Domain Transfer: Weak and Strong

Nathan Ju, Yuezhou Hu

Abstract

A central question in data mixture design for LLMs is: for which domains (e.g., math, science, code) does training primarily on that domain significantly improve performance in another? We study this question through the lens of attribution, measuring how much a training example influences performance on a downstream task. Inspired by gradient-based attribution methods, we show that gradient overlaps between domains (e.g., math–science) reliably predict transferability when compared against ground-truth benchmark data. Using this perspective, we find that: mathematics transfers strongly to science, science transfers weakly to math, and neither math nor science transfers meaningfully to code.

We then examine the *limits* of domain transfer. When training on 100% mathematics examples identified via gradient overlap, we achieve performance on curriculum-style science benchmarks (e.g. physics subset of MMLU) comparable to mixtures dominated by science data. However, this substitution fails on expert-authored science benchmarks requiring narrow, specialized knowledge (GPQA-diamond). Taken together, our results shed light on when cross-domain transfer is feasible and indicate that a gradient-based measure predicts domain transfer during fine-tuning.

1 Introduction

Recent progress in large language models (LLMs) has been driven not only by advances in model architecture and optimization but also by increasingly deliberate choices of training data. Modern LLMs now excel in domains such as mathematics, software engineering, and scientific reasoning (Yang et al., 2025; OpenAI et al., 2025). These developments have in turn motivated a growing research effort around data mixture design (Xie et al., 2023; Chen et al., 2025; Guha et al., 2025). These efforts range from principled approaches (e.g. optimization-based) to more heuristic approaches (by measuring performance on smaller models). However, much of this work focuses on *constructing* performant data mixtures, rather than *understanding* the instrinsic transferrability of one domain of data to another. For example, if a model is trained predominantly—or even entirely—on data from a single domain (e.g. math), how far can this push performance on a distinct target domain (e.g. physics)?

Understanding this form of *strong* domain transfer is crucial for characterizing the extremes of data mixture construction and for clarifying when cross-domain generalization is, or is not, possible. This perspective prompts three concrete questions: Which domains transfer to which? Can training-time signals predict transferability? Can these signals be leveraged to intentionally push strong domain transfer?

We address these questions using simple gradient-based measurements, inspired by attribution methods (Pruthi et al., 2020; Xia et al., 2024). Specifically, we compute the gradient overlap between the gradients induced by a source domain (e.g. math) and a target domain (e.g. science) and examine how they relate to downstream transferability. Our analysis yields two key findings.

First, we show that gradient overlap provides a reliable, training-time signal of domain transfer. Domains with large overlap to a target domain (e.g. math to science) tend to yield higher downstream performance when used as source domains. Similarly, domains with weak overlap to a target domain (e.g. science to code) yield weaker downstream

performance when used as a source domain. Notably, we find that mathematics emerges as a privileged domain: its gradients exhibit exceptionally high overlap with those of science tasks, which is not characteristic of science and code data.

Second, we use these gradient signals to explore the limits of strong domain transfer. Our experiments show that, for certain science benchmarks—such as the Physics subsets of MMLU, which are drawn from standard high-school and college curricula—models trained on 100% mathematics data can perform competitively with models trained on in-domain science. However, this behavior does not generalize to more specialized or expert-level evaluations: on GPQA-diamond, whose graduate-level questions require narrow scientific knowledge, the same substitution fails. This contrast suggests that the feasibility of strong domain transfer depends sensitively on the nature of the target benchmark—its difficulty, breadth, and reliance on domain-specific content—and motivates a deeper examination of when one domain can meaningfully stand in for another.

1.1 Prior Work

Cross-Domain Transfer in LLMs. A growing body of empirical work demonstrates that performance gains in one reasoning domain can—under the right conditions—positively transfer to others. Huan et al. (2025) show that models trained *only on math* via reinforcement learning (RL) can improve not only on math but also on science, coding, and even non-reasoning tasks, whereas supervised fine-tuning (SFT) on the same math-only data often harms out-of-domain performance. While both settings use identical training data, the choice of optimization method (RL vs. SFT) determines whether *effective cross-domain transfer*—i.e., competitive or even superior performance on a target domain after training solely on a different source domain—is observed.

Independently, Guha et al. (2025) observe similar cross-domain transfer in a purely SFT setting. Through over 1,000 controlled ablations, they find that models trained exclusively on high-quality *math* data (e.g., from `OpenMath-2-Math` (Toshniwal et al., 2024)) achieve the strongest gains on science (GPQA (Rein et al., 2023), JEEBench (Arora et al., 2023)) and code (LiveCodeBench (Jain et al., 2024)) despite zero in-domain examples. Notably, mixing this math data with strong code data *fails to further improve* science performance beyond math alone (see their Table 24), suggesting math carries a uniquely effective inductive bias for reasoning.

Together, these results confirm that *effective cross-domain transfer*—i.e., nontrivial performance gains on a target domain after training on a different source domain—is empirically achievable. However, two key questions remain open: (1) To what extent can such transfer approach the performance of *in-domain training*? (2) How can we reliably *identify* or *predict* when such cross-domain transfer will occur?

2 Predicting domain transfer

2.1 Gradient overlaps predict domain transfer

We begin by noting that transferability cannot be assessed solely by examining a model trained entirely on a single domain. For example, a model fine-tuned only on mathematics may perform poorly on coding tasks, yet a small amount of coding supervision can dramatically improve its coding performance. Thus, to understand cross-domain transfer, we require a measure that captures how individual training examples from one domain influence performance on another. Motivated by techniques from the data attribution literature (Pruthi et al., 2020; Xia et al., 2024) we employ a gradient-based influence score to quantify these interactions.

Constructing the gradient-based heatmap: In our experiments, we fine-tune a Llama-3.2-1B-Instruct model using LoRA on the OpenThoughts (Guha et al., 2025) reasoning dataset, which consists of three domains: mathematics, science, and coding. For a model with

parameters θ, we denote by

$$f_{\text{math}}(\theta) \quad f_{\text{science}}(\theta) \quad f_{\text{code}}(\theta)$$

the average loss on the corresponding domain subsets of OpenThoughts.

For any datapoint i, let $\tilde{\theta}_{i,1}, \ldots, \tilde{\theta}_{i,\tilde{\xi}}$ denote the per-epoch parameter updates attributable to that example during fine-tuning (e.g., the contributions to stochastic gradient or Adam updates). Let θ_{final} be the model parameters at the end of training. The influence of datapoint i on a domain loss $f(\theta)$ is defined as the cumulative alignment between the parameter updates from i and the final gradient of that domain:

$$\text{Influence}_i(f) = \sum_{j=1}^{\tilde{\xi}} \cos(\tilde{\theta}_{i,j}, \nabla_\theta f(\theta_{\text{final}})) \tag{1}$$

For each target domain $f \in \{f_{\text{math}}, f_{\text{science}}, f_{\text{code}}\}$, we examine the top 20% most influential datapoints given by scores, Equation (1). For each target domain, we count the proportion of this top 20% from each source domain to construct the heatmap in Figure 1a.

Although prior work (Park et al., 2023) shows that such gradient-overlap–based estimates correlate weakly with counterfactual influence (often exhibiting Pearson correlations below 0.2), we find that they nevertheless capture domain-level transfer effects remarkably well.

Constructing the OpenThoughts-based heatmap: To evaluate this, we compare the gradient-based heatmap in Figure 1a to a second heatmap constructed in a completely different manner using results reported in OpenThoughts. Table 3 of (Guha et al., 2025) provides the downstream benchmark performance of a Qwen-2.5-7B-Instruct model fine-tuned exclusively on each domain of training data. For each target domain (math, science, and code), we obtain a triplet of scores

$$(\alpha_{\text{math}}, \alpha_{\text{science}}, \alpha_{\text{code}})$$

corresponding to training solely on math, solely on science, and solely on code data.[1]

To obtain a comparable measure of relative domain effectiveness, we normalize each triplet by mapping its lowest value to 0% and scaling the remaining two values linearly according to their improvement above this minimum, so that all three values lie between 0 and 100. These normalized advantage scores yield the heatmap shown in Figure 1b.

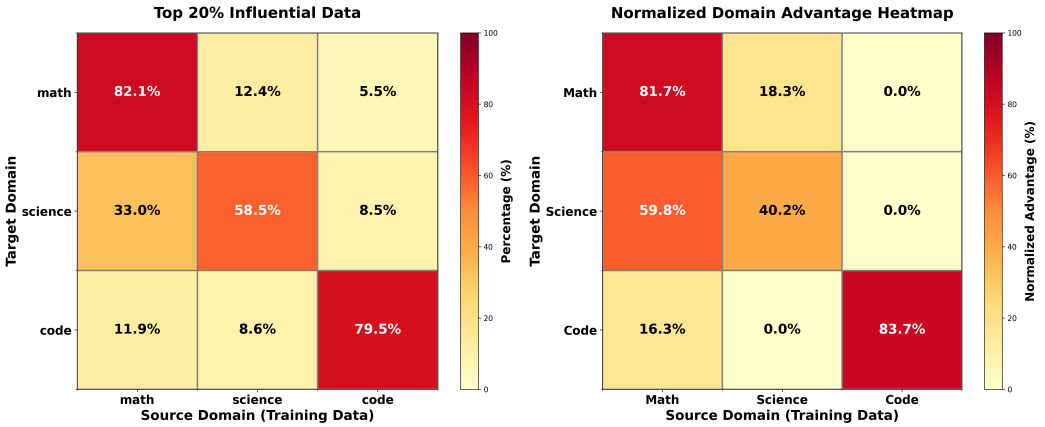
Both heatmaps share qualitative structure: Despite being derived from entirely different signals (in-training gradient overlaps vs. downstream benchmark performance), the two heatmaps show the same qualitative structure: (1) Strong transfer from math to science. Science benefits heavily from training on math examples, consistent with the OpenThoughts data. (2) Moderate transfer from science to math. Both methods indicate a symmetric but weaker transfer in the opposite direction. (3) Minimal transfer into code. Neither math nor science helps meaningfully on code tasks; code is largely self-contained.

This qualitative agreement suggests that despite known limitations, gradient overlaps accurately reflect qualitative relationships between domains that correlate with real transfer behavior.

2.2 Properties of transferability predicted by gradient overlaps

In this section, we quantitatively analyze the contributions of different data sources to specific downstream tasks. Building upon the framework introduced earlier, we examine the composition of high-influence examples—defined as the top-X% samples ranked by gradient overlap with a target domain—across five distinct data sources spanning the three domains: science, math, and code. The sources include:

[1] OpenThoughts reports results per data source; we average over sources belonging to the same domain.



(a) Gradient-based heatmap (b) OpenThought-derived heatmap

Figure 1: Heatmaps for domain-to-domain transferability. Transfer scores given by gradient-based influence scores share qualitative similarities to ground-truth OpenThoughts benchmark data.

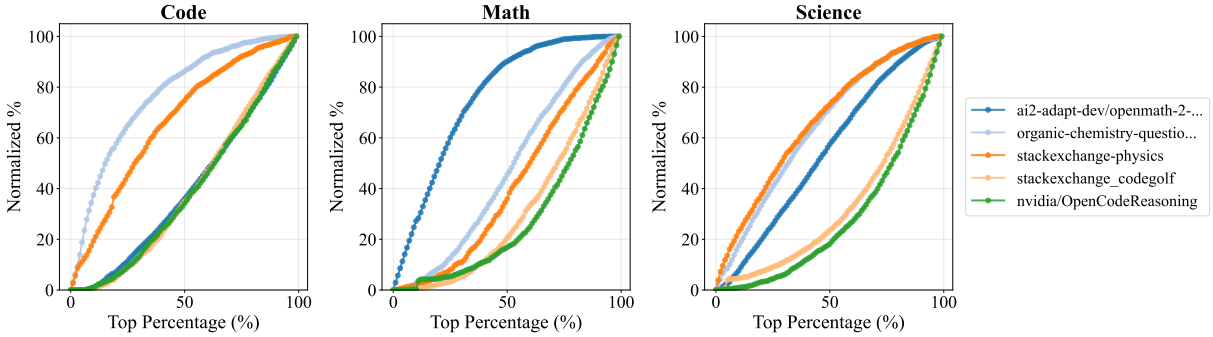


Figure 2: Normalized composition of high-influence examples by data source across different target tasks. The x-axis denotes the percentile of top-ranked examples selected, while the y-axis shows the normalized percentage of examples contributed by each source. Each panel (left to right) represents a different target task: science, math, and code.

- `ai2-adapt-dev/openmath-2-math` (math)
- `organic-chemistry-questions` (chemistry/science)
- `stackexchange-physics` (physics/science)
- `stackexchange_codegolf` (programming)
- `nvidia/OpenCodeReasoning` (programming)

Figure 2 visualizes the cumulative proportion of each source within the high-influence subset as a function of the selection percentile, for science, math, and code as target domains.

Key findings from these plots are summarized below:

1. For science tasks, high-influence data is concentrated in chemistry and physics, with math also playing a significant role. As shown in Figure 2(a), when science is the target, the top-ranked examples are overwhelmingly drawn from `organic-chemistry-questions` and `stackexchange-physics`. These two sources exhibit similar influence profiles, indicating comparable contributions to science performance. Following closely behind is the math dataset (`ai2-adapt-dev/openmath-2-math`), whose curve rises steeply, confirming that mathematical reasoning also plays a dominant role among the most influential examples. This

reinforces the strong "math → science" transfer and highlights that both scientific thinking (physics/chemistry) and abstract reasoning (math) serve as core drivers.

In contrast, programming sources (`stackexchange_codegolf`, `nvidia/OpenCodeReasoning`) contribute negligibly until the selection percentile exceeds 10%, suggesting their impact on science performance is marginal and supplementary at best.

2. For math tasks, domain-specific data dominates early, but science data contributes meaningfully—and surprisingly, so does code. Figure 2(b) reveals that for math as the target, the math dataset (`ai2-adapt-dev/openmath-2-math`) almost exclusively occupies the top ranks under small budgets, reflecting strong intra-domain influence. However, as the budget expands, science-related sources—particularly physics (`stackexchange-physics`) and chemistry (`organic-chemistry-questions`)—grow rapidly and contribute substantially, confirming the moderate "science → math" transfer driven primarily by scientific reasoning.

3. For code tasks, high-influence data is highly self-contained, with minimal external contribution. As shown in Figure 2(c), when targeting code, the vast majority of high-influence examples come from the two programming datasets: `nvidia/OpenCodeReasoning` and `stackexchange_codegolf`. Both curves rise sharply and remain near the top, confirming they are the primary drivers of performance.

All non-code sources—math, physics, and chemistry—remain at the bottom with slow growth, demonstrating their limited influence on code performance. This pattern mirrors the near-zero off-diagonal gradient overlap observed in the code row of Figure 1(a), reinforcing that code exhibits minimal cross-domain transfer under gradient-based measures.

3 Optimizing domain transfer

In this section, we study strong domain transfer: the ability of a model fine-tuned primarily on one domain to perform well on a different target domain. Guided by Section 2, where we observed the strongest transfer signal from mathematics to science, we focus specifically on this pair. To maximize the performance of the trained model, we use the gradient-overlap scores from Section 3 to construct optimized math/science mixtures for downstream science tasks.

Our experimental setup is as follows. We fix a compute budget, and we vary the mixture proportion p between math and science data given this compute budget. Among the p% allocated to math, we select the top-p% math datapoints ranked by influence toward science; similarly, among the $(1-p)$% allocated to science, we select the top-$(1-p)$% science datapoints. We then LoRA fine-tune Llama-3.2-1B-Instruct on this selected subset, and evaluate the resulting model on two science benchmarks:

1. **Physics MMLU** (high_school_physics and college_physics subsets), representing high-school and college-level scientific reasoning.
2. **GPQA-diamond** (40-question random subset), representing graduate-level scientific questions requiring conceptual and domain-specific knowledge.

MMLU (see Figure 3) On Physics MMLU, the best mixture under this influence-guided strategy occurs at 80% science and 20% math, consistent with the strong math-to-science transfer observed earlier. Surprisingly, however, we also observe a clear instance of strong domain transfer: training entirely on math achieves performance comparable to the optimal mixed strategy. Given that the MMLU physics questions are drawn from standard high-school and college practice materials and examinations (Hendrycks et al., 2021), this suggests that, for this style and difficulty of physics assessment, targeted math pretraining can substitute for a significant portion of in-domain physics supervision.

GPQA (see Figure 4) In contrast, for GPQA-diamond, the optimal mixture remains 80% science and 20% math, but pure-math training performs substantially worse. Here, mathematics provides only weak domain transfer. We believe this to be consistent with the construction of GPQA-diamond: the questions are graduate-level, narrow in subdomain,

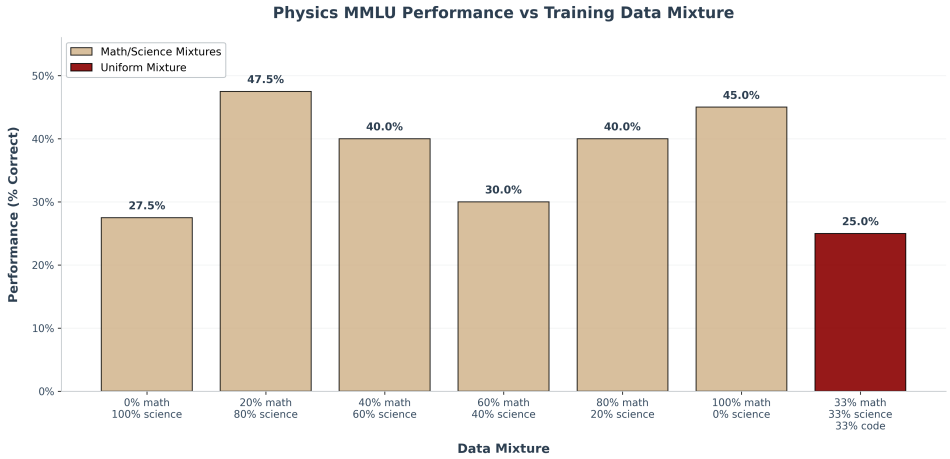


Figure 3: Physics MMLU performance as a function of math/science mixture proportions.

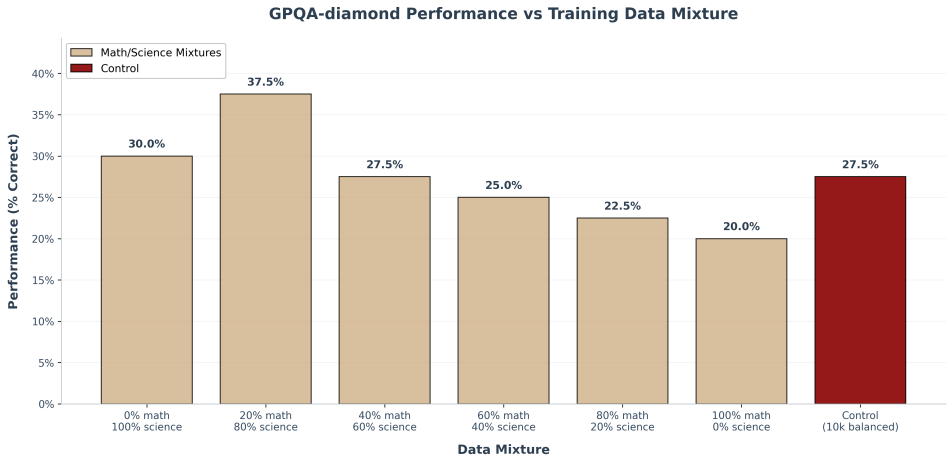


Figure 4: GPQA-diamond performance as a function of math/science mixture proportions.

and explicitly designed to be difficult even for non-expert PhDs with access to web search, while remaining objective for domain experts (Rein et al., 2023). In this regime, in-domain scientific supervision appears considerably more important than in the MMLU physics setting.

Taken together, these experiments indicate that strong math-to-science transfer is more pronounced on exam-style physics benchmarks drawn from standard curricula (as in Physics MMLU) than on expert-authored, graduate-level science benchmarks (as in GPQA-diamond). One plausible explanation, consistent with the dataset constructions, is that MMLU physics emphasizes broadly taught concepts and problem types that overlap heavily with mathematical training, whereas GPQA-diamond targets specialized scientific knowledge and reasoning that is less easily substituted by math alone. More generally, our results suggest that observed strong domain transfer depends not only on the source and target domains, but also on how the target benchmark is constructed (grade level, breadth, and degree of expert curation).

This refines our understanding from Section 2: while gradient overlaps capture relative transferability across domains, the strength of domain substitution is inherently task-dependent.

4 Conclusions and future directions

Our findings show that cross-domain transfer is measurable and often asymmetric: mathematics provides substantial benefit to science tasks, science provides less benefit to mathematics, and very little except for code transfers into code. Strong domain transfer, training entirely on one domain while achieving near in-domain performance, appears feasible for curriculum-style benchmarks such as Physics MMLU, but not for more specialized, expert-authored evaluations like GPQA-diamond. Gradient overlap offers a simple training-time signal that reflects these patterns and helps identify data most relevant for a given target domain. Overall, our results suggest that while gradient-based measures capture broad relationships between domains, the strength of domain substitution depends on the construction and difficulty of the downstream task.

In addition, our results should be viewed as preliminary. Gradient–based overlap provides a useful signal for domain relationships, but we have only examined a small subset of its potential implications. Future work includes testing whether these scores can predict data usefulness under varying compute budgets—for example, whether the highest-influence examples identified for a given budget correspond to those that actually improve downstream performance in controlled experiments. We also hope to study more systematically when strong domain transfer is possible, and what properties of a benchmark or domain pair make such transfer viable or unlikely. Our current findings offer an initial exploration of these questions, and we expect that a deeper analysis will reveal a more complete picture of how domain structure shapes transfer behavior in practice.

5 Contributions

NJ and YH contributed equally to the ideation of the project.

NJ conducted the experiments, collected the data, and conducted the analysis in section 2.1 and section 3.

YH conducted the analysis in section 2.2.

References

Daman Arora, Himanshu Gaurav Singh, and Mausam. Have llms advanced enough? a challenging problem solving benchmark for large language models, 2023. URL https://arxiv.org/abs/2305.15074.

Mayee F. Chen, Michael Y. Hu, Nicholas Lourie, Kyunghyun Cho, and Christopher Ré. Aioli: A unified optimization framework for language model data mixing, 2025. URL https://arxiv.org/abs/2411.05735.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin 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, 2025. URL https://arxiv.org/abs/2506.04178.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

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, 2025. URL https://arxiv.org/abs/2507.00432.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale, 2023. URL https://arxiv.org/abs/2303.14186.

Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. Estimating training data influence by tracing gradient descent, 2020. URL https://arxiv.org/abs/2002.08484.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data, 2024. URL https://arxiv.org/abs/2410.01560.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning, 2024. URL https://arxiv.org/abs/2402.04333.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining, 2023. URL https://arxiv.org/abs/2305.10429.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li,

Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.