Massive Supervised Fine-tuning Experiments Reveal How Data, Layer, and Training Factors Shape LLM Alignment Quality

Yuto Harada[1,2*], **Yusuke Yamauchi**[1,2*], **Yusuke Oda**[1,3], **Yohei Oseki**[1,2], **Yusuke Miyao**[1,2†], **Yu Takagi**[4†]
[1]NII LLMC, [2]The University of Tokyo, [3]NAIST, [4]Nagoya Institute of Technology,
{harada-yuto, yamauchi_y}@nii.ac.jp
yusuke@is.s.u-tokyo.ac.jp takagi.yu@nitech.ac.jp

Abstract

Supervised fine-tuning (SFT) is a critical step in aligning large language models (LLMs) with human instructions and values, yet many aspects of SFT remain poorly understood. We trained a wide range of base models on a variety of datasets including code generation, mathematical reasoning, and general-domain tasks, resulting in 1,000+ SFT models under controlled conditions. We then identified the dataset properties that matter most and examined the layer-wise modifications introduced by SFT. Our findings reveal that some training–task synergies persist across all models while others vary substantially, emphasizing the importance of model-specific strategies. Moreover, we demonstrate that perplexity consistently predicts SFT effectiveness, often surpassing superficial similarity between the training data and the benchmark, and that mid-layer weight changes correlate most strongly with performance gains. We release these 1,000+ SFT models and benchmark results to accelerate further research. All resources are available at https://github.com/llm-jp/massive-sft.

1 Introduction

Recent advances in large language models (LLMs) have greatly improved natural language understanding and generation. However, pretrained LLMs often fail to align with human intentions or specific tasks (Ouyang et al., 2022), motivating alignment methods. Supervised fine-tuning (SFT) trains models to follow human instructions and remains a widely used and effective approach for improving downstream performance (Wei et al., 2022; Guan et al., 2024).

Although recent works have explored how model size and training-data characteristics influence downstream tasks in the context of SFT (Jin and Ren, 2024; Dong et al., 2024), large-scale research specifically examining which aspects of SFT datasets benefit different base models remains limited. While some studies compare or analyze publicly available models (Oyama et al., 2025), these are not controlled experiments and often introduce biases, such as favoring certain model families. Consequently, it remains unclear how SFT of various models on different datasets affects benchmark performance, how relationships among datasets and benchmarks vary across models, and which internal weights are most responsible for these effects. Furthermore, there are several SFT training approaches including Low-Rank Adaptation (LoRA) (Hu et al., 2022), and there is ongoing debate about the optimal amount of data required (Zhou et al., 2024; Chen et al., 2023); however, there has yet to be a comprehensive, quantitative comparison. Hence, a comprehensive examination of these issues on SFT is urgently needed.

In this study, we trained twelve base models on multi-domain datasets, produced a large suite of SFT models, and evaluated them across diverse tasks (Figure 1). Specifically, we address the following Research Questions (RQs):

1. How do models, training data, and benchmarks interact for downstream performance? Do any training datasets yield consistent gains across models, or are improvements model-specific? Are relationships between datasets and benchmarks stable across models?
2. Which properties of the training data used for SFT affect downstream performance?
3. Which layers in the model are most critical for SFT? Are there universal patterns across different models?
4. How do factors debated in SFT, including training method, sample size, and cross-lingual transfer, relate to performance?

*Equal Contribution.
†Corresponding authors: Yusuke Miyao and Yu Takagi

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 22360–22381
November 4-9, 2025

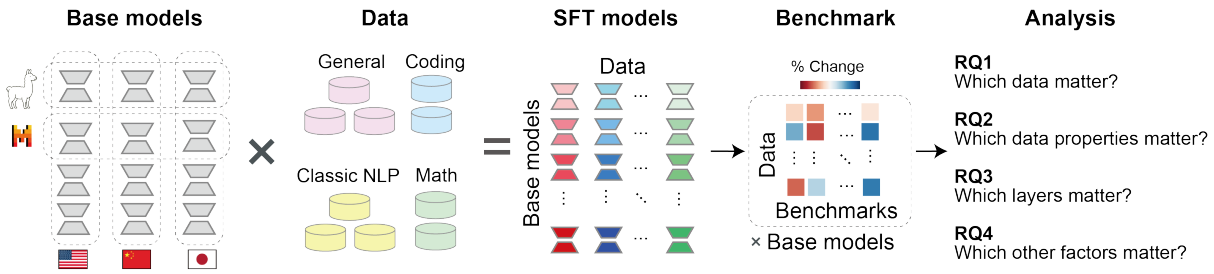


Figure 1: Overview of this study. We conduct SFT on numerous combinations of base models and training data. These models are evaluated on a variety of benchmark tasks to comprehensively examine the relationships among the base models, training data, and benchmark tasks.

In summary, our contributions are as follows:

Large-Scale, Integrated Evaluation By systematically performing SFT on multiple base models and various training datasets, we uncover the complexity of relationships among models, data, and downstream tasks. While the relationships between training data and evaluation tasks follow broadly similar patterns across models, they also exhibit model-specific characteristics.

Revealing a Simple "Perplexity Is Key" Law We find that training data with lower perplexity for the base model consistently leads to greater improvements in downstream performance. In contrast, factors once considered crucial, such as content similarity between training and evaluation data or tokenizer compatibility, do not exhibit as strong an effect as perplexity.

Strong Correlation Between Mid-Layer Weight Changes and Performance We observe that changes in mid-layer weights correlate more strongly with downstream performance gains than changes in either the top or bottom layers. Indeed, intrinsic dimensionality analysis of embeddings revealed that the embedding space begins to diverge substantially from the base model at mid-layer positions, suggesting these layers actively expand the model's representational subspace during SFT. This pattern appears consistent across multiple models, offering critical insights for efficient fine-tuning and model monitoring.

Embedding the SFT Landscape Projecting the log-likelihood vectors of fine-tuned models into a common latent space lets us compare diverse training dynamics in one coordinate system. The resulting map shows that the global layout is determined by model family rather than training corpus, that checkpoints from successive epochs converge toward a shared instruction-following region, that enlarging the instruction set from 1k to 20k nudges models only slightly outward from this center, and that LoRA trajectories almost perfectly overlap those of full-parameter tuning.

Resource Release for Future Research All fine-tuned models produced in this study are publicly released. We expect this comprehensive set of models to accelerate deeper investigations of SFT and to foster rapid progress in the field.

2 Related Work

The role of training data characteristics in SFT has been highlighted in many prior studies. For instance, mixing code-generation data has been suggested to enhance a model's reasoning and logical abilities (Dong et al., 2024). Similarly, incorporating instruction data that includes procedural knowledge could improve mathematical reasoning (Ruis et al., 2024). Furthermore, considering task relevance when selecting datasets can lead to more robust performance (Huang et al., 2024; Zhang et al., 2024).

While early work focused on how to fine-tune, comparing full-parameter updates against LoRA (Ivison et al., 2023; Zhuo et al., 2024; Dettmers et al., 2024; Zhao et al., 2024b; Biderman et al., 2024), or debating sample size (Zhou et al., 2024; Zhao et al., 2024a; Chen et al., 2023). More recent studies have shifted attention to the statistics of the training data itself. For example, Jin and Ren (2024) and Wu et al. (2025) independently show that lower perplexity and moderate sequence length are stronger predictors of SFT success than sheer volume.

Overall, most studies focus on particular models or tasks, and there remains a lack of comprehensive, large-scale evaluations across multiple models. This study aims to offer a broader perspec-

tive by controlling for model, data, and fine-tuning methods on a larger scale, thus providing more integrated insights into SFT behavior.

3 Methods

This section describes the base models, SFT procedures, and evaluation benchmarks.

3.1 Base Models

We employed a total of 12 models with approximately 7B parameters each across English, Chinese, and Japanese for SFT experiments. Specifically, we selected **English models**: OLMo-7B (Groeneveld et al., 2024), Llama3-8B (Dubey et al., 2024), Mistral-7B (Jiang et al., 2023), and Gemma2-9B (Team et al., 2024); **Chinese models**: Qwen2.5-7B (Yang et al., 2024), Chinese-Llama3-8B (Cui et al., 2023), Chinese-Mistral-7B (Hsu et al., 2024), and Yi1.5-9B (AI et al., 2025); and **Japanese models**: LLMjp-3-7B (LLM-jp et al., 2024), Llama3-Swallow-8B (Fujii et al., 2024), Swallow-Mistral-7B (Fujii et al., 2024), and Sarashina2-7B[1]. By comparing these diverse models, we investigate not only cross-lingual differences but also behaviors during continual pretraining within model families such as the Llama family (Llama3, Chinese-Llama3, Llama3-Swallow) and the Mistral family (Mistral, Chinese-Mistral, Swallow-Mistral). To facilitate fair comparison at the peak effectiveness of instruction-tuning, all base models used in this experiment had not undergone any subsequent post-training. More information on each model can be found in Appendix A.

3.2 Training Datasets

We utilized 10 distinct datasets categorized into 4 major groups. Although our base models cover English, Chinese, and Japanese, all training datasets used for SFT are exclusively in English. Specifically, we selected **General Tasks**: Alpaca (Taori et al., 2023), LIMA (Zhou et al., 2024), and UltraChat (Ding et al., 2023); **Coding Tasks**: CodeAlpaca (Chaudhary, 2023) and Magicoder (Wei et al., 2024); **Math Tasks**: OpenMathInstruct (Toshniwal et al., 2024) and MathInstruct (Yue et al., 2023); and **Classic NLP Tasks**: FLAN (Wei et al., 2022). The FLAN dataset further consists of 3 subcategories. FLAN Knowledge includes BoolQ (Clark et al., 2019), NaturalQuestions (Kwiatkowski et al., 2019b), and TriviaQA (Joshi et al., 2017). FLAN Reasoning includes ARC-Easy & Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2019), and PIQA (Bisk et al., 2020). FLAN Comprehension includes QuAC (Choi et al., 2018) and SQuAD v2 (Rajpurkar et al., 2018). The categorization of FLAN follows the criteria defined in Dubey et al. (2024); Contributors (2023).

To uniformly compare a wide variety of base models, all datasets were preprocessed under consistent conditions. Initially, samples exceeding the maximum sequence length supported by all models' tokenizers were removed, as overly long samples cannot be adequately learned. Subsequently, either 1k or 20k samples were randomly extracted from each dataset. Further details on the training datasets are provided in Appendix B.

3.3 Training Settings

We trained a total of 1,070 models by varying several conditions. First, all 12 models underwent both full-parameter and LoRA training with a sample size of 1k for each individual dataset. Additionally, we conducted training using a combined dataset (All Dataset) to assess the effect of mixing all data.

For further validation, we conducted additional experiments using 3 primary models (OLMo, Qwen, and LLM-jp), focusing on the impact of dataset size by comparing training results using 1k and 20k samples. In this specific experiment, the learning rate schedule was switched from cosine (used in regular training) to constant to isolate the effect of dataset size.

Through preliminary experiments, we determined optimal hyperparameters for both full-parameter fine-tuning and LoRA, ensuring that the supervised fine-tuning process was conducted under stable and well-tuned conditions. Details of the preliminary experiments are provided in Appendix C, while training configurations, computational costs, and a few exceptional cases where training did not complete successfully are described in Appendix D.

3.4 Evaluation

We evaluated all models on downstream tasks using OpenCompass[2] (Contributors, 2023), a large-scale evaluation tool. We evaluated model performance across 12 benchmark datasets spanning 5 categories: covering **Math** (MATH (Hendrycks

[1]https://huggingface.co/sbintuitions/sarashina2-7b

[2]We used the GitHub repository from OpenCompass: https://github.com/open-compass/opencompass

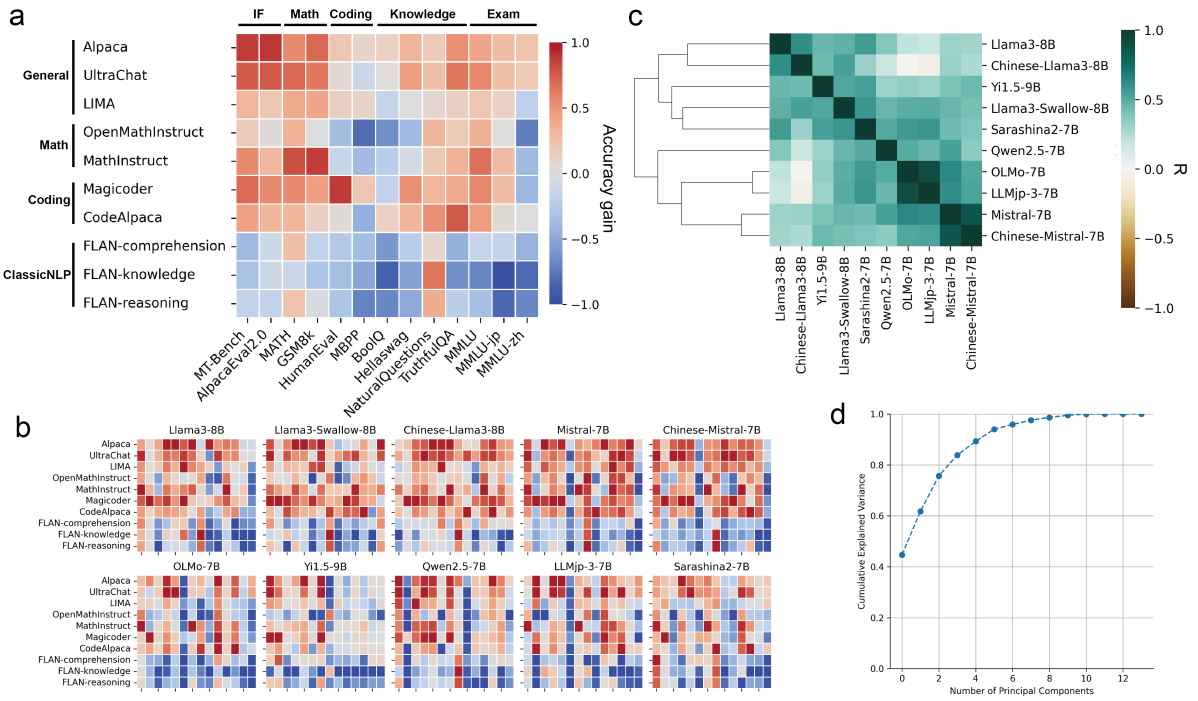


Figure 2: **a** Average of the performance change for diverse benchmarks from the each baseline model after SFT on each training dataset. Each column is min-max scaled to the $[-1, 1]$ range. **b** The performance changes visualized for each model individually. **c** Pairwise correlation matrix of performance changes across all SFT models, with the corresponding hierarchical-clustering dendrogram superimposed. **d** The cumulative explained variance ratio obtained by applying PCA to all concatenated results from **b**.

et al., 2021c), GSM8K (Cobbe et al., 2021)), **Coding** (HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021)), **Knowledge** (BoolQ (Clark et al., 2019), NaturalQuestions (Kwiatkowski et al., 2019a), TruthfulQA (Lin et al., 2022)), **Examination** (MMLU (Hendrycks et al., 2021b,a), MMLU-zh (Li et al., 2023a), MMLU-jp) and **Instruction-following** (MT-Bench (Zheng et al., 2023), AlpacaEval v2.0 (Li et al., 2023b)). A detailed description is provided in the Appendix E. As all models were trained in a zero-shot instruction-response format, we focus primarily on zero-shot inference results in our evaluation. Gemma2-9B and Swallow-Mistral-7B were excluded due to inconsistent evaluation conditions, and we report results mainly for the remaining 10 models.

4 Results

4.1 RQ1. Relationship Among Models, Training Data, and Downstream Tasks

First, we examine how various base models interact with different training datasets and how these relationships shape downstream performance. We aim to determine whether certain datasets provide uniform benefits across models or if each model exhibits unique sensitivities. To this end, we analyze evaluation results obtained by fine-tuning each of the ten base language models with each of the ten SFT training datasets, every dataset containing 1k examples.

Figure 2a visualizes the relationship between training datasets and downstream tasks when aggregating results across all models. Some datasets show clear improvements for multiple tasks, while others offer minimal, or even negative gains. For instance, Alpaca and UltraChat generally deliver consistent performance boosts, whereas FLAN is detrimental to most tasks (except Natural Questions, which aligns with its domain). In addition, MathInstruct and OpenMathInstruct particularly boost MATH and GSM8K, whereas Magicoder benefits coding benchmarks yet still improves a wider task range than the math corpora. Notably, English-only SFT already transfers to Japanese (MMLU-jp) and Chinese (MMLU-zh) evaluation—see Appendix F for a dedicated cross-lingual analysis. It is also noteworthy that LIMA, a carefully curated dataset for SFT, did not yield substantial performance gains in our controlled setting compared to Alpaca and UltraChat.

Figure 2b plots these relationships separately

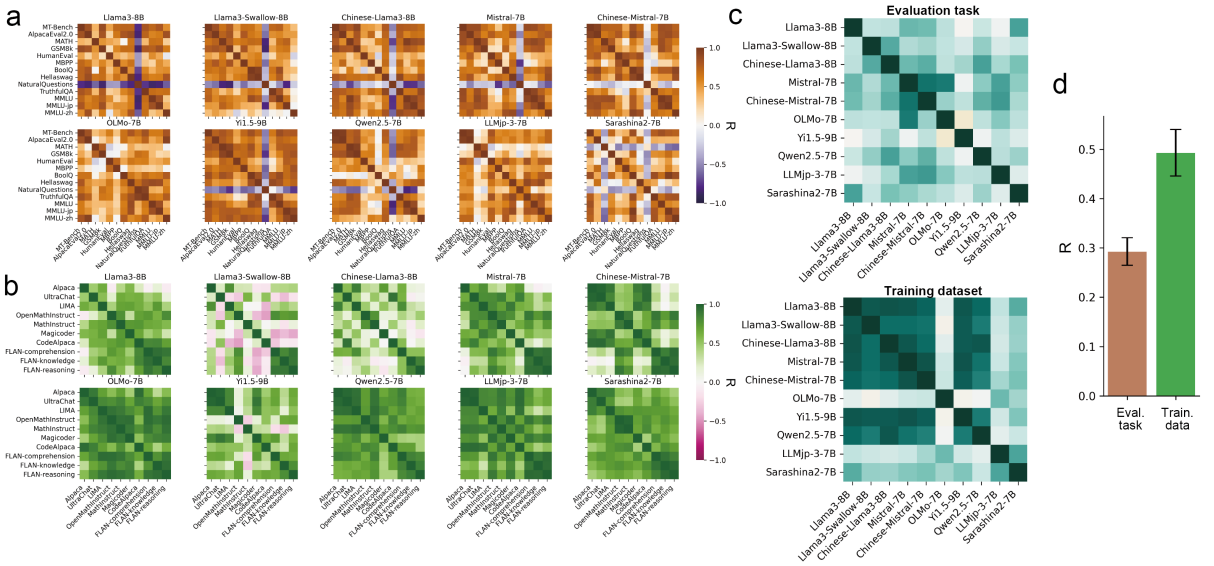


Figure 3: **a** Pairwise correlations between evaluation tasks in terms of performance improvements across training datasets. **b** Similar to **a**, but focusing on relationship between correlations between training datasets. **c** Model-to-model similarity for **a** (top) and **b** (bottom), respectively. **d** Comparison of the lower-triangle elements of the two similarity matrices in **c**.

for each model. Overall tendencies are similar, but there are also considerable differences across models—revealed only because we employed a unified experimental procedure. Some models benefit from almost all training data, whereas others demonstrate minimal gains.

In Figure 2c, we show a correlation matrix of performance gains across different models. As anticipated, models belonging to the same family exhibit high correlations, suggesting that even with additional training, the impact of SFT remains similar within each family. Surprisingly, the language in which a model was initially trained does not appear to substantially affect its overall similarity to others.

Figure 2c also reveals that, in general, the performance structures of the models are quite similar. To examine this more thoroughly, we vertically concatenated the data $\times$ benchmark matrices for each model, applied PCA, and then computed the cumulative explained variance ratio (Figure 2d). As shown, about five principal components explain over 90% of the total variance, indicating a considerable degree of similarity in how different datasets influence SFT outcomes. Nonetheless, certain differences among models persist.

Figure 3a, pairwise correlation performance improvements across training datasets, highlights that the similarity or synergy across training datasets varies substantially by model: the same pair of datasets could be complementary in one model but neutral or even conflicting in another. Conversely, Figure 3b, pairwise correlation across evaluation tasks, shows a consistency across models, suggesting that tasks requiring similar reasoning skills (e.g., Math tasks) remain closely grouped. A paired t-test on the lower-triangle distributions of Figure 3c shows that the correlations across evaluation tasks significantly exceeds that of training datasets ($p < 0.01$), confirming that the effects of training datasets is more diverse than evaluation tasks (Figure 3d). Overall, these findings underscore that while some training datasets offer consistent improvements, the degree of benefit often depends on the model. Furthermore, although fine-tuning effects on evaluation tasks are similar across models, those on training datasets are highly model-specific.

4.2 RQ2. Which Properties of Training Data Matter Most?

Next, we investigate which characteristics of training data most influence performance. Our focus includes perplexity, average token length, and semantic similarity to clarify which factors truly drive effective SFT.

As shown in Figure 4a, there is a clear negative correlation in many tasks and models between lower perplexity (w.r.t. the base model) and improved downstream performance. This implies that

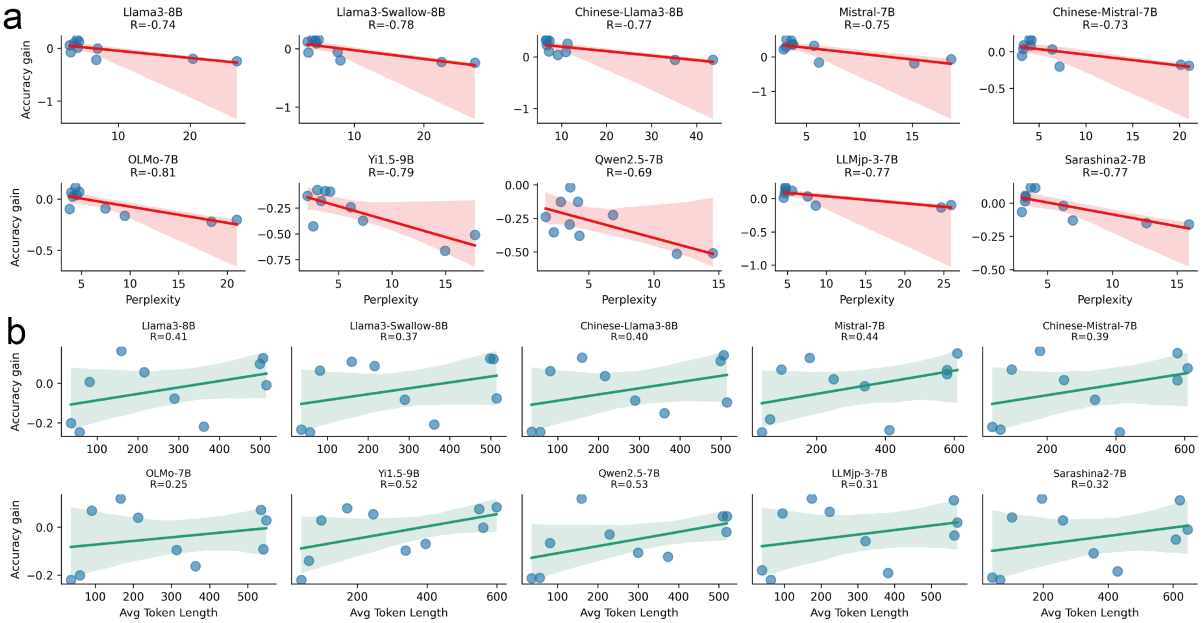


Figure 4: Analysis of training data properties that affect downstream performance. We compare perplexity (**a**), and token length (**b**) with the average performance changes of benchmark tasks for the SFT models, highlighting that lower perplexity is a strong predictor of higher performance.

data lying in a domain or language distribution already "understood" by the model can be leveraged more effectively in SFT.

Figure 4b reveals a modest correlation between the mean token length of a dataset and downstream performance, suggesting that simply using shorter or longer texts does not strongly drive better results. A prior study has reported that longer texts could be important for performance (Zhao et al., 2024a), and our findings partially support a straightforward link between text length and outcome quality.

Finally, we compare semantic embedding-based similarity between training and evaluation benchmark against performance improvement. Surprisingly, direct semantic similarity is not as strong a predictor as perplexity. Although we observe domain-specific gains (e.g., math data helps on Math tasks, code data helps on coding tasks), a broader trend indicates that linguistic and structural closeness (as reflected in perplexity) may be more decisive than topical resemblance alone. See Appendix G for the details.

In sum, perplexity relative to the base model emerges as a strong predictor of downstream gains, surpassing factors like token length or broad semantic alignment. It thus serves as a practical indicator of model–data compatibility for SFT, even when the base model's pretraining data are unknown.

4.3 RQ3. Layer-wise weight changes, their relationship to performance, and the effect of SFT on representational dimensionality

We then explore how model parameters shift during fine-tuning by analyzing layer-wise weight updates across multiple models in detail. Our goal is to identify which layers are most critical in translating SFT into performance gains.

Figure 5a plots two curves: specifically, the blue line is the Pearson correlation between weight-delta magnitude and overall accuracy gain, whereas the orange line shows the raw weight-delta magnitude. The orange line grows toward upper layers, yet the blue line peaks in the middle, indicating that the largest edits are not the most consequential ones. Rather, we find that the middle layers in particular exhibit the strongest positive correlation with performance gains.

Figure 5b compares the similarity of these layer-wise change patterns across different models. Even though models differ at the architectural level, their mid-layer updates under SFT can follow surprisingly similar trajectories. Still, some model-specific nuances remain.

Figure 5c quantifies cross-model similarity at each layer. We vectorize the SFT weight change per layer for each model, compute its correlation with the corresponding vectors from all other models, and average the resulting pairwise correla-

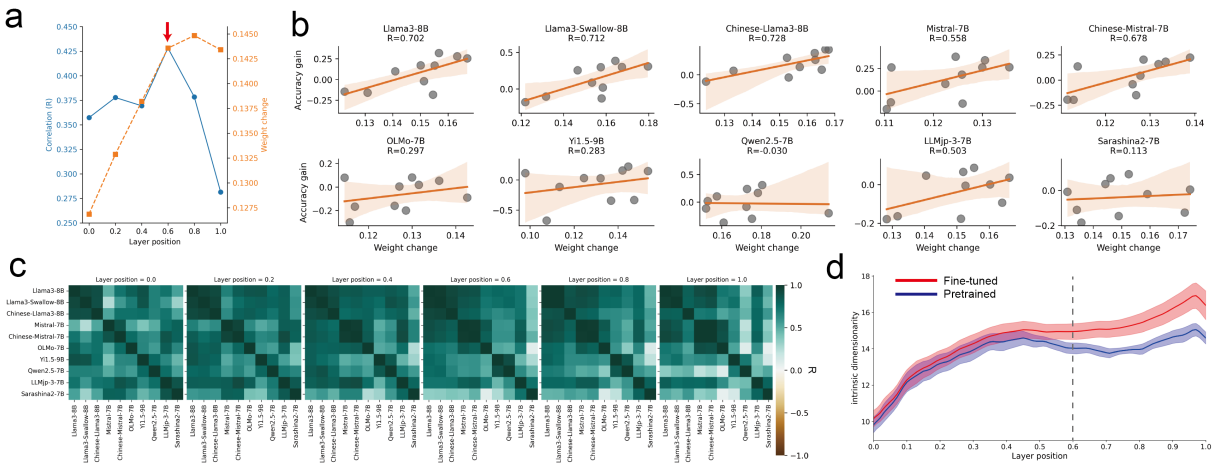


Figure 5: Layer-wise weight changes and their correlations with performance improvements. **a** Blue line indicates correlation coefficients between the amount of weight change from the base model and the overall improvement in accuracy, plotted as a function of layer position (0 = input; 1 = output). Compared to early and late layers, the mid-layers (0.6, indicated by red arrow) exhibit the strongest correlation. Orange line indicates the amount of weight change from the base model. **b** Focusing on the mid-layer (0.6), examining the relationship between the amount of weight change and accuracy change for each model reveals a robust correlation across all models. **c** Correlations calculated across models between weight changes from the base model and those from models trained on specific data. Again, the mid-layers show the strongest model-to-model correlation. **d** Intrinsic dimensionality (ID) of training-data embeddings before (blue line) vs. after SFT (red line). The divergence emerges around layer-position = 0.6 (dashed line), suggesting that mid-layer updates expand the representational subspace.

tions. The strongest agreement again lies in the mid-layers, suggesting that SFT enforces a shared instruction-following mechanism across models.

Figure 5d complements the weight-change analysis by quantifying how SFT alters the geometry of the training corpus in embedding space. For every layer we computed the intrinsic dimensionality (ID) of the sentence-level embeddings produced before and after SFT (methodological details and additional results in Appendix H). The difference between the fine-tuned and pretrained ID curves is minimal in the lower half of the network, but from layer-position = 0.6 onward the dimensionality increases sharply and remains elevated through the output layers. The inflection point coincides with the correlation peaks in Figure 5a, implying that mid-layer updates do more than reduce loss—they actively expand the model's representational subspace.

Our findings indicate that changes in the mid-layers show the strongest correlation with improved results, suggesting they play a pivotal role in capturing the benefits of SFT.

4.4 RQ4. Other Factors

Finally, we consider additional aspects of SFT, including LoRA versus full-parameter tuning, the effect of sample size, and cross-lingual transfer, each of which may influence final performance.

To disentangle the multiple factors in SFT, we mapped the 757 fine-tuned models—covering 10 base architectures × 10 training datasets and spanning LoRA vs. full-parameter updates, 1–10 training epochs, and sample sizes of 1k or 20k—into a common latent space using log-likelihood-vector projection (Oyama et al., 2025). For every model we computed a 1,950-dimensional vector of token-level log-likelihoods by randomly sampling 150 questions from each of the 13 evaluation tasks. t-SNE then embedded these vectors into two dimensions, giving five complementary views in Fig. 6.

Model families dominate. When points are coloured by **model** (Fig. 6a) the clusters group almost perfectly by architecture, whereas colouring by **training data** (Fig. 6b) produces only weak separation. Thus the inductive biases of the base model outweigh the specific SFT corpus in determining the final representation.

Epoch-wise trajectories converge. For the three checkpointed models (Qwen, LLM-jp, OLMo) we plot epochs 1–10 (Fig. 6c). Irrespective of dataset, trajectories spiral toward a common sub-region, suggesting that SFT gradually aligns the representations toward a shared "instruction-following" direction.

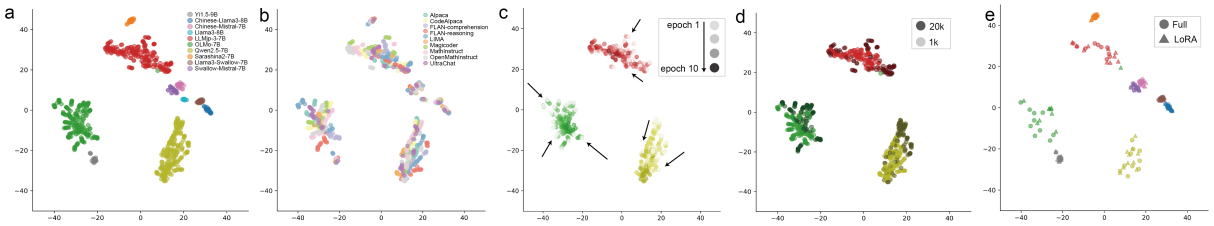


Figure 6: t-SNE visualization of log-likelihood vector. **a** Colour = model; **b** colour = training data; **c** epoch trajectories for three models; **d** colour = sample size; **e** shape = tuning method (circle = full, triangle = LoRA).

Small sample size is often sufficient. Colouring by **training-set size** separates models trained on 20k samples from those trained on 1k samples. The 20k-sample–trained points occupy the outer rim of the manifold more often, whereas the 1k-sample–trained points cluster nearer the core. Thus a compact 1k instruction set already supplies sufficient signal for effective instruction-tuning, while scaling up to 20k samples can sometimes pull the representation away from the optimum. Indeed, our quantitative evaluations showed no consistent accuracy advantage for the 20k-sample models over their 1k-sample counterparts.

LoRA vs. full-parameter fine-tuning. Shape-coding full-parameter models as circles and LoRA models as triangles reveals minimal separation; LoRA points are only slightly more peripheral. Quantitatively, full-parameter tuning still excels on reasoning-heavy maths tasks, but LoRA enjoys a small mean advantage on open-ended QA benchmarks.

Cross-lingual transfer persists. We also examined the effect of SFT effects on Japanese and Chinese MMLU variants (full results and plots are in Appendix F). While we only used English training datasets, performance gains on MMLU are strongly correlated on those of MMLU-jp and MMLU-zh. This supports the hypothesis that content overlap between benchmarks, rather than surface-level language similarity, governs cross-lingual transfer in SFT. See Appendix F for the details.

5 Discussion and Conclusion

We conducted a comprehensive set of SFT experiments involving multiple 7B-scale base models, diverse training datasets, and a wide array of downstream tasks. Our analysis revealed that, while certain dataset–task synergies are observed consistently across models, their effects can vary greatly depending on the specific model in question. Notably, perplexity emerged as a particularly robust predictor of SFT success, outperforming both topic similarity and average sequence length. Perplexity can be shaped by multiple latent properties of both the data and the model. Accordingly, we view it as a practical proxy for compatibility between the model and the data rather than a causal factor. Identifying the causal drivers behind this association is an important direction for future work.

Furthermore, mid-layer weight changes were found to correlate most strongly with performance improvements, indicating that critical adaptations often take place in these layers. By embedding every model checkpoint into a common latent space, we found that (i) model architecture exerts a stronger influence than the SFT corpus, (ii) training epochs drive diverse runs toward a shared instruction-compatible region, (iii) large instruction sets tend to relocate models toward the periphery—often reducing accuracy relative to smaller sets—and (iv) LoRA trajectories almost coincide with full-parameter ones, diverging only slightly on the periphery; this mirrors the small but systematic trade-off we observed between knowledge-heavy tasks (full-parameter advantaged) and open-ended QA (LoRA advantaged).

Contrary to the typical assumption that a dataset closely resembling the target task is best, we find data with a lower perplexity (where the model requires minimal additional learning or unlearning) generally yields more robust improvements. Additionally, our observations of code data helping math tasks suggest significant cross-domain transfer beyond simple topic alignment.

Discovering the importance of mid-layer changes could reshape fine-tuning strategies. Updating the mid-layers, or monitoring them closely, could provide more efficient or interpretable SFT. Observing common mid-layer change patterns across models suggests a shared mechanism for task-related knowledge acquisition.

Limitations

In this study, we conducted comprehensive fine-tuning experiments on pre-trained models in the 7-9B parameter class. Due to limited computational resources, we were unable to extend our verification to larger models (e.g., 70B, 175B, or MoE architectures). It is unclear whether the findings obtained in this research can be generalized to these larger models.

We emphasize the transparency and reproducibility of our analysis, and therefore, we trained our models using only open-access training datasets. Compared to existing publicly available English corpora, there are still limited multilingual instruction-tuning datasets. Consequently, our study used only English training datasets. A comprehensive investigation into cross-lingual knowledge transfer and its differing effects remains a subject for future work. We use about 10 popular training datasets for SFT, possibly limiting generality for highly specialized tasks or broader multilingual corpora.

While perplexity proved insightful, it can fluctuate based on tokenizer design and base training distributions, indicating a need for more nuanced measures.

Ethical Considerations

This work uses only publicly available and properly licensed datasets and base models. Their licenses permit research use and redistribution. All datasets and models were used in accordance with their intended research purposes, and our released models will maintain this intended use.

We did not collect any new data. While we did not manually inspect all samples, we acknowledge the possibility of residual personally identifiable or harmful content in the original datasets and rely on the original curators' filtering processes.

We will release over 1,000 fine-tuned models as part of this study. While we do not anticipate major risks, we acknowledge the potential for misuse—such as generating harmful or misleading content. To mitigate this, all released models will include a responsible use clause and detailed model cards describing limitations. We encourage responsible use.

We used AI tools to assist in writing training and evaluation scripts, and to support basic analysis tasks such as summarizing experimental results.

Acknowledgements

We used the "mdx: a platform for building data-empowered society" (Suzumura et al., 2022) for our experiments and analyses.

References

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yanpeng Li, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2025. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. 2024. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya

Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *Preprint*, arXiv:2307.08691.

Francesco Denti, Diego Doimo, Alessandro Laio, and Antonietta Mira. 2022. The generalized ratios intrinsic dimension estimator. *Scientific Reports*, 12(1):20005.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

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

Diego Doimo, Alessandro Serra, Alessio Ansuini, and Alberto Cazzaniga. 2024. The representation landscape of few-shot learning and fine-tuning in large language models. *Preprint*, arXiv:2409.03662.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 177–198, Bangkok, Thailand. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Maria d'Errico, Elena Facco, Alessandro Laio, and Alex Rodriguez. 2021. Automatic topography of high-dimensional data sets by non-parametric density peak clustering. *Information Sciences*, 560:476–492.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities. *Preprint*, arXiv:2404.17790.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Olmo: Accelerating the science of language models. *Preprint*.

Melody Y Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, et al. 2024. Deliberative alignment: Reasoning enables safer language models. *arXiv preprint arXiv:2412.16339*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021c. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Chan-Jan Hsu, Chang-Le Liu, Feng-Ting Liao, Po-Chun Hsu, Yi-Chang Chen, and Da-Shan Shiu. 2024. Breeze-7b technical report. *Preprint*, arXiv:2403.02712.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. 2024. O1 replication journey–part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *arXiv preprint arXiv:2411.16489*.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2:193–218.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Xisen Jin and Xiang Ren. 2024. Demystifying language model forgetting with low-rank example associations. *arXiv preprint arXiv:2406.14026*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019a. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023a. Cmmlu: Measuring massive multitask language understanding in chinese. *Preprint*, arXiv:2306.09212.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

LLM-jp, :, Akiko Aizawa, Eiji Aramaki, Bowen Chen, Fei Cheng, Hiroyuki Deguchi, Rintaro Enomoto, Kazuki Fujii, Kensuke Fukumoto, Takuya Fukushima, Namgi Han, Yuto Harada, Chikara Hashimoto, Tatsuya Hiraoka, Shohei Hisada, Sosuke Hosokawa, Lu Jie, Keisuke Kamata, Teruhito Kanazawa, Hiroki Kanezashi, Hiroshi Kataoka, Satoru Katsumata, Daisuke Kawahara, Seiya Kawano, Atsushi Keyaki, Keisuke Kiryu, Hirokazu Kiyomaru, Takashi Kodama, Takahiro Kubo, Yohei Kuga, Ryoma Kumon, Shuhei Kurita, Sadao Kurohashi, Conglong Li, Taiki Maekawa, Hiroshi Matsuda, Yusuke Miyao, Kentaro Mizuki, Sakae Mizuki, Yugo Murawaki, Akim Mousterou, Ryo Nakamura, Taishi Nakamura, Kouta Nakayama, Tomoka Nakazato, Takuro Niitsuma, Jiro Nishitoba, Yusuke Oda, Hayato Ogawa, Takumi Okamoto, Naoaki Okazaki, Yohei Oseki, Shintaro Ozaki, Koki Ryu, Rafal Rzepka, Keisuke Sakaguchi, Shota Sasaki, Satoshi Sekine, Kohei Suda, Saku Sugawara, Issa Sugiura, Hiroaki Sugiyama, Hisami Suzuki, Jun Suzuki, Toyotaro Suzumura, Kensuke Tachibana, Yu Takagi, Kyosuke Takami, Koichi Takeda, Masashi Takeshita, Masahiro Tanaka, Kenjiro Taura, Arseny Tolmachev, Nobuhiro Ueda, Zhen Wan, Shuntaro Yada, Sakiko Yahata, Yuya Yamamoto, Yusuke Yamauchi, Hitomi Yanaka, Rio Yokota, and Koichiro Yoshino. 2024. Llm-jp: A cross-organizational project for the research and development of fully open japanese llms. *Preprint*, arXiv:2407.03963.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Momose Oyama, Hiroaki Yamagiwa, Yusuke Takase, and Hidetoshi Shimodaira. 2025. Mapping 1,000+ language models via the log-likelihood vector. *Preprint*, arXiv:2502.16173.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual*

Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Laura Ruis, Maximilian Mozes, Juhan Bae, Siddhartha Rao Kamalakara, Dwarak Talupuru, Acyr Locatelli, Robert Kirk, Tim Rocktäschel, Edward Grefenstette, and Max Bartolo. 2024. Procedural knowledge in pretraining drives reasoning in large language models. *arXiv preprint arXiv:2411.12580*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Douglas Steinley. 2004. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386.

Toyotaro Suzumura, Akiyoshi Sugiki, Hiroyuki Takizawa, Akira Imakura, Hiroshi Nakamura, Kenjiro Taura, Tomohiro Kudoh, Toshihiro Hanawa, Yuji Sekiya, Hiroki Kobayashi, Yohei Kuga, Ryo Nakamura, Renhe Jiang, Junya Kawase, Masatoshi Hanai, Hiroshi Miyazaki, Tsutomu Ishizaki, Daisuke Shimotoku, Daisuke Miyamoto, Kento Aida, Atsuko Takefusa, Takashi Kurimoto, Koji Sasayama, Naoya Kitagawa, Ikki Fujiwara, Yusuke Tanimura, Takayuki Aoki, Toshio Endo, Satoshi Ohshima, Keiichiro Fukazawa, Susumu Date, and Toshihiro Uchibayashi. 2022. mdx: A cloud platform for supporting data science and cross-disciplinary research collaborations. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 1–7.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

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, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. 2024. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *arXiv preprint arXiv:2402.10176*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu,

Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2024. Magicoder: Empowering code generation with OSS-instruct. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 52632–52657. PMLR.

Chao-Chung Wu, Zhi Rui Tam, Chieh-Yen Lin, Hung yi Lee, and Yun-Nung Chen. 2025. Clear minds think alike: What makes llm fine-tuning robust? a study of token perplexity. *Preprint*, arXiv:2501.14315.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Xinlu Zhang, Zhiyu Zoey Chen, Xi Ye, Xianjun Yang, Lichang Chen, William Yang Wang, and Linda Ruth Petzold. 2024. Unveiling the impact of coding data instruction fine-tuning on large language models reasoning. *arXiv preprint arXiv:2405.20535*.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024a. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. *arXiv preprint arXiv:2402.04833*.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024b. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*.

Zheng Zhao, Yftah Ziser, and Shay B Cohen. 2024c. Layer by layer: Uncovering where multi-task learning happens in instruction-tuned large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15195–15214, Miami, Florida, USA. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Terry Yue Zhuo, Armel Zebaze, Nitchakarn Suppattarachai, Leandro von Werra, Harm de Vries, Qian Liu, and Niklas Muennighoff. 2024. Astraios: Parameter-efficient instruction tuning code large language models. *arXiv preprint arXiv:2401.00788*.

A Description of Base Models

OLMo (Groeneveld et al., 2024) is developed by *Allen Institute for AI*. An English-centric, 7B-parameter decoder model pre-trained on a carefully filtered mix of web pages, books, and code (totaling 2.5 trillion tokens). Flash-Attention 2 support was added in later versions, enabling fast, memory-efficient inference. Model-card results show competitive GSM8K and MMLU scores, rivaling some 10B-class models.

Llama3 (Dubey et al., 2024) is developed by *Meta AI*. An 8B English model trained on multi-trillion-token mixed-domain data with a byte-level BPE tokenizer and scaled RoPE. Safety alignment combines RLHF and rejection sampling. Delivers strong, well-rounded performance across reasoning, code, and chat benchmarks.

Mistral (Jiang et al., 2023) is developed by *Mistral AI*. An English 7B model whose pre-training corpus mixes web, academic text, and code. Grouped-query and sliding-window attention enable very long-sequence processing while retaining high speed. Matches or exceeds Llama-2-13B on many English tasks.

Gemma2 (Team et al., 2024) is developed by *Google DeepMind*. An English 9B model trained on a large quality-filtered corpus and enhanced with internal architectural refinements such as improved normalization and position encoding, building on modern Transformer techniques. Public reports

Language	Model Name (Params)	Repository (Hugging Face)	Context Length
English	OLMo (7B)	allenai/OLMo-7B-hf	2048
	Llama3 (8B)	meta-llama/Meta-Llama-3-8B	8192
	Mistral (7B)	mistralai/Mistral-7B-v0.1	32768
	Gemma2 (9B)	google/gemma-2-9b	8192
Chinese	Qwen2.5 (7B)	Qwen/Qwen2.5-7B	131072
	Chinese-Llama3 (8B)	hfl/llama-3-chinese-8b	8192
	Chinese-Mistral (7B)	itpossible/Chinese-Mistral-7B-v0.1	32768
	Yi1.5 (9B)	01-ai/Yi-1.5-9B	4096
Japanese	LLMjp-3 (7B)	llm-jp/llm-jp-3-7.2b	4096
	Llama3-Swallow (8B)	tokyotech-llm/Llama-3-Swallow-8B-v0.1	8192
	Swallow-Mistral (7B)	tokyotech-llm/Swallow-MS-7b-v0.1	4096
	Sarashina2 (7B)	sbintuitions/sarashina2-7b	4096

Table 1: Overview of the 12 base models employed for SFT experiments. The table summarizes their parameter sizes, primary training language, and maximum supported context lengths.

show it surpasses most open 7–13B baselines on language-understanding leaderboards.

Qwen2.5 (Yang et al., 2024) is developed by *Alibaba's Qwen team*. A Chinese–English bilingual 7B model further pre-trained on high-quality proprietary Chinese data. RoPE extrapolation enables extremely long inputs. The model card provides agent-style prompting templates and strong results on tool use and code generation.

Chinese-Llama3 (Cui et al., 2023) is developed by *Harbin NLP (HFL)*. An 8B Chinese model obtained by continual pre-training of Llama-3 on an extensive Chinese corpus with vocabulary augmentation. Significantly boosts Chinese QA and CMMLU scores over the original Llama-3.

Chinese-Mistral (Hsu et al., 2024) is developed by *itpossible*. A 7B Chinese variant of Mistral-v0.1, additionally trained on Chinese Wikipedia, news, and conversation data. Improves cross-lingual performance on Chinese benchmarks while preserving the original architecture.

Yi1.5 (AI et al., 2025) is developed by *01.AI*. A 9B multilingual model (Chinese + English focus) based on the original Yi model trained on 3.1 trillion tokens, with an additional 500 billion tokens used for continual pretraining, including substantial code and low-resource-language data. Shows solid zero-shot transfer to many Asian and European languages as well as code-related tasks.

LLMjp-3 (LLM-jp et al., 2024) is developed by *LLM-jp*. A 7.2B Japanese-centric model built from scratch on a 2.1 trillion token multilingual corpus, predominantly composed of Japanese web, book, and dialogue texts, along with a smaller portion of English and other languages. Public experiments indicate it surpasses Llama-2-13B on Japanese QA and summarization.

Llama3-Swallow (Fujii et al., 2024) is developed by *TokyoTech LLM Group*. An 8B Japanese model produced by continual pre-training of Llama-3-8B on large Japanese corpora plus vocabulary extension. Reports notable gains for Japanese NER and academic-paper summarization.

Swallow-Mistral (Fujii et al., 2024) is developed by *TokyoTech LLM Group*. A 7B Japanese follow-up to Mistral-7B with memory-footprint optimizations. Excels at Japanese dialogue and technical writing according to model-card evaluations.

Sarashina-2[3] is developed by *sbintuitions*. A 7B Japanese Llama derivative further trained on Japanese text and code. Distributed with LoRA adapters, making domain-specific fine-tuning straightforward.

B Description of Training Datasets

Alpaca (Taori et al., 2023) is a 52k-example English corpus obtained by filtering the original Stanford Alpaca to remove hallucinating prompts, merged instructions, empty outputs, and other defects. The resulting instruction/input/output triples serve as a cleaner general-purpose starting point for instruction tuning.

LIMA (Zhou et al., 2024) is a compact set of 1000 prompt–response pairs—750 mined from

[3]https://huggingface.co/sbintuitions/sarashina2-7b

Category	Dataset	Repository (Hugging Face)	Samples	Lengths
General	Alpaca	yahma/alpaca-cleaned	51,760	122.19
	LIMA	GAIR/lima	1,330	391.97
	Ultrachat	HuggingFaceH4/ultrachat_200k	773,913	397.45
Coding	CodeAlpaca	sahil2801/CodeAlpaca-20k	20,022	44.36
	Magicoder	ise-uiuc/Magicoder-Evol-Instruct-110K	111,183	300.04
Math	OpenMathInstruct	nvidia/OpenMathInstruct-1	6,078,712	140.43
	MathInstruct	TIGER-Lab/MathInstruct	262,039	125.36
Classic NLP	FLAN Knowledge	Open-Orca/FLAN	226,575	21.69
	FLAN Reasoning	Open-Orca/FLAN	92,770	41.74
	FLAN Comprehension	Open-Orca/FLAN	208,605	262.31

Table 2: Repository is the original source of the data used and Samples represents its total number of samples. Lengths indicates the average number of words in each data at 1k sample pre-processing.

Stack Exchange, wikiHow, and r/WritingPrompts plus 250 author-written items—selected for diversity and a consistent assistant style. It probes how well a strong language model can be aligned with minimal but high-quality supervision.

UltraChat (Ding et al., 2023) is a 774k multi-turn English dialogue corpus synthesized by two ChatGPT-Turbo agents. We use a reformatted version of the original release [4]. In our preprocessing pipeline, we extract only the initial user prompt and the first assistant reply as each training sample.

CodeAlpaca 20k (Chaudhary, 2023) is a collection of 20k English programming instructions generated with the Self-Instruct pipeline using text-davinci-003. About 40% of the samples include an input field, and the schema mirrors Alpaca but focuses exclusively on code generation and editing.[5]

Magicoder (Wei et al., 2024) contains 111k licence-clean code-centric instructions obtained by de-contaminating the Evol-CodeAlpaca corpus. Every example is a single-turn instruction→response pair, offering a larger companion to CodeAlpaca.

OpenMathInstruct (Toshniwal et al., 2024) is a 1.8M mathematics corpus whose step-by-step solutions were generated with Mixtral-8×7B and a Python interpreter, then automatically validated.

MathInstruct (Yue et al., 2023) aggregates 262k math-reasoning problems from 13 sources and augments them with both chain-of-thought and program-of-thought rationales, supplying lightweight yet generalizable coverage for mathematical fine-tuning.

[4] https://huggingface.co/datasets/stingning/ultrachat

[5] https://github.com/sahil280114/codealpaca

FLAN Collection (Wei et al., 2022) is the remix file flan2021_zsnoopt_submix_data.json. Specifically, it corresponds to the FLAN-2021 sub-mix and employs the zero-shot, no-options template variant (i.e., prompts contain only the instruction without in-context examples or candidate options). We follow the taxonomy of Dubey et al. (2024); Contributors (2023) and split the data into three thematic subsets.

FLAN Knowledge uses BoolQ (bool_q:1.0.0), NaturalQuestions (natural_questions_open:1.0.0), and TriviaQA (trivia_qa/rc:1.1.0). Samples whose output field is "none" are discarded.

FLAN Reasoning combines ARC-Easy (ai2_arc/ARC-Easy:1.0.0), ARC-Challenge (ai2_arc/ARC-Challenge:1.0.0), HellaSwag (hellaswag:1.1.0), WinoGrande (winogrande:1.1.0), and PIQA (piqa:1.0.0).

FLAN Comprehension contains QuAC (quac:1.0.0) and SQuAD v2.0 (squad/v2.0:3.0.0). Samples with an output of ‘none‘ are omitted.

C Preliminary Experiments

In our main experiments, we conduct SFT using various base models and diverse training datasets. To ensure valid and reliable results across different configurations, it is crucial to select appropriate hyperparameters. Therefore, we conducted preliminary experiments aimed at determining suitable hyperparameters.

These preliminary experiments were carried out under the following conditions. We employed the Llama3-8B model and utilized six different datasets, each comprising approximately 1,000 samples: Magicoder, LIMA, Code Alpaca, FLAN, Openmath, and Alpaca.

We examined several hyperparameter settings: learning rate = {2e-7, 1e-6, 2e-6, 1e-5, 2e-5, 1e-4}, batch size = {32, 64, 128, 256}, weight decay = {0, 0.1}, training method = {LoRA, full-parameter tuning}.

This combination of hyperparameters resulted in 96 unique experimental conditions. For each condition, we trained for 10 epochs and saved a checkpoint after every epoch (epochs 1 to 10). We treated these per-epoch checkpoints as distinct candidate models, yielding 960 models per dataset. Given that we utilized six datasets, the total number of trained models reached 5,760, consuming at least 23,040 GPU-hours.

For evaluation purposes, we utilized two benchmarks: MMLU and MT-bench, ensuring comprehensive performance assessment across diverse tasks.

D Description of Training Settings

This section summarizes the training configurations, computational cost, and other implementation details used in our supervised fine-tuning experiments.

Out of a total of 1,070 training runs, 1,059 models were successfully trained. Training failed in 11 cases, all related to out-of-memory (OOM) errors involving the Gemma model trained on the Magicoder dataset. Specifically, one failure occurred in a single-dataset setting, while the remaining ten failures arose during the All Dataset setting, where checkpoints were saved at every epoch (resulting in ten distinct training jobs).

We used separate hyperparameter settings for full-parameter fine-tuning and LoRA. Full-parameter fine-tuning was conducted with a learning rate of 1.0×10^{-5}, batch size of 32, weight decay of 0.0, and 10 training epochs. For LoRA, we used a learning rate of 2.0×10^{-6}, batch size of 128, weight decay of 0.0, and the same number of epochs. These values were determined based on preliminary grid-search experiments.

The computational time for fine-tuning on 1k samples varied depending on the model, batch size, and training method, but on average, each run took approximately 30 minutes. To accelerate training, we employed Flash Attention 2 (Dao, 2023) and DeepSpeed (Rasley et al., 2020) for all models. Training these 1,059 runs required at least 530 GPU-hours.

To investigate the impact of individual datasets, we conducted a dataset ablation study using three representative models: OLMo, Qwen, and LLM-jp. In this setting, we trained models on nine datasets at a time, excluding one dataset in each run (i.e., leave-one-out strategy). This allowed us to observe how the absence of specific datasets affected downstream performance. The ablation experiments were performed under the same conditions as regular 1k-sample training, using both full-parameter and LoRA-based fine-tuning.

As noted in Section 3.2, all datasets were preprocessed under consistent conditions. During training, we formatted all samples using a standardized instruction-response template:

```
###Question: {instruction}
###Answer: {response}
```

E Description of the Evaluation Dataset

This appendix provides an overview of the datasets used for evaluation. The test cases and evaluation settings follow the format provided by OpenCompass.

MATH (Hendrycks et al., 2021c) consists of 12,500 problems from high school math competitions. Each problem in MATH has a full step-by-step solution and models are tasked with generating tokens to construct the final answer.

GSM8K (Cobbe et al., 2021) is a dataset of 8,500 high quality linguistically diverse grade school math word problems created by human problem writers. Compared to MATH, the problems are easier and include basic knowledge questions, such as asking for the number of days in a week.

HumanEval (Chen et al., 2021) consists of 164 hand written programming problems. It assesses language comprehension, reasoning, algorithms, and simple mathematics.

MBPP (Austin et al., 2021) consists of 974 programming tasks, designed to be solvable by entry-level programmers. The problems range from basic computations to those requiring external mathematical knowledge.

BoolQ (Clark et al., 2019) is a question answering dataset for yes/no questions containing 15942 examples. The questions are real user queries—unprompted and written without knowing the answers—making them more inferential and challenging than synthetic datasets.

NaturalQuestion (Kwiatkowski et al., 2019a) consists of over 300,000 questions. This corpus features questions posed by actual users and chal-

lenges QA systems to read and understand a full Wikipedia article, which might or might not include the answer.

MMLU (Hendrycks et al., 2021b,a) consists of multiple-choice question-answer pairs divided into 57 subjects spanning STEM fields, the humanities, social sciences, and beyond. The questions vary in complexity, from elementary to expert-level, and assess both factual knowledge and reasoning skills.

MMLU-zh (Li et al., 2023a) contains 11,528 multiple-choice questions across 67 diverse subjects, including STEM, humanities, social sciences, and China-specific topics (e.g., Chinese law, traditional medicine, and ancient Chinese). The dataset is specifically constructed to reflect the linguistic and cultural nuances of Chinese, with many questions that are not easily translatable from English benchmarks like MMLU.

MMLU-jp We evaluated the models' Japanese generation ability using the Japanese-translated version of the multilingual MMLU test set [6]. While the content of the questions remains the same as the original MMLU, both the questions and answers are presented in Japanese.

TruthfulQA (Lin et al., 2022) comprises 817 questions that span 38 categories, including health, law, finance and politics. The questions are carefully crafted to trigger imitative falsehoods—answers that are commonly believed but factually incorrect.

MTBench (Zheng et al., 2023) is a benchmark designed to evaluate a model's instruction-following capabilities in a multi-turn dialogue format, consisting of 80 two-turn question sets. We conducted evaluations using the LLM-as-a-Judge framework, employing gpt-4o-2024-08-06 as the evaluator LLM.

AlpacaEval v2 (Li et al., 2023b) AlpacaEval v2 is an instruction-following benchmark consisting of 805 questions, created by integrating existing benchmarks and incorporating insights from real user interactions. We conducted pairwise evaluations by comparing the responses of our fine-tuned models against those of GPT-4-Turbo, and report the win rate as the evaluation metric. We used gpt-4o-2024-07-18 as the evaluator LLM.

All models evaluated in this experiment used the same prompt across all benchmarks. Therefore, it should be noted that the scores on downstream tasks may differ from those reported in technical reports, as the pre-trained models used a prompt template that differs from the one originally provided. Among the trained models, the following models failed to produce results for certain tasks:

- **Swallow-Mistral-7B**: All 41 trained models encountered out-of-memory errors across all tasks.
- **Mistral-7B**: The 20 LoRA-tuned models encountered out-of-memory errors on few-shot tasks.
- **Gemma2-9B**: For the models trained with all data using LoRA (from epoch 1 to epoch 4), responses that made evaluation with Alpaca Eval v2 impossible (extremely long, repetitive outputs) were generated. As a result, the win rates were recorded as `NaN`.

The completion of the evaluation process required approximately 16,700 GPU-hours, including the results from models and downstream tasks not described in the paper.

F Additional Results: Cross-lingual Transfer

We group models by their pre-training language (English, Chinese, Japanese) and compute pairwise Pearson correlations between MMLU-family scores across English, Chinese, and Japanese test sets (Figure A.1). All language pairs show strong positive correlations: substantial zero-shot transfer even though every SFT run used only English data. Evaluating SFT conducted in multiple languages remains an open avenue for future work.

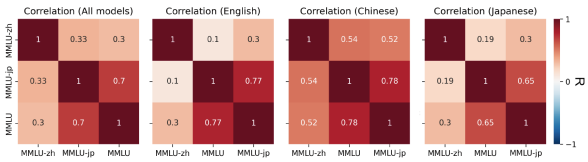


Figure A.1: Correlation coefficients of performance gain across models, focusing on MMLU, MMLU-jp and MMLU-zh, split by training languages for the models.

G Additional results: semantic similarity between training dataset and evaluation tasks

To calculate semantic similarity between training datasets and evaluation tasks, we computed BERTScore F1 between every training-dataset and evaluation-task pair using a pretrained BERT-base

[6] https://huggingface.co/datasets/openai/MMMLU

Category	Dataset	queries	metric
Math	MATH	5000	Exact Match Accuracy
	GSM8K	1319	Exact Match Accuracy
Coding	HumanEval	164	pass@1
	MBPP	257	pass@1
Knowledge	BoolQ	3270	Exact Match Accuracy
	NaturalQuestions	3610	Lenient Matching Accuracy
	TruthfulQA	817	BLEU Accuracy
Examination	MMLU	14042	Exact Match Accuracy
	MMLU-zh	11582	Exact Match Accuracy
	MMLU-jp	14042	Exact Match Accuracy
Instruction-following	MT Bench	160	Total Score
	Alpaca Eval v2	805	Win rate

Table 3: The list of downstream tasks used to evaluate the fine-tuned models is shown. All tasks are supported by the OpenCompass library, and the evaluation metrics are consistent with those used in OpenCompass.

model. Correlating these scores with the average SFT performance gains yielded only a small, non-significant positive relationship (Pearson's R = 0.112, $P > 0.05$). Hence, semantic closeness—as captured by BERTScore—offers little predictive value for fine-tuning benefit.

H Analyzing Hidden Representation Shift

We analyze the impact of fine-tuning on the hidden representations of LLMs. Previous studies have shown that task-specific information is encoded in the intermediate layers, and predictions are adjusted toward task-specific representations in the subsequent layers (Zhao et al., 2024c). We analyzed the global and local structural changes in the representation space by performing clustering analysis on the hidden representations of the training dataset.

H.1 Methods

LLMs and token representations analyzed.

We analyzed the hidden representations of a total of 110 models. This includes 10 base models that were primarily used in our evaluation. In addition to the base models, we also analyze the fine-tuned models that were trained on 1k-example subsets from 10 training datasets. This represents the simplest training setup, and analyzing models trained under different settings remains a topic for future work. To examine the effects of fine-tuning on both in-distribution and out-of-distribution data, we used a collection of 1k-example subsets from the training datasets, totaling N = 9974 examples. Following prior work (Doimo et al., 2024), we extracted the embedding of the final token in the prompt of each training example—formatted as `###Question: {instruction}`—from the outputs of all Transformer blocks. The final token is expected to encode contextual information accumulated from the preceding input, and its embedding is likely to vary across fine-tuned models. Finally, for each model, we obtain an L-layer, d-dimensional embedding space for N examples.

Representation quality measures.

To quantitatively evaluate the properties of text embeddings, we apply the Advanced Density Peaks (ADP) algorithm (d'Errico et al., 2021), a density-based clustering method. The algorithm first estimates the intrinsic dimensionality (ID) using the Gride (Denti et al., 2022). ID reflects how many parameters are needed to describe the data manifold. Based on this estimated dimensionality, it detects local density peaks using neighborhood-based criteria, and retains only statistically significant peaks via a t-test to form s clusters. To evaluate the properties of the estimated clusters, we use the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985; Steinley, 2004). ARI is computed by the following formula.

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{N_{ij}}{2} - \frac{\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}}{\binom{N}{2}}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \frac{\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}}{\binom{N}{2}}} \quad (1)$$

where a_i represents the number of samples contained in the cluster A_i ($1 \le i \le s$) and b_j represents the number of samples belonging to the dataset label B_j ($1 \le j \le 10$). ARI measures

how well the found clusters match the ground-truth labels, adjusting for chance grouping. It counts pairwise agreements between the clustering and true labels. 1.0 denotes perfect recovery of true classes by clusters, 0 indicates random alignment, and negative values imply worse-than-random clustering.

H.2 The global change in the hidden representation space

We observe two complementary effects of SFT on the model's embedding space. First, the number of identifiable clusters decreases A.2b: representations that were once scattered into many small groups collapse into a smaller set of semantically coherent modes, indicating that the model has learned to emphasize only those coarse distinctions that are most relevant for the downstream task. Second, the ID of each remaining cluster increases A.2a: within each merged mode, embeddings spread out along additional directions, reflecting the model's acquisition of subtler, task-specific features. Together, these trends suggest a trade-off in which fine-tuning simplifies the global structure of the representation (fewer clusters) while enriching its local expressiveness (higher ID), thereby balancing coarse category separation with finer-grained feature encoding.

A.2c shows that ARI fluctuates markedly across layers in every model, highlighting its sensitivity to representational changes. Fine-tuned variants generally exhibit lower ARI than their pretrained counterparts, indicating that clustering consistency does not directly predict generative performance. Moreover, because ARI here is computed over the full set of training-set embeddings, its overall trend may obscure differences between in-distribution and out-of-distribution samples. To disentangle these effects, we next perform an analysis of embedding–dataset correspondence.

H.3 The local change in the hidden representation space

As we show in the Figure A.2b, the number of clusters tends to be smallest in the final layer. We interpret this as the model forming semantically meaningful groupings in the embedding space at the final layer. Therefore, we examine the breakdown of hidden representations in the final layer of OLMo-7B. To evaluate the properties of the hidden representations for both in-distribution and out-of-distribution training datasets, we compute kNN consistency as defined by the following equation.

$$\text{kNN consistency}_c := \frac{1}{N_c} \sum_{i:y_i=c} \left[\frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} \mathbb{I}(y_j = y_i) \right]$$

Let y_i be the label of the i-th hidden representation in the training dataset, where $i \in 1, 2, \ldots, N$. For each embedding i, let $\mathcal{N}_k(i)$ denote the set of its k-nearest neighbors in the hidden representation space. To understand how well the local neighborhood of each data point matches its label, we calculate the kNN consistency for each label c. Specifically, for each data point whose label is c, we compute the proportion of its k nearest neighbors that have the same label. We then average this value over all points with label c: A label consistency of 1.0 means every point's neighbors are all the same class (perfect local purity), whereas lower values signify that points are often neighbored by different classes. In our experiments, we set $k = 300$.

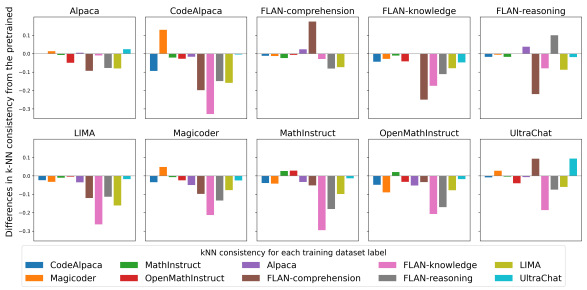
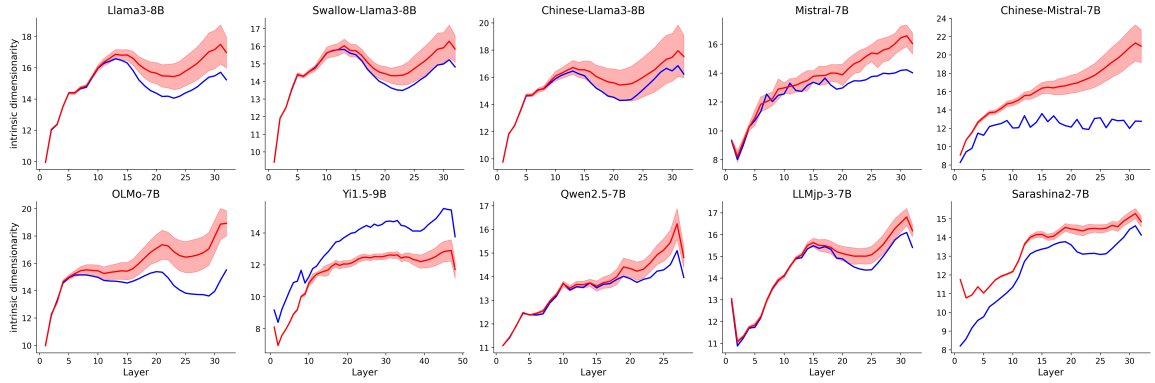


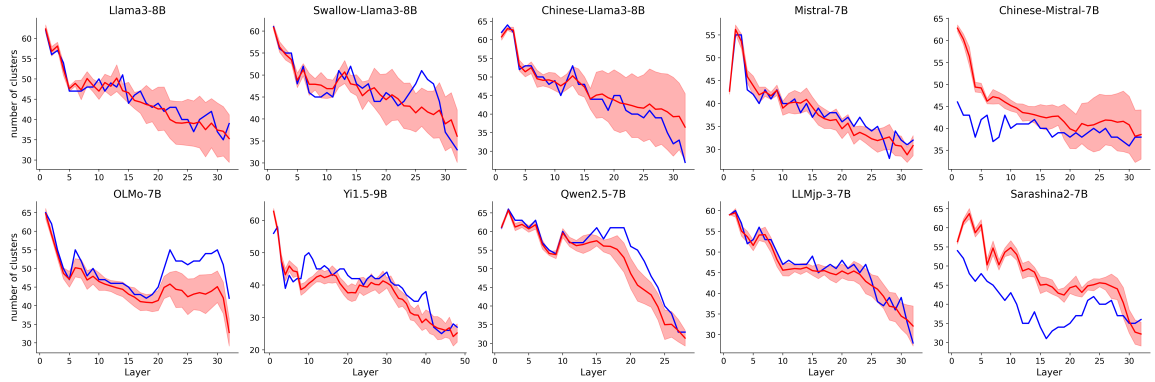
Figure A.3: **Differences in kNN consistency from the pretrained model for OLMo-7B.** This shows how the kNN consistency in the final layer changes for each dataset label when OLMo-7B is fine-tuned on 1k examples from each training dataset. For example, when OLMo-7B is fine-tuned on 1k examples from CodeAlpaca, it becomes better at embedding sentences from Magicoder more closely. On the other hand, for other datasets, the pretrained model demonstrates better separability.

Figure A.3 shows how kNN consistency of the embedding representations changes due to fine-tuning. Intuitively, the embeddings of datasets that belong to the same category as the training dataset—i.e., in-distribution—tend to become more tightly clustered, while embeddings of out-of-distribution datasets become harder to distinguish. In practice, when fine-tuned on FLAN-comprehension, FLAN-reasoning, or Magicoder, we observed a decrease in kNN consistency for datasets other than the one used for training. Similarly, when fine-tuned on MathInstruct, kNN con-

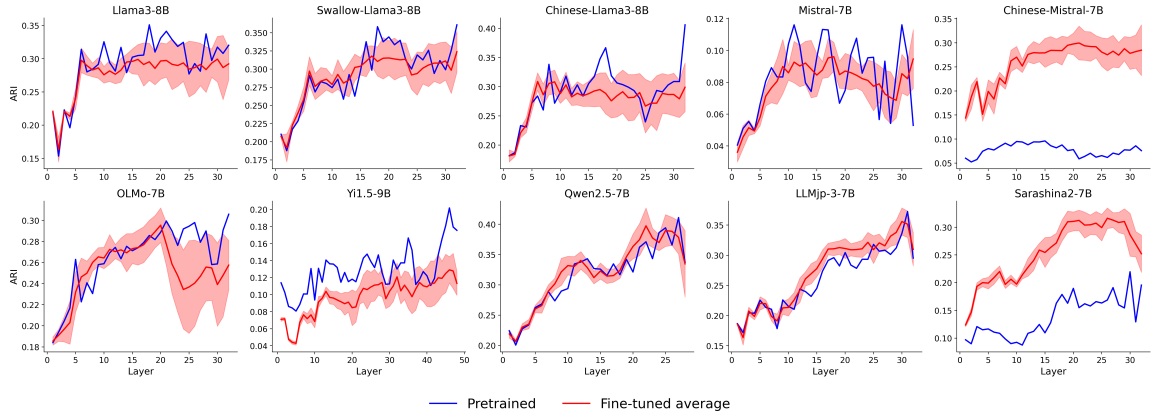
sistency decreased for all datasets except MathInstruct and OpenMathInstruct. This phenomenon is illustrated in Figure A.4 by projecting the embedding space into two dimensions using t-SNE. The pretrained model produces many small clusters, but it can still distinguish the labels of the training datasets. In contrast, the models fine-tuned on each training dataset show embeddings that are more tightly clustered together, making it more difficult to distinguish between the dataset labels. The mechanism of unlearning is likely caused by the model's embedding representations becoming less distinguishable for out-of-distribution datasets. Therefore, it will be beneficial to train on low-perplexity datasets that do not deviate too far from the base model's original distribution.



(a) Intrinsic Dimensionality across layers



(b) Number of clusters across layers



(c) Adjusted Rand Index across layers

Figure A.2: (a) **Intrinsic dimensions (ID) per layer of the sentence embeddings from the training dataset.** The blue lines represent the ID of each pretrained model. The red lines indicate the average ID of the models fine-tuned on a single training dataset for each pretrained model. Except for Yi1.5-9B, all models show an increase in ID due to fine-tuning, with the difference becoming apparent from the middle layers onward.
(b) **Number of clusters in the hidden representation space.** In many models, the final layer has the fewest number of clusters. Furthermore, fine-tuning reduces the number of clusters, showing a strong negative correlation with ID.
(c) **Adjusted Rand Index (ARI) of the density based clustering.** The values and trends vary significantly across models and layers. This suggests that the local structural arrangement of the training dataset is highly sensitive to the influence of the model and the dataset.

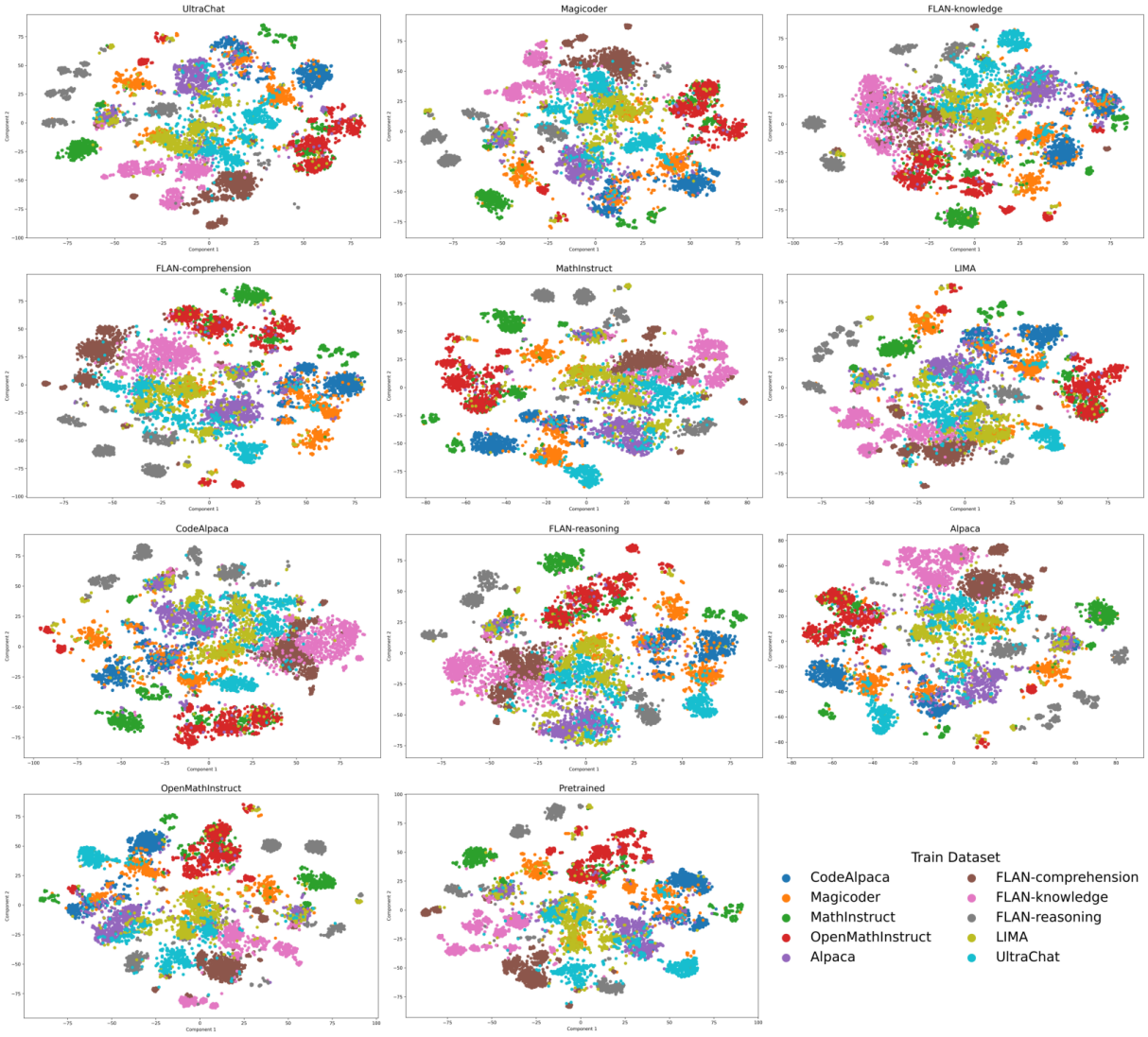


Figure A.4: **t-SNE visualization of OLMo-7B at last layer.** As an overall trend, the hidden representations of the trained (in-distribution) dataset become more tightly clustered, while the representations of the untrained (out-of-distribution) datasets show reduced discriminability and their distributions become more mixed with those of other datasets.