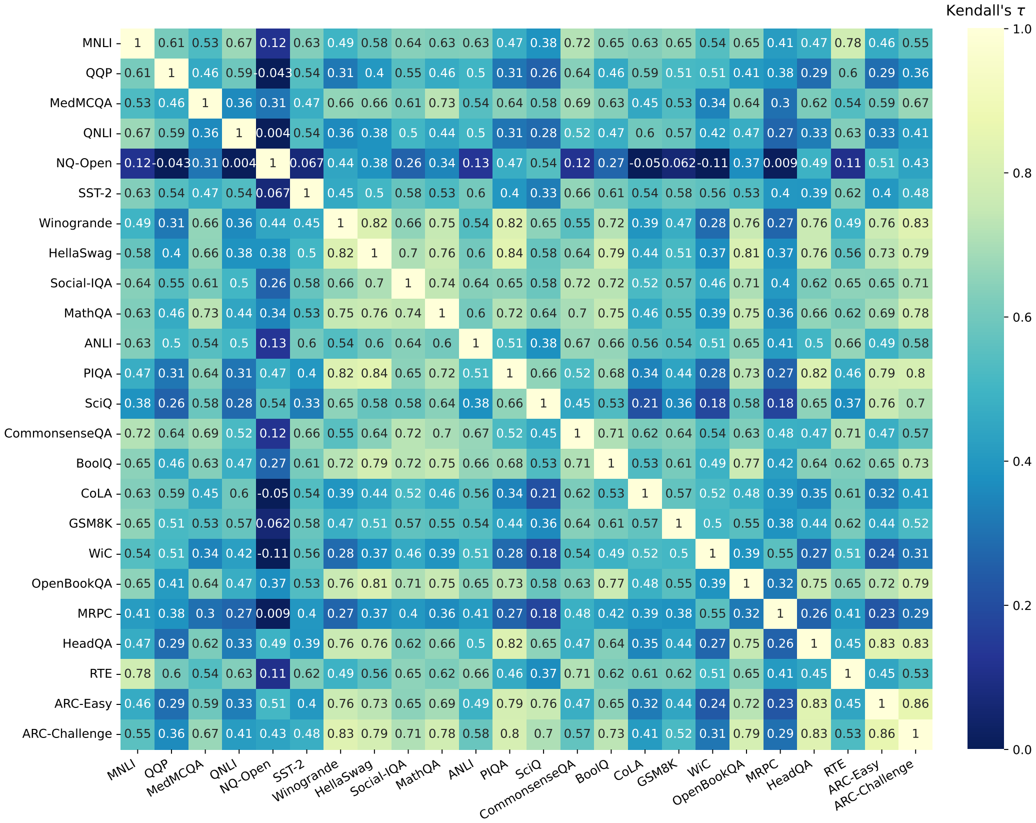
C Additional Experiment Results

C.1 Downstream Ranking Agreement

We further split all benchmarks into six categories (e.g., language understanding, math), see Table 1. For each category pair, we report in Figure 6 the intra-category average ranking correlations and inter-category average ranking correlations across all relevant benchmark pairs. Consistent with our previous findings, we observe reasonably poor ranking agreements across categories under direct evaluation. While one might expect high intra-category agreement—after all, tasks within the same category tend to be relatively similar—direct evaluation results in low intra-category agreement in many cases. For example, the intra-category mean Kendall's τ is 0.54 for language understanding and 0.55 for math. This further underscores the difficulty of selecting models based on direct evaluation. Even if the goal is to choose a model that excels not across all tasks but within a specific domain, the low intra-category agreement makes this decision challenging.

In contrast, train-before-test boosts both intra- and inter-category consistency. For example, the intra-category mean Kendall's τ for language understanding raises from 0.52 to 0.75, as well as from 0.55 to 0.84 for the math category. Moreover, agreement between categories is often nearly as high as agreement within categories. This suggests that models with higher potential in one domain tend to also perform well across other domains after adaptation.

We plot detailed pairwise ranking correlation agreement between benchmarks in Figure 7 (direct evaluation) and 8 (train-before-test), corresponding to Figure 1 in the main text.



	MNLI	QQP	MedMCQA	QNLI	NQ-Open	SST-2	Winogrande	HellaSwag	Social-IQA	MathQA	ANLI	PIQA	SciQ	CommonsenseQA	BoolQ	CoLA	GSM8K	WiC	OpenBookQA	MRPC	HeadQA	RTE	ARC-Easy	ARC-Challenge
MNLI	1	0.61	0.53	0.67	0.12	0.63	0.49	0.58	0.64	0.63	0.63	0.47	0.38	0.72	0.65	0.63	0.65	0.54	0.65	0.41	0.47	0.78	0.46	0.55
QQP	0.61	1	0.46	0.59	-0.043	0.54	0.31	0.4	0.55	0.46	0.5	0.31	0.26	0.64	0.46	0.59	0.51	0.51	0.41	0.38	0.29	0.6	0.29	0.36
MedMCQA	0.53	0.46	1	0.36	0.31	0.47	0.66	0.66	0.61	0.73	0.54	0.64	0.58	0.69	0.63	0.45	0.53	0.34	0.64	0.3	0.62	0.54	0.59	0.67
QNLI	0.67	0.59	0.36	1	0.004	0.54	0.36	0.38	0.5	0.44	0.5	0.31	0.28	0.52	0.47	0.6	0.57	0.42	0.47	0.27	0.33	0.63	0.33	0.41
NQ-Open	0.12	-0.043	0.31	0.004	1	0.067	0.44	0.38	0.26	0.34	0.13	0.47	0.54	0.12	0.27	-0.05	0.062	-0.11	0.37	0.009	0.49	0.11	0.51	0.43
SST-2	0.63	0.54	0.47	0.54	0.067	1	0.45	0.5	0.58	0.53	0.6	0.4	0.33	0.66	0.61	0.54	0.58	0.56	0.53	0.4	0.39	0.62	0.4	0.48
Winogrande	0.49	0.31	0.66	0.36	0.44	0.45	1	0.82	0.66	0.75	0.54	0.82	0.65	0.55	0.72	0.39	0.47	0.28	0.76	0.27	0.76	0.49	0.76	0.83
HellaSwag	0.58	0.4	0.66	0.38	0.38	0.5	0.82	1	0.7	0.76	0.6	0.84	0.58	0.64	0.79	0.44	0.51	0.37	0.81	0.37	0.76	0.56	0.73	0.79
Social-IQA	0.64	0.55	0.61	0.5	0.26	0.58	0.66	0.7	1	0.74	0.64	0.65	0.58	0.72	0.72	0.52	0.57	0.46	0.71	0.4	0.62	0.65	0.65	0.71
MathQA	0.63	0.46	0.73	0.44	0.34	0.53	0.75	0.76	0.74	1	0.6	0.72	0.64	0.7	0.75	0.46	0.55	0.39	0.75	0.36	0.66	0.62	0.69	0.78
ANLI	0.63	0.5	0.54	0.5	0.13	0.6	0.54	0.6	0.64	0.6	1	0.51	0.38	0.67	0.66	0.56	0.54	0.51	0.65	0.41	0.5	0.66	0.49	0.58
PIQA	0.47	0.31	0.64	0.31	0.47	0.4	0.82	0.84	0.65	0.72	0.51	1	0.66	0.52	0.68	0.34	0.44	0.28	0.73	0.27	0.82	0.46	0.79	0.8
SciQ	0.38	0.26	0.58	0.28	0.54	0.33	0.65	0.58	0.58	0.64	0.38	0.66	1	0.45	0.53	0.21	0.36	0.18	0.58	0.18	0.65	0.37	0.76	0.7
CommonsenseQA	0.72	0.64	0.69	0.52	0.12	0.66	0.55	0.64	0.72	0.7	0.67	0.52	0.45	1	0.71	0.62	0.64	0.54	0.63	0.48	0.47	0.71	0.47	0.57
BoolQ	0.65	0.46	0.63	0.47	0.27	0.61	0.72	0.79	0.72	0.75	0.66	0.68	0.53	0.71	1	0.53	0.61	0.49	0.77	0.42	0.64	0.62	0.65	0.73
CoLA	0.63	0.59	0.45	0.6	-0.05	0.54	0.39	0.44	0.52	0.46	0.56	0.34	0.21	0.62	0.53	1	0.57	0.52	0.48	0.39	0.35	0.61	0.32	0.41
GSM8K	0.65	0.51	0.53	0.57	0.062	0.58	0.47	0.51	0.57	0.55	0.54	0.44	0.36	0.64	0.61	0.57	1	0.5	0.55	0.38	0.44	0.62	0.44	0.52
WiC	0.54	0.51	0.34	0.42	-0.11	0.56	0.28	0.37	0.46	0.39	0.51	0.28	0.18	0.54	0.49	0.52	0.5	1	0.39	0.55	0.27	0.51	0.24	0.31
OpenBookQA	0.65	0.41	0.64	0.47	0.37	0.53	0.76	0.81	0.71	0.75	0.65	0.73	0.58	0.63	0.77	0.48	0.55	0.39	1	0.32	0.75	0.65	0.72	0.79
MRPC	0.41	0.38	0.3	0.27	0.009	0.4	0.27	0.37	0.4	0.36	0.41	0.27	0.18	0.48	0.42	0.39	0.38	0.55	0.32	1	0.26	0.41	0.23	0.29
HeadQA	0.47	0.29	0.62	0.33	0.49	0.39	0.76	0.76	0.62	0.66	0.5	0.82	0.65	0.47	0.64	0.35	0.44	0.27	0.75	0.26	1	0.45	0.83	0.83
RTE	0.78	0.6	0.54	0.63	0.11	0.62	0.49	0.56	0.65	0.62	0.66	0.46	0.37	0.71	0.62	0.61	0.62	0.51	0.65	0.41	0.45	1	0.45	0.53
ARC-Easy	0.46	0.29	0.59	0.33	0.51	0.4	0.76	0.73	0.65	0.69	0.49	0.79	0.76	0.47	0.65	0.32	0.44	0.24	0.72	0.23	0.83	0.45	1	0.86
ARC-Challenge	0.55	0.36	0.67	0.41	0.43	0.48	0.83	0.79	0.71	0.78	0.58	0.8	0.7	0.57	0.73	0.41	0.52	0.31	0.79	0.29	0.83	0.53	0.86	1

Figure 7: Cross benchmark ranking agreement under direct evaluation. Benchmarks are sorted based on the training dataset size. Kendall's τ is calculated for every benchmark pair.

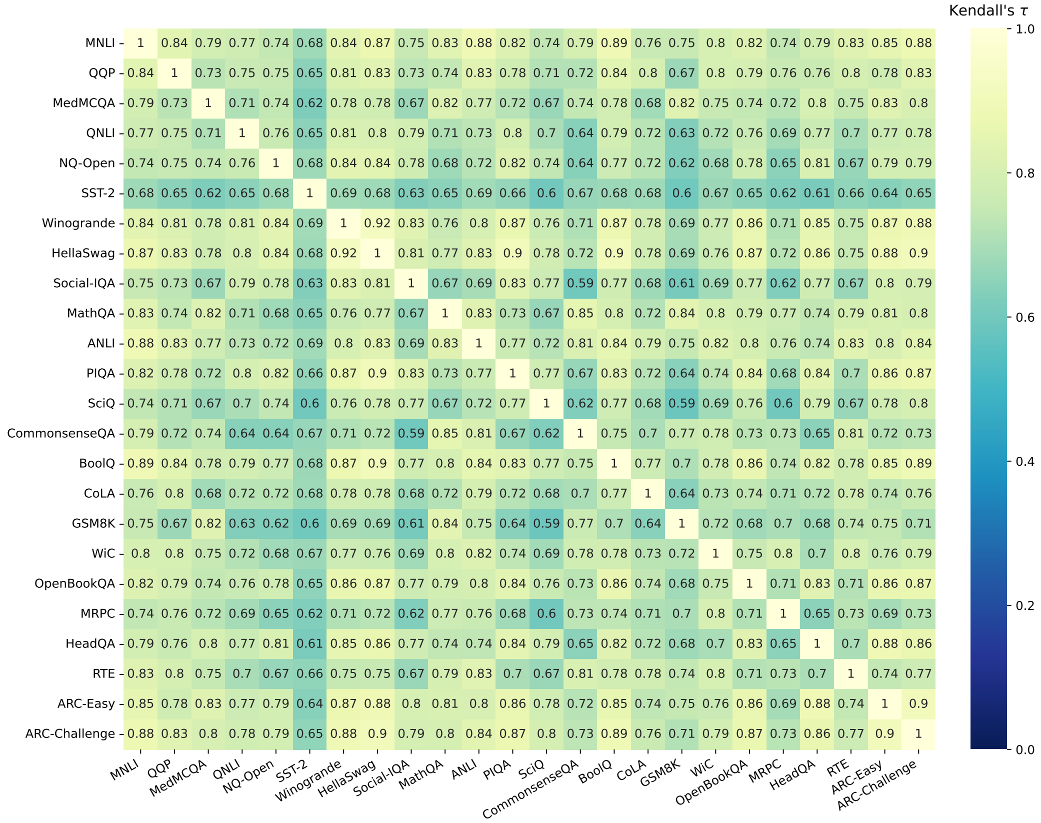


Figure 8: Cross benchmark ranking agreement under train-before-test. Benchmarks are sorted based on the training dataset size. Kendall's τ is calculated for every benchmark pair.

Table 3: Bits per byte (BPB) of eight excluded Gemma models compared to Pythia-410M across the three newly collected corpora. The Gemma models exhibit abnormally high BPB values on `Wiki` and `Stack`, likely due to the greater average sequence length in these two datasets. Specifically, `Arxiv` has an average of 163 words per document, compared to 250 for `Stack` and 1502 for `Wiki`.

	`Arxiv`	`Wiki`	`Stack`
Gemma-2B	0.766	1.578	1.139
Gemma-2B-IT	0.770	1.524	1.222
Gemma-7B	1.013	4.780	4.053
Gemma-7B-IT	1.053	18.711	20.958
Gemma-2-2B	0.730	1.784	1.340
Gemma-2-2B-IT	0.705	1.191	0.997
Gemma-2-9B	0.709	2.216	1.685
Gemma-2-9B-IT	0.638	1.234	0.978
Pythia-410M	0.791	1.065	0.945

C.2 Perplexity Ranking Agreement

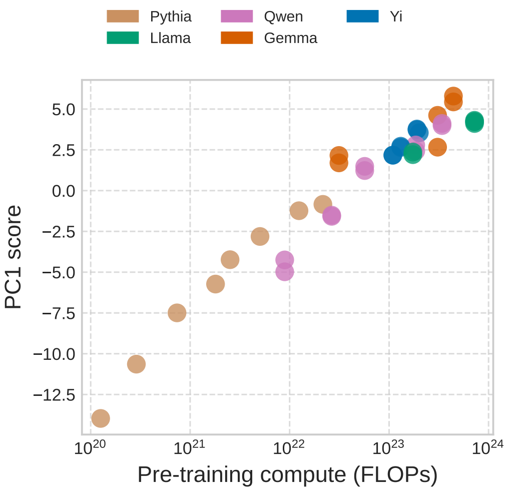
In this work, we collect three corpora from `Wiki`pedia, `Stack`Exchange, and `arXiv`. We only collect documents from 2025. More specifically, we collect 3,366 documents for `Wiki`, 6,001 for `Stack`Exchange and 44,384 documents for `arXiv`. These datasets are split into training, validation, and testing sets, in an 8:1:1 ratio. For `arXiv`, we utilize only the paper abstracts, while for `Stack`Exchange, we use only the questions. Consequently, the average document length is 163 words for `arXiv`, 250 words for `Stack`Exchange, and 1,502 words for `Wiki`pedia.

We exclude Gemma models from our perplexity agreement experiments, as `lm-eval-harness` provides unreliable perplexity measurements for Gemma models[1]. We report the bits per byte (BPB) for the Gemma models in Table 3. While the BPB values for Gemma on `arXiv` (the dataset with the shortest average sequence length) are mostly reasonable, the performance on `Stack`Exchange and `Wiki`pedia is notably worse, even compared to smaller models like Pythia-410M.

This anomaly stems from how `lm-eval-harness` handles long sequences via a rolling window mechanism. Unlike other models, Gemma requires every input sequence to begin with the `BOS` token. If this constraint is not met, perplexity degrades significantly. Consequently, when processing long sequences that are chunked into multiple windows, Gemma's performance degrades.

Additional results. Drawing inspiration from prior work [46, 83, 23, 20, 87], we further examine the correlation between model rankings according to *average* perplexity across the three text corpora and *average* downstream performance across the 24 benchmarks. Gadre et al. (2024, [23]) show that, when models are trained on the same pretraining data, perplexity is well-correlated with aggregate benchmark performance. Our setup differs in that we consider a diverse set of model families, each trained on different pretraining data. Under direct evaluation, we find that the ranking correlation is modest, with a Kendall's τ of only 0.55. We hypothesize that this relatively weak agreement is due to differences in pretraining data and instruction tuning, resulting in varying levels of exposure to benchmark tasks during training [19]. Fortunately, when applying our train-before-test methodology, the ranking correlation between average perplexity and average downstream performance improves substantially, with Kendall's τ increasing from 0.55 to 0.84.

[1] See discussion at `https://github.com/huggingface/transformers/issues/29250`.



t

Figure 9: PC1 scores under train-before-test align with the pre-training compute.

C.3 PC1 Score under Train-before-Test

We compare PC1 under train-before-test with pre-training compute in Figure 9. We only plot models whose number of training tokens is publicly available. See Table 4 for details. We further plot the PC1 scores under train-before-test in Figure 10.

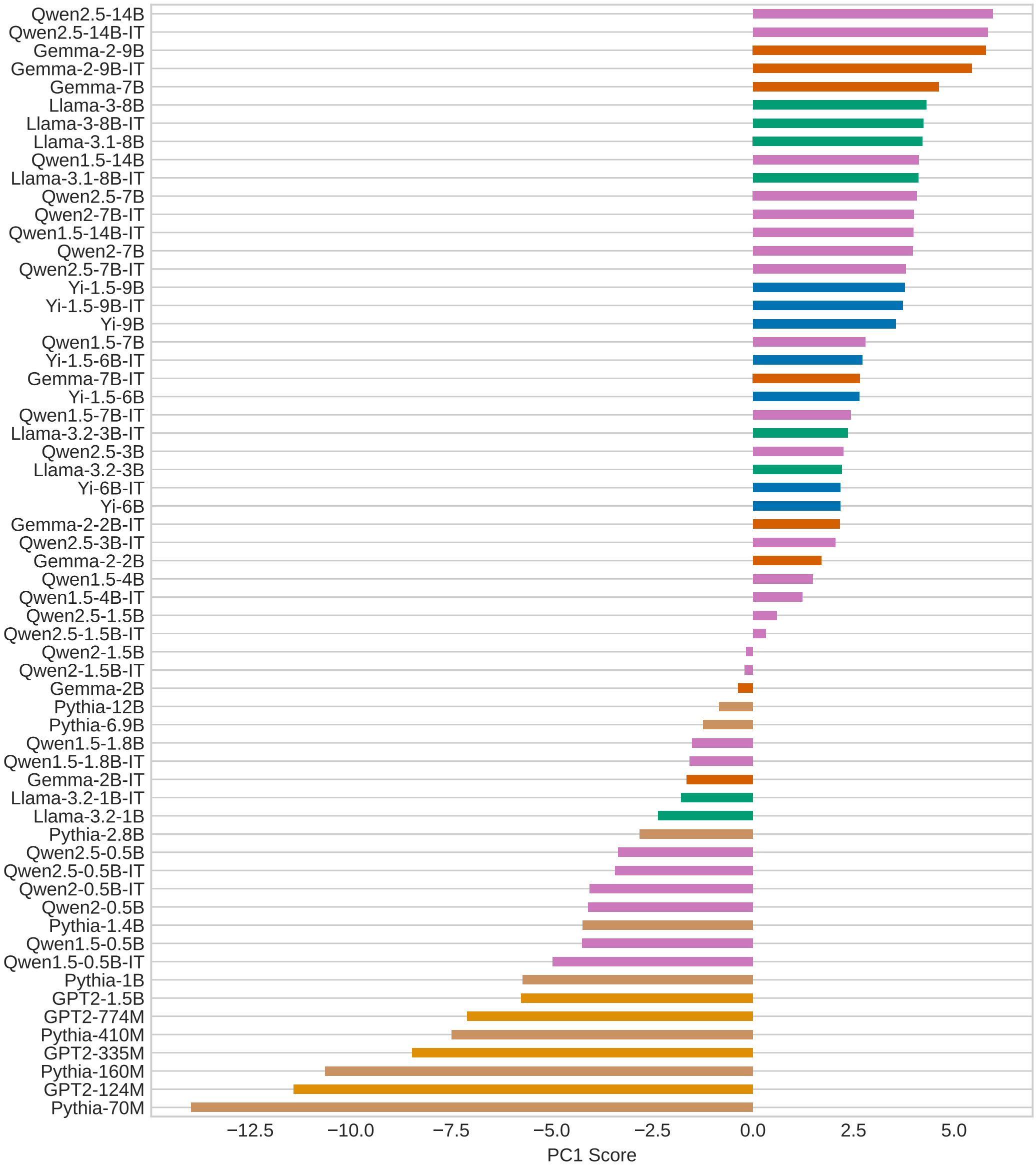


Figure 10: PC1 scores under train-before-test.

Table 4: The models used in Figure 9. The number of training tokens of these models is publicly available. We compute the number of pre-training FLOPs as 6 × #Parameters × #Tokens.

Model	#Parameters (B)	#Tokens (B)	#FLOPs (10^18)
Llama-3-8B	8.03	15000.0	722700.00
Llama-3-8B-IT	8.03	15000.0	722700.00
Llama-3.1-8B	8.03	15000.0	722700.00
Llama-3.1-8B-IT	8.03	15000.0	722700.00
Llama-3.2-3B	3.21	9000.0	173340.00
Llama-3.2-3B-IT	3.21	9000.0	173340.00
Qwen1.5-0.5B	0.62	2400.0	8928.00
Qwen1.5-1.8B	1.84	2400.0	26496.00
Qwen1.5-4B	3.95	2400.0	56880.00
Qwen1.5-7B	7.72	4000.0	185280.00
Qwen1.5-14B	14.20	4000.0	340800.00
Qwen1.5-0.5B-IT	0.62	2400.0	8928.00
Qwen1.5-1.8B-IT	1.84	2400.0	26496.00
Qwen1.5-4B-IT	3.95	2400.0	56880.00
Qwen1.5-7B-IT	7.72	4000.0	185280.00
Qwen1.5-14B-IT	14.20	4000.0	340800.00
Gemma-7B	8.54	6000.0	307440.00
Gemma-7B-IT	8.54	6000.0	307440.00
Gemma-2-2B	2.61	2000.0	31320.00
Gemma-2-2B-IT	2.61	2000.0	31320.00
Gemma-2-9B	9.24	8000.0	443520.00
Gemma-2-9B-IT	9.24	8000.0	443520.00
Pythia-70M	0.07	300.0	126.00
Pythia-160M	0.16	300.0	288.00
Pythia-410M	0.41	300.0	738.00
Pythia-1B	1.00	300.0	1800.00
Pythia-1.4B	1.40	300.0	2520.00
Pythia-2.8B	2.80	300.0	5040.00
Pythia-6.9B	6.90	300.0	12420.00
Pythia-12B	12.00	300.0	21600.00
Yi-6B	6.06	3000.0	109080.00
Yi-6B-IT	6.06	3000.0	109080.00
Yi-9B	8.83	3800.0	201324.00
Yi-1.5-6B	6.06	3600.0	130896.00
Yi-1.5-6B-IT	6.06	3600.0	130896.00
Yi-1.5-9B	8.83	3600.0	190728.00
Yi-1.5-9B-IT	8.83	3600.0	190728.00

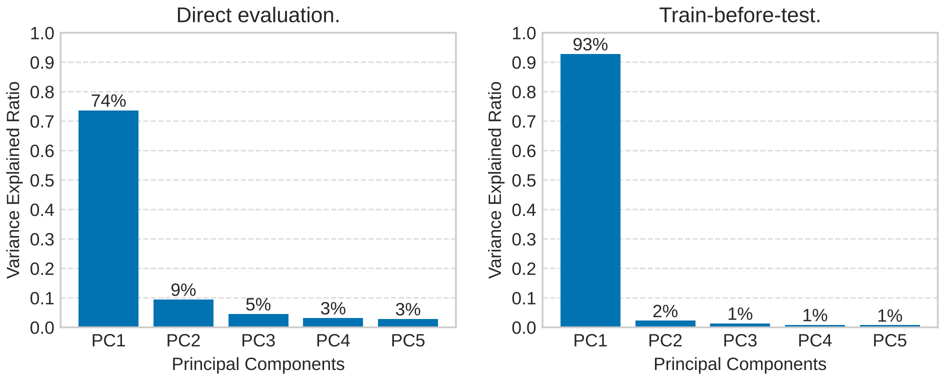


Figure 11: Explained variance ratios of the top five principal components of the QWEN score matrix. For train-before-test, the explained variance ratio of PC1 increases to 93%, making the QWEN score matrix essentially rank one.

C.4 Case Study for Qwen Models.

We repeat our PCA analysis on the score matrix containing only QWEN models, depicted in Figure 11. Remarkably, we find that PC1 for train-before-test explains 93% of the variance, roughly as much as the variance explained by the top five principal components under direct evaluation. That is, whereas for direct evaluation the score matrix is low-rank; train-before-test renders the score matrix essentially rank one.

D Accounting for Statistical Significance

D.1 Ranking Alignment in Figure 5

We plot the rankings of 61 language models on two question-answering benchmarks: Natural Questions Open and ARC Challenge in Figure 5. We greedily align each ranking as much as possible without violating confidence intervals, thus revealing only those ranking changes that are statistically significant. See Algorithm 3 for more details.

D.2 Downstream Ranking Agreement

We additionally supplement the experiments presented in the main text by modifying the ranking correlation metric to account for statistical significance in the benchmark evaluations. Specifically, we use Kendall's τ-b [39], which adjusts for ties in rankings. We consider two models tied on a given benchmark if their performance difference is not statistically significant at the 95% confidence level. We assess statistical significance using a t-test based on the standard error of the mean performances.

We reproduce the ranking correlation figures of the main text using the modified Kendall's τ which treats non-statistically significant performance differences as ties. See Figure 12 and 13; as well as Figure 14 and Figure 15 for more detailed results. We observe that accounting for statistical significance in models' performance differences leads to slightly higher ranking correlations, as measured by Kendall's τ-b. For direct evaluation, average agreement increases from 0.52 to 0.58. For train-before-test, average agreement increases from 0.76 to 0.77. Therefore, train-before-test continues to lead to large improvements in raking agreement (from Kendall's τ-b 0.58 to 0.77).

Algorithm 1: build_partial_order(scores, stderrs)

```
Input: Model performance scores and standard errors
Output: Directed graph $G$ representing significant model orderings
Initialize graph $G$ with models as nodes
foreach pair of distinct models $(m_1, m_2)$ do
    if $m_1$ is significantly better than $m_2$ then
        Add directed edge $(m_1 \to m_2)$ to $G$
return $G$
```

Algorithm 2: parallel_greedy_rank(models, G_1, G_2, score$_1$, score$_2$)

```
Input: List of models, two directed graphs $G_1$, $G_2$, and two score series
Output: Two lists representing the parallel ranking order for each task
Initialize vanillaRank$_1$, ← rankdata(score$_1$), vanillaRank$_2$ ← rankdata(score$_2$)
Initialize available$_1$ and available$_2$ as models with zero in-degree in $G_1$ and $G_2$
Initialize empty lists order$_1$, order$_2$
for $i = 1$ to number of models do
    Initialize empty list pairs
    foreach $m_1$ in available$_1$ do
        foreach $m_2$ in available$_2$ do
            Compute cost for pair $(m_1, m_2)$ based on:
                (1) Placement of $m_1$ in order$_2$ and $m_2$ in order$_1$
                (2) Whether $m_1 = m_2$ (prefer matching)
                (3) Combined vanilla ranks: vanillaRank$_2[m_1]$ + vanillaRank$_1[m_2]$
            Append $(cost, m_1, m_2)$ to pairs
    Sort pairs by cost (ascending)
    Select $(m_1, m_2)$ with minimal cost
    Append $m_1$ to order$_1$, $m_2$ to order$_2$
    Remove $m_1$ from $G_1$ and update available$_1$
    Remove $m_2$ from $G_2$ and update available$_2$
return order$_1$, order$_2$
```

Algorithm 3: rank_models(score$_1$, stderr$_1$, score$_2$, stderr$_2$)

```
Input: Scores and standard errors for two tasks
Output: Parallel rankings for both tasks
    $G_1$ ← build_partial_order(score$_1$, stderr$_1$)
    $G_2$ ← build_partial_order(score$_2$, stderr$_2$)
    $(order_1, order_2)$ ← parallel_greedy_rank(models, $G_1$, $G_2$, score$_1$, score$_2$)
    rank$_1[m]$ = position of $m$ in $order_1$ (starting from 1)
    rank$_2[m]$ = position of $m$ in $order_2$ (starting from 1)
return rank$_1$, rank$_2$
```

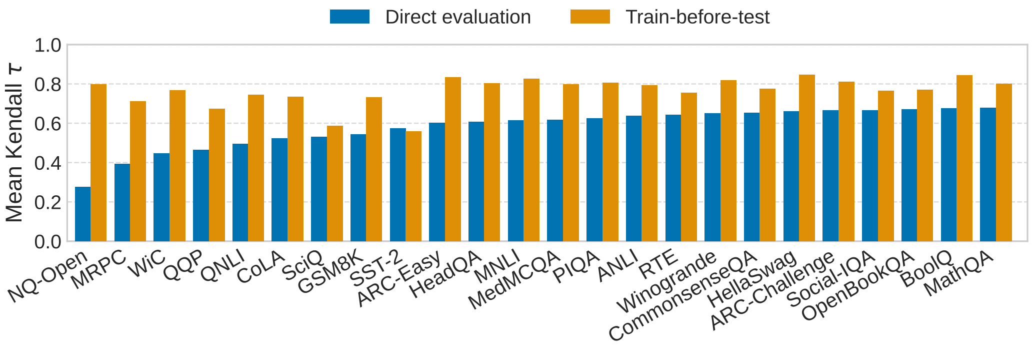
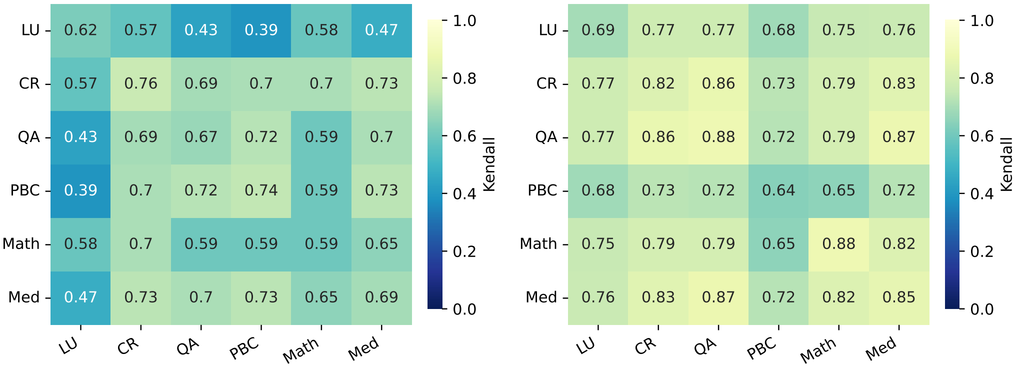


Figure 12: Mean ranking agreement between each benchmark and all others, measured by Kendall's tau-b, *with non-statistically significant performance differences being treated as ties*. We calculate Kendall's τ-b between each benchmark and every other one, and then average. Compared to direct evaluation, train-before-test consistently improves ranking agreement–often by a large margin. On average, the overall average Kendall's τ is 0.58 for direct evaluation and 0.77 for train-before-test.



(a) Direct evaluation.

(b) Train-before-test.

Figure 13: Cross-category ranking agreement for direct evaluation (left) and train-before-test (right), measured by Kendall's tau-b, *with non-statistically significant performance differences being treated as ties*. We consider language understanding (LU), commonsense reasoning (CR), question answering (QA), physics/biology/chemistry (PBC), math (Math), and medicine (Med) categories. Kendall's τ-b is averaged across all pairs of benchmarks that belong to two given categories. The diagonal represents the intra-category agreement and the others represent the inter-category agreement. train-before-test improves both intra- and inter-category ranking agreement in all instances.

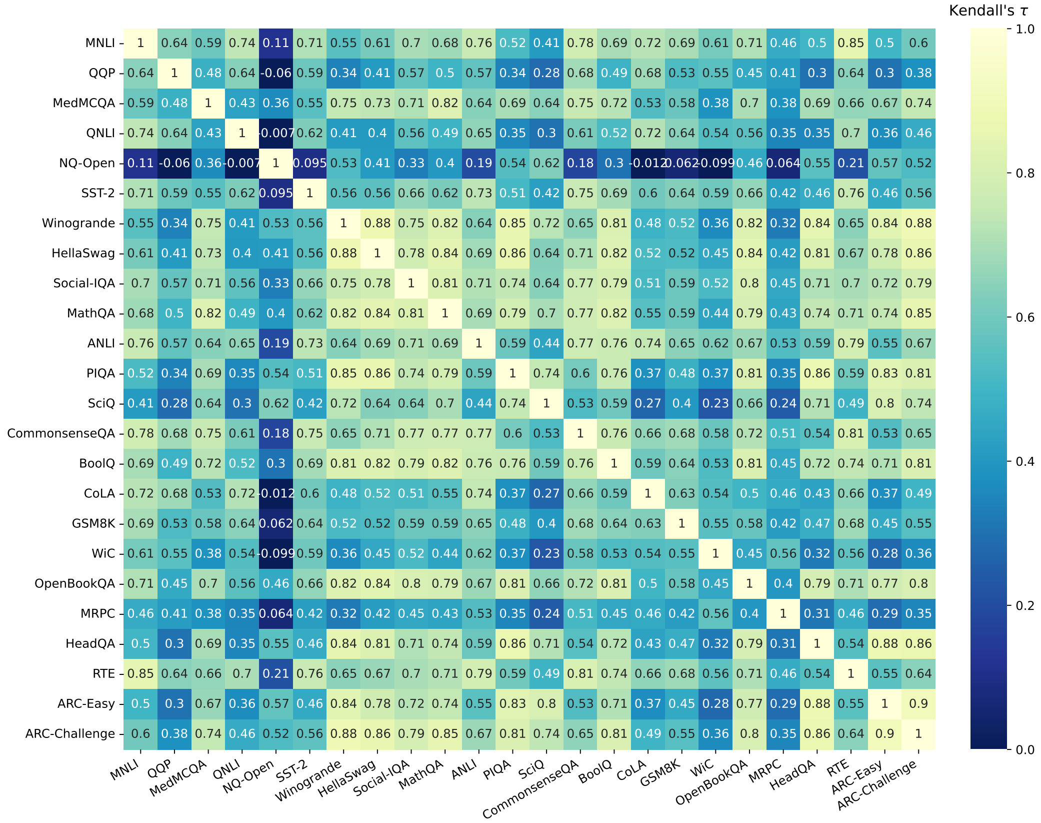
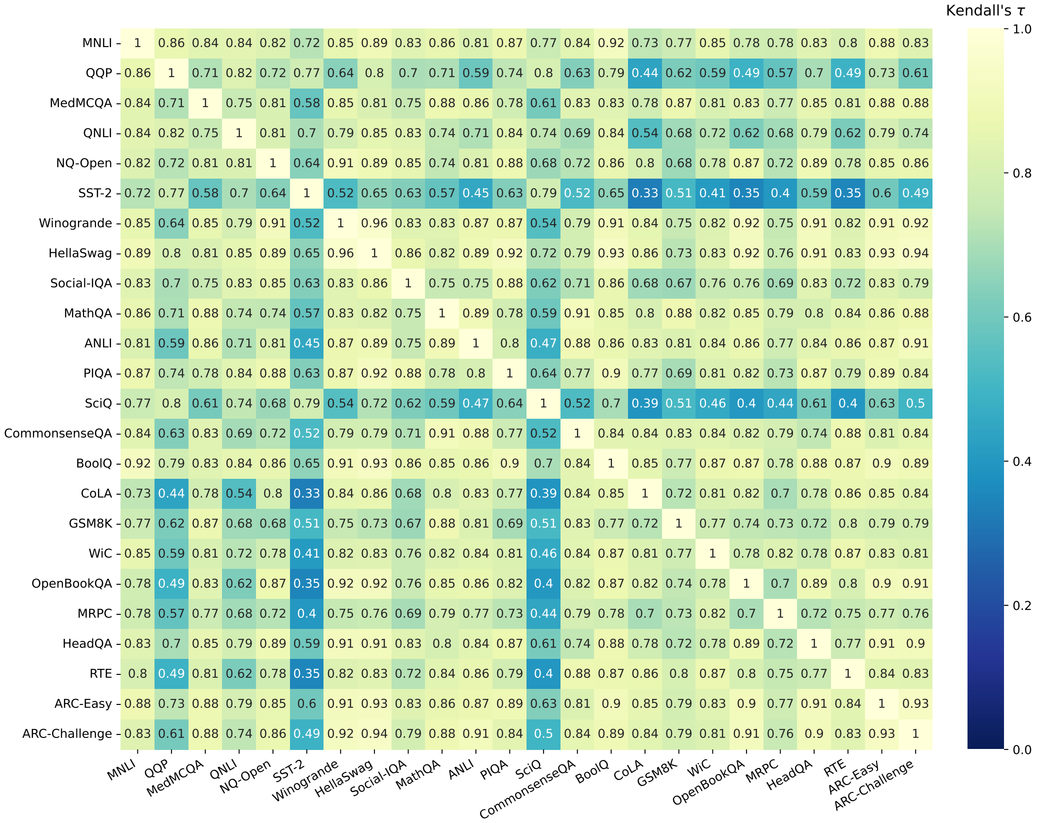


Figure 14: Cross benchmark ranking agreement under direct evaluation, measured by Kendall's tau-b with insignificant model comparisons treated as ties.



	MNLI	QQP	MedMCQA	QNLI	NQ-Open	SST-2	Winogrande	HellaSwag	Social-IQA	MathQA	ANLI	PIQA	SciQ	CommonsenseQA	BoolQ	CoLA	GSM8K	WiC	OpenBookQA	MRPC	HeadQA	RTE	ARC-Easy	ARC-Challenge
MNLI	1	0.86	0.84	0.84	0.82	0.72	0.85	0.89	0.83	0.86	0.81	0.87	0.77	0.84	0.92	0.73	0.77	0.85	0.78	0.78	0.83	0.8	0.88	0.83
QQP	0.86	1	0.71	0.82	0.72	0.77	0.64	0.8	0.7	0.71	0.59	0.74	0.8	0.63	0.79	0.44	0.62	0.59	0.49	0.57	0.7	0.49	0.73	0.61
MedMCQA	0.84	0.71	1	0.75	0.81	0.58	0.85	0.81	0.75	0.88	0.86	0.78	0.61	0.83	0.83	0.78	0.87	0.81	0.83	0.77	0.85	0.81	0.88	0.88
QNLI	0.84	0.82	0.75	1	0.81	0.7	0.79	0.85	0.83	0.74	0.71	0.84	0.74	0.69	0.84	0.54	0.68	0.72	0.62	0.68	0.79	0.62	0.79	0.74
NQ-Open	0.82	0.72	0.81	0.81	1	0.64	0.91	0.89	0.85	0.74	0.81	0.88	0.68	0.72	0.86	0.8	0.68	0.78	0.87	0.72	0.89	0.78	0.85	0.86
SST-2	0.72	0.77	0.58	0.7	0.64	1	0.52	0.65	0.63	0.57	0.45	0.63	0.79	0.52	0.65	0.33	0.51	0.41	0.35	0.4	0.59	0.35	0.6	0.49
Winogrande	0.85	0.64	0.85	0.79	0.91	0.52	1	0.96	0.83	0.83	0.87	0.87	0.54	0.79	0.91	0.84	0.75	0.82	0.92	0.75	0.91	0.82	0.91	0.92
HellaSwag	0.89	0.8	0.81	0.85	0.89	0.65	0.96	1	0.86	0.82	0.89	0.92	0.72	0.79	0.93	0.86	0.73	0.83	0.92	0.76	0.91	0.83	0.93	0.94
Social-IQA	0.83	0.7	0.75	0.83	0.85	0.63	0.83	0.86	1	0.75	0.75	0.88	0.62	0.71	0.86	0.68	0.67	0.76	0.76	0.69	0.83	0.72	0.83	0.79
MathQA	0.86	0.71	0.88	0.74	0.74	0.57	0.83	0.82	0.75	1	0.89	0.78	0.59	0.91	0.85	0.8	0.88	0.82	0.85	0.79	0.8	0.84	0.86	0.88
ANLI	0.81	0.59	0.86	0.71	0.81	0.45	0.87	0.89	0.75	0.89	1	0.8	0.47	0.88	0.86	0.83	0.81	0.84	0.86	0.77	0.84	0.86	0.87	0.91
PIQA	0.87	0.74	0.78	0.84	0.88	0.63	0.87	0.92	0.88	0.78	0.8	1	0.64	0.77	0.9	0.77	0.69	0.81	0.82	0.73	0.87	0.79	0.89	0.84
SciQ	0.77	0.8	0.61	0.74	0.68	0.79	0.54	0.72	0.62	0.59	0.47	0.64	1	0.52	0.7	0.39	0.51	0.46	0.4	0.44	0.61	0.4	0.63	0.5
CommonsenseQA	0.84	0.63	0.83	0.69	0.72	0.52	0.79	0.79	0.71	0.91	0.88	0.77	0.52	1	0.84	0.84	0.83	0.84	0.82	0.79	0.74	0.88	0.81	0.84
BoolQ	0.92	0.79	0.83	0.84	0.86	0.65	0.91	0.93	0.86	0.85	0.86	0.9	0.7	0.84	1	0.85	0.77	0.87	0.87	0.78	0.88	0.87	0.9	0.89
CoLA	0.73	0.44	0.78	0.54	0.8	0.33	0.84	0.86	0.68	0.8	0.83	0.77	0.39	0.84	0.85	1	0.72	0.81	0.82	0.7	0.78	0.86	0.85	0.84
GSM8K	0.77	0.62	0.87	0.68	0.68	0.51	0.75	0.73	0.67	0.88	0.81	0.69	0.51	0.83	0.77	0.72	1	0.77	0.74	0.73	0.72	0.8	0.79	0.79
WiC	0.85	0.59	0.81	0.72	0.78	0.41	0.82	0.83	0.76	0.82	0.84	0.81	0.46	0.84	0.87	0.81	0.77	1	0.78	0.82	0.78	0.87	0.83	0.81
OpenBookQA	0.78	0.49	0.83	0.62	0.87	0.35	0.92	0.92	0.76	0.85	0.86	0.82	0.4	0.82	0.87	0.82	0.74	0.78	1	0.7	0.89	0.8	0.9	0.91
MRPC	0.78	0.57	0.77	0.68	0.72	0.4	0.75	0.76	0.69	0.79	0.77	0.73	0.44	0.79	0.78	0.7	0.73	0.82	0.7	1	0.72	0.75	0.77	0.76
HeadQA	0.83	0.7	0.85	0.79	0.89	0.59	0.91	0.91	0.83	0.8	0.84	0.87	0.61	0.74	0.88	0.78	0.72	0.78	0.89	0.72	1	0.77	0.91	0.9
RTE	0.8	0.49	0.81	0.62	0.78	0.35	0.82	0.83	0.72	0.84	0.86	0.79	0.4	0.88	0.87	0.86	0.8	0.87	0.8	0.75	0.77	1	0.84	0.83
ARC-Easy	0.88	0.73	0.88	0.79	0.85	0.6	0.91	0.93	0.83	0.86	0.87	0.89	0.63	0.81	0.9	0.85	0.79	0.83	0.9	0.77	0.91	0.84	1	0.93
ARC-Challenge	0.83	0.61	0.88	0.74	0.86	0.49	0.92	0.94	0.79	0.88	0.91	0.84	0.5	0.84	0.89	0.84	0.79	0.81	0.91	0.76	0.9	0.83	0.93	1

Figure 15: Cross benchmark ranking agreement under train-before-test, measured by Kendall's tau-b with insignificant model comparisons treated as ties.

E Discussion, limitations, and conclusion

Train-before-test fundamentally reframes how we interpret model evaluation. Whereas direct evaluation yields benchmark-specific rankings that often contradict one another, train-before-test harmonizes rankings across a wide array of tasks and datasets. This shift from measuring *performance* (out-of-the-box ability) to measuring *potential* (achievable ability after task-specific fine-tuning) equips the community with a more stable and externally valid evaluation methodology.

From a practical standpoint, this external validity is critical. Practitioners rarely deploy models without adaptation; instead, they fine-tune models on their own data and objectives. Direct evaluation, while useful for assessing deployment readiness, is of limited relevance in such cases. Potential evaluation, on the other hand, provides more actionable guidance for model selection by revealing which models are best positioned to excel after adaptation. Our empirical results show that this methodology consistently aligns rankings across benchmarks, restores coherence between perplexity and downstream performance, and distills benchmark outcomes into a single dominant latent factor.

One might argue that ranking consistency is unnecessary if we can simply choose benchmarks close to a given downstream application. However, our findings highlight three challenges with that view. First, even benchmarks that purport to measure the same skill (e.g., question answering) produce contradictory rankings under direct evaluation. Second, no benchmark perfectly captures the specifics of an application, making some degree of generalization unavoidable. Third, in realistic deployment scenarios, models are almost always fine-tuned, making their *potential* the relevant signal for comparison. Together, these points underscore why consistency and external validity are essential features of any evaluation methodology.

Limitations. Train-before-test requires that we fine-tune models on agreed upon task-specific data prior to evaluation. This certainly increases the cost of evaluation and might be too costly in some cases. However, this investment yields dividends through improved reliability. Our findings suggest that fewer benchmarks suffice under train-before-test, as rankings from one benchmark reliably transfer to others. This reduction in required evaluations can offset the per-benchmark cost increase. A second problem is that, unfortunately, many benchmarks no longer come with training data, making it more difficult to apply train-before-test. In light of our findings, we recommend that future benchmarks provide fine-tuning data for the benchmark. A third limitation is that some commercial model providers do not easily allow fine-tuning of their models. We contend that in this case the problem is with the model provider. There is clearly scientific value in creating an ecosystem of models that can be fine-tuned. Train-before-test evaluation creates additional incentives for making models easy to fine-tune.

Conclusion. Overall, train-before-test complements existing evaluation practices by distinguishing between *performance* and *potential*. Direct evaluation remains useful for gauging immediate deployment readiness, while train-before-test offers deeper insight into long-term adaptability and development prospects. Together, they provide a more complete picture of language model capabilities. We believe that adopting train-before-test as a standard alongside direct evaluation can significantly improve the reliability, interpretability, and practical utility of the model evaluation ecosystem.

F Broader Impacts and Limitations

Due to resource constraints, we do not conduct exhaustive hyperparameter searches for each model and benchmark. Instead, we use the same set of hyperparameters for all models and benchmarks. We do not tune these hyperparameters for any given model or benchmark. Therefore, all models received the same amount of hyperparameter search budget, albeit small.

We restrict our analysis to benchmarks that have a train set. However, recent benchmarks seldom include a train split. Thus, most of the benchmarks considered in our analysis, while typically highly cited and influential, were released before 2022.

We do not anticipate any direct societal impacts from this work, such as potential malicious or unintended uses, nor do we foresee any significant concerns involving fairness, privacy, or security considerations. Additionally, we have not identified potential harms resulting from the application of this technology.