How Instruction and Reasoning Data shape Post-Training: Data Quality through the Lens of Layer-wise Gradients

Ming Li[1], Yanhong Li[2], Ziyue Li[1], Tianyi Zhou[1]
[1]University of Maryland; [2]University of Chicago
{minglii, tianyi}@umd.edu
Project: https://github.com/MingLiiii/Gradient_Unified

Abstract

As the post-training of large language models (LLMs) advances from instruction-following to complex reasoning tasks, understanding how different data affect finetuning dynamics remains largely unexplored. In this paper, we present a spectral analysis of layer-wise gradients induced by low/high-quality instruction and reasoning data for LLM post-training. Our analysis reveals that widely-studied metrics for data evaluation, e.g., IFD, InsTag, Difficulty, and Reward, can be explained and unified by spectral properties computed from gradients' singular value decomposition (SVD). Specifically, higher-quality data are usually associated with lower nuclear norms and higher effective ranks. Notably, effective rank exhibits better robustness and resolution than nuclear norm in capturing subtle quality differences. For example, reasoning data achieves substantially higher effective ranks than instruction data, implying richer gradient structures on more complex tasks. Our experiments also highlight that models within the same family share similar gradient patterns regardless of their sizes, whereas different model families diverge significantly. Providing a unified view on the effects of data quality across instruction and reasoning data, this work illuminates the interplay between data quality and training stability, shedding novel insights into developing better data exploration strategies for post-training.

1 Introduction

Large language models (LLMs) have shown remarkable potential for various complex tasks [30, 27], yet their success in real-world applications hinges not just on model size but also on the quality of data used for training [1, 23, 31]. It is widely recognized that high-quality, diverse, and complex instruction-following examples during post-training [26, 5, 9, 14, 24, 10, 8], such as supervised fine-tuning (SFT), is crucial for eliciting generalization performance and reliability. There is growing recent interest in automated metrics that can evaluate data quality and select data for more efficient and effective post-training [2, 13, 12]. Moreover, beyond instruction-following, the reasoning capability of LLMs [17, 4] has also been proven to be largely dependent on data quality. For instance, s1.1 [16] utilizes only 1k difficult math problems and DeepSeek-R1 generated responses to elicit LLM's strong reasoning capability.

Despite the verified importance of data quality to post-training, **how the quality of instruction/reasoning data affect the gradients during post-training** still remains largely unexplored. In addition, **can we *unify* different data quality metrics?** Prior work has mostly treated data quality filtering as a preprocessing step, evaluating its benefits in terms of end-task performance. But a systematic study is lacking to reveal the mechanism of how data quality affects the training

First Workshop on Foundations of Reasoning in Language Models

dynamics. Meanwhile, there does not exist work that compares the learning dynamics induced by reasoning data and general instruction-following data or compares different data quality metrics' effects on post-training with data selection. A recent study of LLM post-training [11] on fast vs. slow thinking [7] for the first time analyzes the training dynamics on different data through the lens of layer-wise gradients. It discovers that learning detailed intermediate reasoning steps leads to smaller and more stable gradient updates than learning final answers only. Yet, this study focuses only on comparing "fast thinking" (a few answer tokens) vs. "slow thinking" (CoT paths) but does not extend to more challenging problems requiring more complex reasoning. The metric used to measure gradients is limited to magnitude instead of more sophisticated spectral properties.

We address these gaps by conducting **a layer-wise, gradient-based spectral analysis** of LLM post-training when using **instruction/reasoning data of low/high quality.** Our study spans multiple diverse LLM families, including Qwen2.5 [18], Llama3.1, Llama3.2 [6], Gemma2 [20]), and different sizes (1.5B - 14B), to ensure the generality of our findings. Inspired by s1 [16], which selects data by difficulty, for reasoning data, we compare the s1.1 data and GSM8K [3] data (response generated by DeepSeek-R1) as low/high-quality data, respectively. For general instruction-following data, we adopt WizardLM [26], Magpie [28], and OpenHermes 2.5 [21] for our experiments and leverage automatic metrics for data quality, such as IFD [13], InsTag [15], Difficulty, and Reward to partition these datasets into low/high-quality subsets. While the first three evaluate the instructions, the reward directly measures the response quality.

We developed several novel metrics measuring the spectral properties of gradients revealed by Singular Value Decomposition (SVD) (**SVD-based Metrics**) and Gradient **Similarity-based Metrics**, which are applied to the projection layers for Query, Key, Value, and Output in transformer architectures [22]: (1) *Nuclear Norm* measures the magnitude of the gradient, indicating the amount of changes and efforts required for post-training. (2) *Effective Rank* [19] captures the dimensionality of the gradient, indicating the diversity of the gradient directions. (3) *Same-layer Similarity* measures the alignment between gradients of different projections within the same layer. (4) *Adjacent-layer Similarity* measures the alignment of gradients between consecutive layers.

Main Contributions: In this study, we present a spectral analysis of layer-wise gradients in modern LLMs when finetuned on datasets of varying quality, namely high-quality, low-quality, instruction-following, and reasoning data. For the broad applicability of our conclusions, we conduct empirical investigations across diverse pretrained LLMs from multiple model families and on different datasets. Various automatic data evaluation metrics are used to split the data into low/high-quality partitions, and advanced reasoning data are also included for comparison. **Notably, we are the first to unify the effects of different data quality metrics and compare the instruction vs. reasoning data through the lens of layer-wise gradients.** Unlike existing work that focuses primarily on the gradient magnitude, we incorporate both SVD-based and similarity-based metrics, offering a more comprehensive analysis. These findings reveal previously overlooked gradient patterns and provide insights into enhancing the stability and efficiency of data synthesis and LLM training.

Our key findings:

1. Existing data quality metrics, e.g., IFD, InsTag, Difficulty, and Reward, can be ***unified*** due to consistent spectral properties of gradients, i.e., lower nuclear norms and higher effective ranks on high-quality data. This finding extends to both instruction and reasoning data, providing a unified view of the data quality effects.

2. Effective rank outperforms nuclear norm to distinguish low- vs. high-quality data. For reasoning data, s1.1 data yields the largest effective ranks across all experiments, suggesting high correlations between reasoning complexity and gradient diversity.

3. Within the same model family, layer-wise gradients' spectral properties remain consistent across different model sizes. In contrast, the gradient patterns diverge significantly across distinct model families, reflecting the unique learning dynamics of each model family.

4. Cosine similarities between gradients from the same layer and adjacent layers remain nearly zero for different types of data, so they cannot reflect data quality.

2 Methodology

2.1 Preliminaries

We investigate gradient behaviors under the most widely adopted SFT approach. Each data point in an SFT dataset D consists of a pair (x, y), where x is the instruction and y is the corresponding response. For the reasoning data, y concatenates both thinking tokens and response tokens. Let p_θ be a large language model with parameters θ. Under the SFT paradigm, p_θ is finetuned on each pair (x, y) by minimizing the following loss, where y_j denotes the j-th token of y, $y_{<j}$ denotes the preceding tokens, and l is the total length of y:

$$L_\theta \;=\; \frac{1}{l}\sum_{j=1}^{l} -\log\big(p_\theta(y_j \mid x,\, y_{<j})\big). \tag{1}$$

In this paper, we focus on the gradients of the layers associated with the attention mechanism [22], namely the *Query (Q)*, *Key (K)*, *Value (V)* projection layers and the final *Output (O)* projection layer. For simplicity, we denote the gradients of these projection layers by $G_{Q,i}$, $G_{K,i}$, $G_{V,i}$, and $G_{O,i}$ for each layer $i \in \{0, 1, \ldots, N-1\}$ in the LLM.

2.2 Gradient Metrics from Spectral Analysis

To quantitatively analyze these gradients, we employ two categories of metrics: **(1)** *SVD-based Metrics* (the **Nuclear Norm** and the **Effective Rank**) and **(2)** *Similarity-based Metrics* (the **Same-layer Similarity** and the **Adjacent-layer Similarity**). While the SVD-based metrics describe properties of an individual gradient matrix, the similarity-based metrics reveal how gradients compare across different projections or layers.

2.2.1 SVD-based Metrics

Consider a gradient matrix $G_{X,i} \in \mathbb{R}^{m\times n}$, where $X \in \{Q, K, V, O\}$ indicates the projection (Query, Key, Value, or Output), and i represents the index the Transformer layer. We can write its Singular Value Decomposition as

$$G_{X,i} \;=\; U\,\Sigma\,V^T,$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix containing the singular values $\sigma_1, \ldots, \sigma_{\min(m,n)}$, sorted in decreasing order. For simplicity, we omit all the subscripts (X and i) for SVD matrices and singular values.

Nuclear Norm. We measure the overall magnitude of the gradient matrix $G_{X,i}$ by its nuclear norm $\mathcal{N}_{X,i}$, which is formulated as

$$\mathcal{N}_{X,i} \;=\; \|G_{X,i}\|_* \;=\; \sum_{j=1}^{\min(m,n)} \sigma_j. \tag{2}$$

A higher nuclear norm indicates a larger overall gradient scale, implying that the model parameters at that layer are being updated more significantly, further indicating a potential distribution shift between the response and the model to be trained.

Effective Rank. We measure how uniformly the singular values of $G_{X,i}$ are distributed by the effective rank $\mathcal{R}_{X,i}$. We normalize the singular values and formulate the effective rank as

$$\mathcal{R}_{X,i} = \exp\Big(-\sum_{j=1}^{\min(m,n)} \tilde{\sigma}_j \ln(\tilde{\sigma}_j)\Big)$$

$$\tilde{\sigma}_j = \frac{\sigma_j}{\sum_{k=1}^{\min(m,n)} \sigma_k}$$

If only a few singular values are large (i.e., the gradient is concentrated in just a few directions), the effective rank is small. If many singular values all contribute significantly, the effective rank is relatively larger. It measures how diverse the directions of the gradient are. A higher effective rank indicates the gradient is spread out over more directions, suggesting richer updates, whereas a smaller value means that only a few directions dominate the gradient directions.

2.2.2 Similarity-based Metrics

While the SVD-based metrics above characterize individual gradient matrices in isolation, it can be equally insightful to analyze how gradients relate across projections or layers. To this end, we introduce cosine similarity measures at two levels: within the same layer and across adjacent layers.

3 Experimental Setup

3.1 Models

We evaluate our method on several pretrained LLMs across multiple families, covering a range of parameter scales. Specifically, we use Qwen2.5 [18] in four configurations (1.5B, 3B, 7B, and 14B parameters), Llama3.1 [6] with 8B parameters, Llama3.2 in 1B and 3B configurations, and Gemma2 [20] in 2B and 9B configurations.

3.2 Datasets

Instruction-following data: WizardLM [26] is an instruction-following dataset created via an LLM-based "evolution" strategy that iteratively rewrites a set of initial prompts into more complex multi-step instructions, automatically generating high-complexity queries beyond what human annotators typically produce. **Magpie** [28] is a fully synthetic alignment dataset obtained by prompting an aligned language model, and we utilize the 300k high-quality subset selected by the authors for our source data. **OpenHermes 2.5** is a large-scale curated compilation of roughly one million instruction–response samples drawn from a diverse range of open-source and GPT-4-generated datasets, designed to maximize diversity and task coverage for robust fine-tuning.

Reasoning data: We employ two sources of reasoning data, each paired with step-by-step traces and final solutions generated by DeepSeek-R1 [4]: **s1.1K** data is provided by s1, which contains difficult math problems with responses generated by DeepSeek-R1. This data can be viewed as high-quality reasoning data since it succeeds in eliciting LLMs' reasoning capability with only $1,000$ samples. To curate the relatively low-quality reasoning data, motivated by s1's success in utilizing difficulty as the metric, we utilize the relatively easy math problems from **GSM8K** [3] with DeepSeek-R1 generated responses.

3.3 Data Quality Metrics to Partition Low/High-Quality Data

We adopt four automated data evaluation metrics to analyze instruction-following data quality. We focus on data evaluation metrics that do not rely on additional evaluation sets or training, which might lead to customized shifting, while we aim at the effects of original properties of the data instances.

IFD [13] quantifies the instruction-following difficulty of a sample by computing the ratio between the model's perplexity when predicting the response without the instruction and with the instruction. A higher IFD indicates that the model fails to benefit from the instruction, suggesting that the instruction is either ambiguous or unhelpful. Following **(author?)** [12], we use a small GPT2 model to efficiently compute IFD scores over large datasets.

InsTag [15] performs open-set multi-label tagging over instructions, capturing semantic attributes such as domain, task type, and intent. We use two derived metrics: (1) *instruction complexity*, defined by the number of tags per sample, and (2) *instruction diversity*, defined by tag vocabulary coverage across the dataset. Higher complexity scores typically correspond to more elaborate, multifaceted instructions. We use the per-sample complexity score as the filtering signal.

Reward Model Score uses *sfairXC/FsfairX-LLaMA3-RM-v0.1*, a reward model fine-tuned via preference modeling to predict human-aligned helpfulness [25]. Given a data pair, the model outputs a reward score reflecting predicted alignment with human preferences.

GPT-4o Difficulty Rating prompts the GPT-4o model to assign a difficulty score ($from 1 to 10$) to each instruction based on the perceived complexity, ambiguity, and reasoning depth required. This method approximates a human-aligned evaluation of instruction difficulty.

For each dataset described above, and for each metric, we select 200 samples with the highest scores and 200 with the lowest scores for calculating the gradients. This allows us to isolate what each

Dataset	Metrics	Nuclear Norm								Effective Rank							
		Proj	High	Low	Gap	**Proj**	High	Low	Gap	**Proj**	High	Low	Gap	**Proj**	High	Low	Gap
WizardLM	**IFD**	k	1.3	6.1	-4.8	q	1.4	4.6	-3.2	k	88.5	14.2	74.3	q	131.9	12.7	119.2
		v	2.5	10.9	-8.4	o	2.7	8.3	-5.6	v	110.9	13.6	97.3	o	165.2	12.5	152.7
	InsTag	k	1.9	4.1	-2.2	q	1.9	3.2	-1.3	k	95.6	19.9	75.7	q	141.2	21.5	119.7
		v	3.1	7.6	-4.5	o	3.3	5.9	-2.6	v	120.7	20.8	99.9	o	180.5	22.3	158.2
	Difficulty	k	1.9	3.5	-1.6	q	1.9	2.7	-0.8	k	91.5	18.5	73.0	q	133.2	20.0	113.2
		v	3.1	6.8	-3.7	o	3.3	5.2	-1.9	v	114.9	19.1	95.8	o	167.8	20.7	147.1
	Reward	k	1.2	4.3	-3.1	q	1.2	3.6	-2.4	k	91.5	35.6	55.9	q	131.4	38.6	92.8
		v	2.1	7.8	-5.7	o	2.3	6.5	-4.2	v	113.2	36.9	76.3	o	166.8	41.4	125.4

Table 1: Nuclear norms and effective ranks of gradients calculated from high- or low-quality data selected by different metrics. *High* represents the high-quality subset, while *Low* represents the low-quality subset. *Gap* is calculated by *High* − *Low*. **All data quality metrics consistently identify the high-quality data, which show similar spectral properties of gradients: lower nuclear norms and higher effective ranks. Hence, the gradient properties can unify all metrics.**

metric considers "good-quality" and "low-quality" data and to conduct gradient-based analysis across these contrasting subsets.

4 Empirical Analysis

4.1 Instruction-Following Data

4.1.1 Unifying Different Data Quality Metrics

In this section, we compare the effects of different data filtering metrics for general instruction-following data toward the gradient properties, including the averaged nuclear norms and effective ranks, as shown in Table 1. The nuclear norms in the table are calculated across the layers, $\mathcal{N}_X = \sum_{i=0}^{N-1} \mathcal{N}_{X,i}$, and the effective ranks are similar. Specifically, we use the WizardLM [26] data as the source data and calculate gradients on the Qwen2.5-7B model, and the data filtering metrics include IFD [13] (calculated on the Qwen2.5-7B model), InsTag [15], Difficulty, and Reward.

As shown in the table, across all metrics, the high-quality subsets exhibit substantially smaller averaged nuclear norms in the layer-wise gradients, represented by the consistent negative values (in red) for the Gap columns. Since the nuclear norm measures the overall magnitude of gradient updates; thus, a smaller value suggests the model requires less energy to adapt to high-quality data. It further indicates that high-quality data should be aligned with the learned knowledge of pretrained LLMs. At the same time, high-quality subsets also yield consistently larger effective ranks in their gradients represented by the large positive values (in green) for the Gap columns. A higher value suggests that more update directions are activated, which means high-quality data leads to richer, more multi-dimensional parameter updates, which likely improves the model's ability to generalize and capture nuanced features of the instruction pairs.

Our finding illustrates that multiple different definitions of "data quality" can converge to overarching gradient properties, revealing a unified view by gradient-based spectral analysis.

4.1.2 Effects of Original Dataset Qualities

In this section, we aim to investigate the potential ineffectiveness of gradient properties for distinguishing high- or low-quality data. Similar to the previous settings, we split the data from different sources, WizardLM [26], OpenHermes 2.5 [21], and Magpie [28], into high- and low-quality subsets, calculating their effects on gradients based on Qwen2.5-7B. The nuclear norm and effective rank changes with layer indexes are shown in Figure 1. The shapes of gradient curves are almost kept the same for one specific model, whatever the data quality.

Moreover, we find that when an entire dataset is already composed of fairly clean and coherent instruction–response samples, e.g., the Magpie dataset shown in the third row, nuclear norms offer limited discriminative power in distinguishing it further, represented by the similar scales for both types of data. We hypothesize that once instructions and responses cross a certain threshold of clarity and consistency (i.e., minimal noise or confusion), the gradient magnitudes needed to adapt to these subsets no longer diverge sharply. In other words, the nuclear norm may effectively "saturate" and stop registering small but meaningful differences among these relatively high-quality groups. On the

Figure 1: Low/high-quality data (measured by Reward) and their gradient properties (nuclear norms and effective ranks) across layers on diverse datasets including WizardLM, OpenHermes 2.5, and Magpie. The y-axis scales are kept the same for nuclear norms, while different for effective ranks, due to the large discrepancy. **For each specific model, the shapes of the gradient curves derived from different data sources are almost the same. The nuclear norm fails to reflect the quality discrepancies between datasets, while the effective ranks still works promisingly, e.g., Magpie has higher rank than others.**

Dataset	Metrics	Nuclear Norm								Effective Rank							
		Proj	High	Low	Gap	**Proj**	High	Low	Gap	**Proj**	High	Low	Gap	**Proj**	High	Low	Gap
Magpie	**Difficulty**	k	1.7	1.7	0.0	q	1.8	1.8	0.0	k	95.9	83.8	12.1	q	153.3	124.4	28.9
		v	3.0	3.1	-0.1	o	3.3	3.3	0.0	v	118.0	102.7	15.3	o	195.1	151.7	43.4
Reasoning	**Difficulty**	k	1.0	1.3	-0.3	q	1.3	1.5	-0.2	k	138.8	106.1	32.7	q	361.2	203.3	157.9
		v	1.9	2.4	-0.5	o	2.5	2.8	-0.3	v	170.4	126.7	43.7	o	509.9	263.1	246.8

Table 2: Comparing the gradient properties between instruction data vs. reasoning data. For the reasoning data, *High* denotes data sampled from s1.1, and *Low* denotes sampled from GSM8K with DeepSeek-R1 responses. Reasoning data shows lower nuclear norms and higher effective ranks compared with instruction data. **Our analysis of gradients unifies the quality evaluation for reasoning and instruction data on both higher effective ranks reflecting higher quality. Moreover, the metric distinguishes low-/high- quality data by large gaps.**

contrary, effective ranks remain sensitive to smaller quality disparities, even within a dataset already recognized for solid instruction–response fidelity. Consequently, effective rank acts as a finer-grained lens, revealing that while both subsets are indeed "good" enough to produce stable gradients, the higher-quality examples still manage to activate a broader range of update directions.

4.2 Reasoning Data

4.2.1 Unifying Quality Evaluation of Instruction and Reasoning Data

Beyond analyzing general instruction-following data, the recent surge in reasoning models encourages us to further explore the effects of reasoning data on LLM gradients. Motivated by recent success on eliciting models' reasoning capabilities by distilling from stronger reasoning models through simple SFT, e.g., s1 [16], LIMO [29], DeepSeek-R1 Distilled Qwen [4], we formulate the exploration on reasoning data in the same structure as on instruction-following data. Compared with general instruction-following data, a key difference of advanced reasoning data is the utilization of dynamic long CoTs and the accordance with the test-time scaling law. To distinguish this data from general instruction-following data, we notate them as reasoning data. As illustrated by s1, $1,000$ difficult math problems are sufficient to elicit LLMs' reasoning capabilities; thus, we ask: **Does reasoning**

Figure 2: Model size scaling law for gradient properties. **Within the same model family, the layer-wise gradient statistics and dynamics are relatively consistent. Gradients on larger models exhibit better capabilities to distinguish data quality**, revealed by the increasing y-axis scales from the 1.5B model to 14B model.

data have similar effects to gradients with previous general instruction data? Can the quality of advanced reasoning data be further distinguished by gradient properties like nuclear norm and effective rank?

To formulate our experiments, we utilize the s1.1 data as the high-quality subset and GSM8K [3] (responses also generated by DeepSeek-R1 [4]) as the low-quality subset for our reasoning data experiments. The results are shown in Table 2, in which the gradients are calculated based on Qwen2.5-7B, comparing the general instruction-following data with the reasoning data. Surprisingly, our experiments demonstrate that the same gradient-derived metrics, nuclear norms and effective ranks, remain applicable for the advanced reasoning data: (i) *Reasoning vs. Instruction data*: Reasoning data, even the lower-quality subset, leads to lower nuclear norms and higher effective ranks compared with previous high-quality instruction data, suggesting the much higher data quality of recent reasoning data. (ii) *Higher- vs. Lower-quality reasoning data*: Even for the reasoning data with supreme data quality, there still exists a consistent and unified trend with the previous instruction data, i.e., smaller nuclear norms and larger effective ranks for higher-quality data. Moreover, the gaps in effective ranks between high- and low-quality subsets are more pronounced for reasoning data than for general instruction data, which might provide a potential explanation on why s1.1 can reach such a promising performance with only 1000 data.

We are the first to investigate and compare the effects of general instruction-following and reasoning data toward LLM gradients in the training process. We reveal that both data types can be *unified* into a consistent pattern of gradient-based signals regarding quality, providing a unified view for understanding the quality effects.

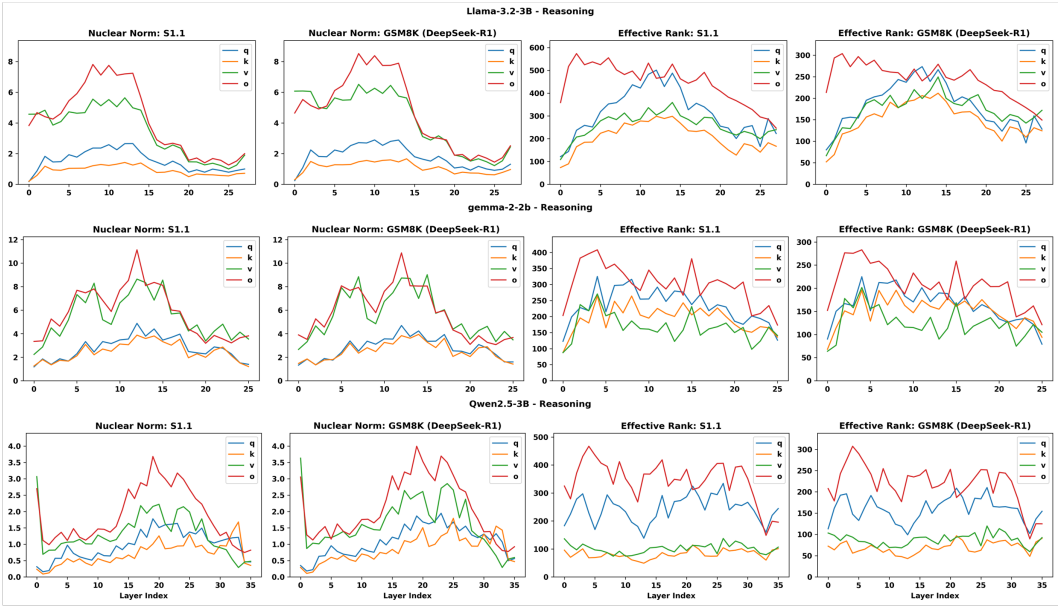

Figure 3: Gradient properties across different model families. **The gradient dynamics of the same data on different model families are largely different. This might be caused by their distinct model structures or training recipes and may reflect their different capabilities.**

Dataset	Metric	Same-layer Similarity						Adjacent-layer Similarity					
		Proj	High	Low	**Proj**	High	Low	**Proj**	High	Low	**Proj**	High	Low
Magpie	**Difficulty**	k - v	-0.7e-3	-0.7e-3	q - o	0.0e-3	0.0e-3	k v	-0.1e-3 0.3e-3	-0.1e-3 0.1e-3	q o	0.1e-3 0.4e-3	-0.1e-3 0.3e-3
Reasoning	**Difficulty**	k - v	0.0e-3	0.0e-3	q - o	0.1e-3	-0.9e-3	k v	-2.0e-3 1.0e-3	-2.0e-3 2.0e-3	q o	0.0e-3 1.0e-3	0.2e-3 2.0e-3

Table 3: Gradient similarity metrics remain excessively small and cannot reflect the differences between instruction/reasoning data of low/high-quality. **It shows that layer-wise gradients in LLM post-training are nearly orthogonal, indicating that the similarity of gradients is not an effective indicator of data quality.**

4.2.2 Effects of Model Sizes

Figure 2 presents the gradient curves of LLMs with different sizes in the Qwen2.5 family on our reasoning data. The shapes of the gradient curves for reasoning data are almost the same as the instruction-following data, which further verifies our unified view on both types of data. Moreover, a consistent observation is that the overall shape of these layer-wise curves remains relatively (not strictly) stable as we move from smaller to larger models. For instance, GSM8K-based fine-tuning triggers a peak in nuclear norm at the mid layers for Qwen2.5-1.5B, a qualitatively similar, albeit scaled, peak in Qwen2.5-14B also can be identified.

Moreover, another interesting finding is that larger models tend to amplify the distinction between high- and low-quality subsets. Specifically, the gaps in effective rank between s1.1 and GSM8K grow accordingly when moving to bigger models. In other words, the larger model is more sensitive to whether the provided reasoning path is coherent and informative.

4.2.3 Effects of Model Families

In this section, we broaden the scope to compare entirely different model families, including Qwen2.5-3B, Llama3.2-3B, and gemma2-2B, each possessing distinct pretraining recipes and architectural configurations. The first notable observation is that the layer-by-layer "shape" of the gradients can vary significantly among families. For instance, Llama3.2-3B might exhibit consistently higher nuclear norms in its early layers compared to Qwen2.5-3B, reflecting differences in embedding or attention initialization. Despite these baseline discrepancies, the relative gap between high- and low-quality reasoning data persists across all families. In other words, regardless of how each model

is architected or pre-trained, high-quality data still yields smaller nuclear norms and larger effective ranks. This cross-family analysis suggests that reasoning data can be a valuable resource regardless of the specific LLM architectures. At the same time, our analysis shows the existence of family-specific "fingerprints" for nuclear norms and effective ranks, reflecting architectural and pretraining differences, which might potentially be useful for a better understanding of the LLM architectures.

4.2.4 Similarity-based Metrics

While our SVD-based metrics are proved consistently effective at capturing data-quality differences, similarity-based metrics, namely, same-layer similarity and adjacent-layer similarity, do not appear to yield meaningful signals in our experiments, as shown in Table 3. We keep the values in the same magnitude for easier comparison. In the table, we compare similarity measures for both instruction and reasoning data across high- and low-quality subsets. Regardless of dataset type or quality level, the reported cosine similarities remain extremely close to zero, with minimal observable variation. These low similarities suggest that the gradients for LLM SFT are nearly orthogonal, indicating that similarity on gradients is not an effective indicator of data quality.

5 Conclusion

We introduce a unified gradient-based framework for analyzing how varying data quality, ranging from general instruction-following to reasoning data, shapes the finetuning of LLMs. By examining the layer-wise gradients, we show that different quality metrics converge on remarkably similar gradient signatures, specifically, smaller gradient magnitudes and broader gradient directions for high-quality data. Notably, this pattern holds across multiple model families and parameter scales and, for the first time, reveals how reasoning data induce even higher effective ranks and thus richer parameter updates.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[2] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training a better alpaca with fewer data, 2023.

[3] Karl Cobbe, Vineet Kosaraju, Mohammad 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, 2021.

[4] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[5] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

[6] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle

Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024.

[7] Daniel Kahneman. Thinking, fast and slow. *Farrar, Straus and Giroux*, 2011.

[8] Ming Li, Han Chen, Chenguang Wang, Dang Nguyen, Dianqi Li, and Tianyi Zhou. Ruler: Improving llm controllability by rule-based data recycling, 2025.

[9] Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, and Tianyi Zhou. Reflection-tuning: Recycling data for better instruction-tuning. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

[10] Ming Li, Pei Chen, Chenguang Wang, Hongyu Zhao, Yijun Liang, Yupeng Hou, Fuxiao Liu, and Tianyi Zhou. Mosaic-it: Free compositional data augmentation improves instruction tuning, 2024.

[11] Ming Li, Yanhong Li, and Tianyi Zhou. What happened in llms layers when trained for fast vs. slow thinking: A gradient perspective, 2024.

[12] Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14255–14273, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[13] Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7595–7628, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[14] Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*, 2023.

[15] Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models, 2023.

[16] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025.

[17] OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc,

Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024.

[18] 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.

[19] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007.

[20] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024.

[21] Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[23] Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning large language models with human: A survey, 2023.

[24] Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. Lamini-lm: A diverse herd of distilled models from large-scale instructions, 2024.

[25] Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint, 2024.

[26] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions, 2023.

[27] Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024.

[28] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing, 2024.

[29] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

[30] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.

[31] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023.