Comprehensive Study of Bilingual and Multi-category Instruction Pre-training

Takashi Kodama, Yusuke Oda
Research and Development Center for LLMs, National Institute of Informatics
{tkodama, odashi}@nii.ac.jp

Abstract

Instruction pre-training (IPT) has recently emerged as an effective intermediate stage between vanilla pre-training and post-training for large language models (LLMs). However, the optimal design of IPT corpora—such as the balance between raw and instruction-response data, languages, and task categories—remains unclear. We systematically study IPT corpus composition using a bilingual (English and Japanese) and multi-category (coding, general, math, and reasoning) instruction-response dataset. Through extensive IPT experiments across four base models, including both English-centric and bilingual LLMs, we find that: (1) more instruction-response data generally enhances model performance, particularly for models with large VPT budgets; (2) Japanese instruction data can improve English performance through cross-lingual transfer; and (3) the effectiveness of post-training varies across categories: coding performance is largely determined during IPT, while math and reasoning continue to improve during post-training.

1 Introduction

Recent advances in natural language processing have been largely driven by the emergence and rapid improvement of large language models (LLMs). The training of LLMs typically follows a two-stage paradigm (Kumar et al., 2025). In the first stage, vanilla pre-training (VPT) is conducted on large-scale raw corpora such as web text, enabling the model to acquire general world knowledge (Brown et al., 2020; Yang et al., 2025). This is followed by post-training, where the model is fine-tuned on relatively small-scale instruction-response data to align with user intent (Wei et al., 2022). Recently, instruction pre-training (Cheng et al., 2024) (IPT) has gained attention as an intermediate stage between pre-training and post-training. IPT involves continuous pre-training on large-scale instruction-response data, often spanning tens of billions of tokens, and has been shown to significantly improve downstream task performance.

While IPT presents a promising approach for building high-performing LLMs, practical methodologies for designing and composing effective IPT corpora remain underexplored. For instance, it is unclear how best to balance raw corpora and instruction-response data, or how the proportions of different languages (e.g., English and Japanese) and task categories (e.g., coding and math) within the instruction-response data impact model capability and on the efficiency of subsequent post-training. To date, few studies have systematically and quantitatively examined these factors, and practical guidelines for IPT data design are still lacking.

To address this problem, we conduct a comprehensive analysis of how the composition of IPT corpora affects downstream performance and the effectiveness of post-training. Specifically, we explore the following three research questions:

RQ1: What is the optimal composition of the IPT dataset, in terms of (i) the balance between raw corpora and instruction-response data, (ii) the language distribution, and (iii) the task category distribution?

RQ2: How do extreme imbalances in language or category distributions affect the final model capabilities?

RQ3: How do the language and category distributions in the IPT datasets impact the efficiency and effectiveness of post-training?

To answer these questions, we prepare an instruction-response dataset containing more than 90 billion tokens, covering multiple languages (English and Japanese) and categories (coding, general, math, reasoning), and conduct extensive instruction pre-training experiments across four base models, including both English-dominant and English-Japanese bilingual models.

Findings of the Association for Computational Linguistics: EACL 2026, pages 1323–1340
March 24-29, 2026 ©2026 Association for Computational Linguistics

Our empirical findings provide practical insights into IPT data construction and its impact on downstream model performance. The key observations are as follows:

- Increasing the proportion of instruction-response data generally boosts performance, especially for well pre-trained models. Regarding language balance, the impact varies depending on the language distribution of the VPT corpora; models pre-trained in a monolingual setting are more susceptible to performance degradation in other languages. For category balance, adding more reasoning data consistently enhances performance across most models.
- Cross-lingual transfer from Japanese to English is feasible, but English reasoning tasks still require direct English instruction-response data during IPT.
- Category performance improves when the corresponding category data is sufficiently included during IPT. However, the impact of subsequent post-training varies across categories: coding performance is mostly determined during IPT, while math and reasoning continue to benefit from post-training.

2 Related Work

2.1 Multi-stage Training

Large language models (LLMs) are typically developed through a multi-stage training process (Kumar et al., 2025). The first stage, known as pre-training, involves training the model on a massive text corpus comprising trillions of tokens collected from diverse web sources. This stage enables the model to acquire general world knowledge (Brown et al., 2020; Touvron et al., 2023b; Yang et al., 2025). In the subsequent stage, known as post-training, the pre-trained model is adapted to specific objectives using smaller and task-oriented datasets (Wei et al., 2022; Trung et al., 2024; Lobo et al., 2025)

Instruction pre-training refers to a stage where the model is trained on a medium-scale corpus (typically tens of billions of tokens) consisting of instruction-response pairs. This step aims to equip the model with the ability to follow user instructions. Cheng et al. (2024) demonstrated the effectiveness of instruction pre-training on downstream tasks under two scenarios: (i) training from scratch on a mixture of general web corpus and instruction-response pairs, and (ii) continuous pre-training for domain adaptation in specialized fields such as medicine and finance. However, their work focuses solely on English data and does not examine the effectiveness of bilingual instruction pre-training.

2.2 Multilingual Training

Multilingual support in LLMs has recently garnered significant attention, and multilingual training is now commonly applied at both the pre-training and post-training stages (Workshop et al., 2023; Grattafiori et al., 2024; Qwen et al., 2025; Yang et al., 2025). A wide range of studies have explored its benefits, including performance improvements and knowledge transfer across languages (Muennighoff et al., 2023; Fujii et al., 2024; Chen et al., 2024; Li et al., 2025; Lin et al., 2025). In the context of instruction-following, prior work has shown that introducing even a small number of target-language instructions—e.g., as few as 40 examples—into an English instruction dataset can significantly enhance performance in the target language (Shaham et al., 2024). However, these findings are based on relatively small-scale experiments with only tens of thousands of samples. In this study, we investigate the effectiveness of bilingual training at scale, using an instruction pre-training corpus comprising 130 million examples.

2.3 Multi-category Training

To equip LLMs with a broad range of capabilities, it is common to train them on datasets spanning multiple task categories such as mathematical reasoning (Hendrycks et al., 2021; Cobbe et al., 2021a; Gou et al., 2024; Li et al., 2024; Yang et al., 2024; Fujii et al., 2025) and coding (Chaudhary, 2023; Luo et al., 2024; Wang et al., 2024a; Wei et al., 2024; Hui et al., 2024). Dong et al. (2024) have conducted a comprehensive analysis of mathematical reasoning, code generation, and general human-aligning abilities in supervised fine-tuning (SFT), focusing on how the composition of category-specific training data affects performance. Building on this perspective, our study systematically manipulates the distribution ratios of instruction-response data across four task categories—coding, general, math, and reasoning—as well as across two languages, English and Japanese, and quantitatively evaluates the impact of these factors.

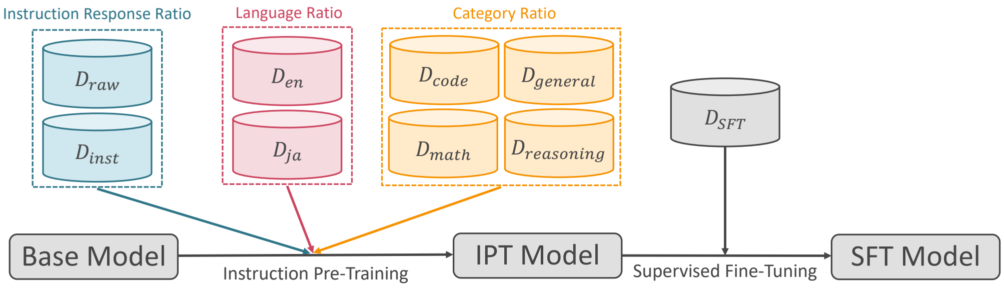

Figure 1: Training pipeline.

3 Experiment Settings

Figure 1 illustrates our overall training pipeline. We start from a pre-trained base model and perform instruction pre-training using datasets systematically varied along three axes: (1) the composition of raw corpora (D_{raw}) and instruction-response data (D_{inst}); (2) the language composition between English (D_{en}) and Japanese (D_{ja}) within the instruction-response data; and (3) the category composition among coding, general, math, and reasoning subsets ($D_{\text{code}}, D_{\text{general}}, D_{\text{math}}, D_{\text{reasoning}}$) within the instruction-response data.

Each instruction pre-trained model (IPT model) is then fine-tuned using the same SFT dataset (D_{SFT}) and training hyperparameters, yielding the final SFT model. This setup allows us to evaluate the impact of IPT data composition on downstream performance under fixed SFT conditions.

3.1 Base models

As base models, we employ three open-weight models: llm-jp-3-1.8b, Qwen2.5-1.5b, and LLaMA3.2-1b. All are in the 1B parameter class, making them suitable for extensive experiments within a feasible computational budget.

llm-jp-3-1.8b[1] (denoted as **llmjp3**) is trained on a 2.1T-token corpus consisting of English, Japanese, and code, with roughly equal amounts of English and Japanese. Qwen2.5-1.5b[2] (Qwen et al., 2025) (denoted as **qwen2.5**) is trained on a corpus of 18T tokens, while Llama3.2-1b[3] (Grattafiori et al., 2024) (denoted as **llama3.2**) is trained on up to 9T tokens. Although both models officially support Japanese, their VPT is assumed to be primarily focused on English (and Chinese in the case of qwen2.5), with relatively limited Japanese exposure. These differences in language exposure enable a comparative analysis between a bilingual model (llmjp3) and English-centric models (qwen2.5 and llama3.2).

However, the pre-training corpora for qwen2.5 and llama3.2 are not publicly disclosed, so their actual language composition remains uncertain. To address this, we pre-train a new LLM on an English-dominant corpus for this study. Specifically, we construct the English-dominant training corpus—91% of which is English—mainly from Fineweb (Penedo et al., 2024a) and train a 1.3B-parameter model compatible with the Llama architecture (Touvron et al., 2023a) on 15.6T tokens, using the same tokenizer as llmjp3.[4] We describe the training corpora and procedure in Appendix B. We refer to this model as **llama-inhouse** in this paper. Note that both llmjp3 and llama-inhouse are not explicitly instruction-pretrained[5], while qwen2.5 and llama3.2 may or may not include instruction-response data.

3.2 Datasets

For IPT, we use not only instruction-response data but also raw corpora such as Wikipedia. To ensure reproducibility and reusability, we only use datasets with permissive licenses for model training, and we publicly release the training corpus used in our experiments.[6]

3.2.1 Instruction-Response Data

We construct a new large-scale corpus consisting of approximately 130 million instruction-response

[1] https://huggingface.co/llm-jp/llm-jp-3-1.8b

[2] https://huggingface.co/Qwen/Qwen2.5-1.5B

[3] https://huggingface.co/meta-llama/Llama-3.2-1B

[4] Details of the model architecture are provided in Appendix A.

[5] It is worth noting that a small number of instruction-response pairs may have been included in their pre-training corpora.

[6] The training corpus is publicly available at https://gitlab.llm-jp.nii.ac.jp/datasets/llm-jp-corpus-midtraining-v1.

Language	Category	# of samples	# of tokens [B]
English	code	12,640,131	13.3 / 11.5 / 11.3
	general	58,385,816	19.4 / 18.1 / 17.7
	math	12,870,501	14.1 / 13.3 / 12.8
	reasoning	7,802,634	8.3 / 7.6 / 7.5
Japanese	code	8,695,799	10.8 / 11.5 / 11.7
	general	13,804,215	9.7 / 13.0 / 13.6
	math	6,710,082	8.2 / 9.0 / 9.0
	reasoning	9,072,005	8.6 / 10.9 / 11.6
Total		129,981,183	92.4 / 94.9 / 95.2

Table 1: Statistics of instruction-response dataset. The number of tokens is shown in the order of llmjp3 / qwen2.5 / llama3.2. “llama-inhouse” is omitted because it shares the same tokenizer as llmjp3.

samples, totaling around 90-95 billion tokens. Table 1 shows statistics broken down by language (English and Japanese) and category (code, general, math, and reasoning).[7] Because each model adopts a different tokenizer, we report the token counts separately for llmjp3, qwen2.5, and llama3.2. Following Cheng et al. (2024), we compute losses over both instructions and responses, and the token counts reflect their total. The dataset contains both single-turn and multi-turn dialogues, and each dialogue is counted as one sample.

Language labels are obtained from metadata such as readme files and dataset tags. Category labels are also based on available metadata. For datasets lacking sufficient category information, we automatically classify them using Qwen2.5-32B-Instruct[8]. We employ a prompt-based method for this classification, and the prompt details are provided in Appendix C.2. In this study, we classify instruction-response data that do not fall under coding, math, or reasoning into the general category.

3.2.2 Raw Corpora

We use the LLM-jp Corpus v3[9] , an open dataset comprising approximately 1.7 trillion tokens spanning English, Japanese, and code. We select this corpus because its balanced language composition makes it particularly suitable for bilingual pre-training in English–Japanese contexts. Detailed statistics and sampling ratios are presented in Appendix C.3.

Language	Category	# of samples	# of tokens [M]
English	code	90,748	68.0 / 57.8 / 57.2
	general	179,531	135.0 / 119.7 / 119.0
	math	10,980	11.2 / 10.9 / 10.6
	reasoning	2,712	3.0 / 2.8 / 2.7
Japanese	code	89,985	67.1 / 74.0 / 75.7
	general	549,870	60.8 / 79.8 / 85.0
	math	37,498	20.8 / 24.5 / 24.5
	reasoning	43,846	23.4 / 29.3 / 30.3
Total		1,005,313	389.3 / 398.6 / 405.0

Table 2: Statistics of SFT dataset. The number of tokens (response only) is shown in the order of llmjp3 / qwen2.5 / llama3.2. The language and category labels are assigned in the same manner as the instruction-response data.

3.3 Model Training

3.3.1 Instruction Pre-Training

We perform instruction pre-training on each pre-trained base model to investigate the effects of different data compositions. Across all models, the number of training steps is fixed at 25k. Data composition is adjusted on a token-count basis; for example, a 50%:50% English-Japanese composition indicates that the training corpus contains an equal number of English and Japanese tokens.

Inspired by the annealing strategy employed in Llama 3 (Grattafiori et al., 2024), we compute the average of model checkpoints over the final 10 steps of instruction pre-training, and use this averaged model as the IPT model. Detailed hyperparameter settings are provided in Appendix C.4.

3.3.2 Supervised Fine-Tuning

After the IPT phase, each IPT model is further fine-tuned using a common SFT setup across all models. The SFT dataset, largely adapted from `llm-jp-3-1.8b-instruct2`[10], contains approximately one million samples. As summarized in Table 2, it spans diverse categories, including coding, general, math, and reasoning, providing a comprehensive basis for evaluating how instruction pre-training impacts downstream performance.

To mitigate the impact of stochasticity during SFT, each IPT model is fine-tuned three times with different random seeds. All resulting models are included in the evaluation, and their results are aggregated to ensure stable and reliable comparisons.

[7] More detailed information is provided in Appendix C.1.

[8] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct

[9] https://gitlab.llm-jp.nii.ac.jp/datasets/llm-jp-corpus-v3

[10] https://huggingface.co/llm-jp/llm-jp-3-1.8b-instruct2

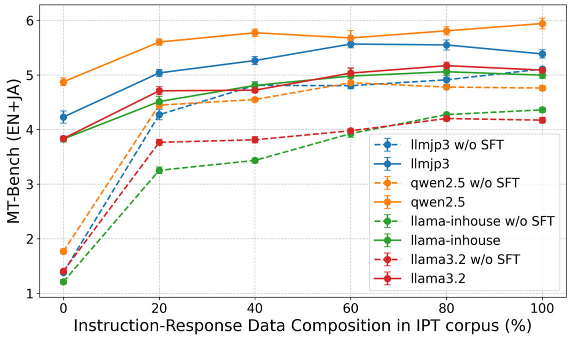

Figure 2: Impact of instruction-response data composition in IPT corpora. Error bars represent the standard deviation across evaluation models and random seeds.

3.4 Evaluation

For performance evaluation, we mainly use MT-Bench in both English (Zheng et al., 2023) and Japanese[11], with gpt-4o-2024-08-06 as the judge model. To reduce score variance, each model is evaluated three times. In addition to MT-Bench, we also assess coding and mathematical abilities using task-specific benchmarks. For coding, we employ HumanEval (Chen et al., 2021) (noted as "HumanEval (EN)") and JHumanEval[12] (noted as "HumanEval (JA)"), which is a Japanese translation of HumanEval. For mathematics, we use GSM8K (Cobbe et al., 2021b) (noted as "GSM8K (EN)") and the Japanese subset of MGSM (Shi et al., 2023) (noted as "GSM8K (JA)"). This setup enables evaluation of both coding and mathematical abilities in both English and Japanese. Details of the evaluation are provided in Appendix E.

We do not include harder mathematical reasoning benchmarks such as AIME in our evaluation. For 1B-scale models, these benchmarks are known to be extremely challenging and often yield scores close to the noise floor (Yang et al., 2025), making them unsuitable for discriminating the effects of different IPT data compositions.

4 Results

4.1 Optimal Data Composition

We first investigate the optimal proportion between raw corpora and instruction-response data. Based on the best-performing setting, we then explore how language and category distributions within the instruction-response data affect performance.

[11] https://github.com/Stability-AI/FastChat

[12] https://huggingface.co/datasets/kogi-jwu/jhumaneval

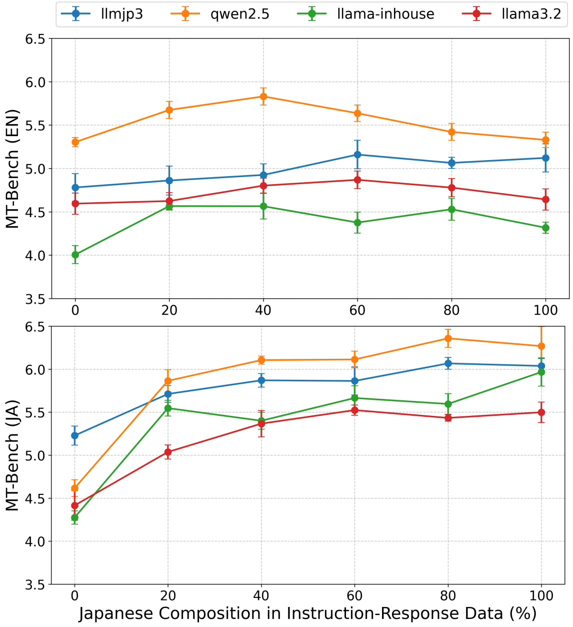

Figure 3: Impact of language data composition in instruction-response data.

4.1.1 Instruction-Response Data

Figure 2 shows the average MT-Bench scores in English and Japanese under varying proportions of instruction-response data in the IPT corpus. Regardless of whether SFT is applied, performance consistently improves as the share of instruction-response data increases. The notably low 0% score of the model without SFT indicates that models trained solely on raw corpora fail to acquire the conversational and instruction-following abilities required for MT-Bench. The most substantial gains occur between 0% and 20%, after which performance saturates around 80%-100%. These results suggest that allocating approximately 80% of the IPT corpus to instruction-response data is sufficient to achieve near-optimal performance.

4.1.2 Languages

Figure 3 shows MT-Bench scores in English and Japanese as we increase the proportion of Japanese instruction-response data. In the Japanese MT-Bench, all models consistently improve with more Japanese instruction-response data.

In contrast, the English MT-Bench results reveal a more complex pattern. While qwen2.5, llama-inhouse, and llama3.2 show moderate gains at intermediate Japanese ratios (20-60%), their performance deteriorates as the Japanese data ratio approaches 100%. These moderate gains may partially reflect cross-lingual knowledge transfer

Data Ratio (%)	llmjp3		qwen2.5		llama-inhouse		llama3.2	
	EN	JA	EN	JA	EN	JA	EN	JA
25/25/25/25	5.07 (0.08)	5.73 (0.06)	5.50 (0.13)	6.14 (0.11)	4.60 (0.11)	5.43 (0.14)	4.72 (0.13)	5.29 (0.06)
0/33/33/33	5.01 (0.07)	5.88 (0.15)	5.53 (0.14)	6.04 (0.08)	4.60 (0.08)	5.35 (0.06)	4.88 (0.09)	5.47 (0.09)
50/17/17/17	4.95 (0.17)	5.76 (0.15)	5.40 (0.09)	6.17 (0.11)	**4.70 (0.12)**	5.50 (0.09)	4.68 (0.08)	5.49 (0.15)
100/0/0/0	4.77 (0.06)	5.42 (0.09)	4.56 (0.09)	5.46 (0.09)	4.04 (0.11)	4.93 (0.11)	4.32 (0.07)	5.02 (0.10)
33/0/33/33	5.02 (0.12)	5.78 (0.12)	5.45 (0.05)	5.98 (0.12)	4.53 (0.12)	5.54 (0.19)	**4.98 (0.09)**	5.48 (0.08)
17/50/17/17	4.90 (0.09)	5.75 (0.13)	5.59 (0.08)	6.24 (0.10)	4.53 (0.19)	5.45 (0.15)	4.79 (0.12)	5.28 (0.10)
0/100/0/0	4.34 (0.07)	5.19 (0.10)	4.89 (0.15)	5.55 (0.08)	3.79 (0.10)	4.74 (0.09)	4.12 (0.09)	4.71 (0.13)
33/33/0/33	4.94 (0.10)	5.77 (0.11)	**5.62 (0.07)**	6.14 (0.11)	4.61 (0.09)	5.30 (0.13)	4.91 (0.15)	5.25 (0.07)
17/17/50/17	5.07 (0.18)	5.81 (0.10)	5.46 (0.10)	6.07 (0.06)	4.55 (0.09)	5.54 (0.05)	4.70 (0.10)	**5.51 (0.09)**
0/0/100/0	4.61 (0.20)	5.57 (0.08)	4.29 (0.12)	5.36 (0.14)	3.97 (0.13)	4.92 (0.12)	4.45 (0.12)	5.14 (0.07)
33/33/33/0	4.86 (0.10)	**5.92 (0.12)**	5.16 (0.07)	6.00 (0.09)	4.49 (0.10)	5.21 (0.08)	4.61 (0.13)	5.19 (0.11)
17/17/17/50	**5.12 (0.13)**	5.90 (0.12)	5.54 (0.12)	**6.25 (0.06)**	4.51 (0.07)	**5.75 (0.08)**	**4.98 (0.10)**	5.42 (0.08)
0/0/0/100	4.90 (0.09)	5.79 (0.07)	5.19 (0.13)	5.74 (0.14)	4.49 (0.09)	5.09 (0.06)	4.82 (0.12)	5.37 (0.12)

Table 3: Impact of category data composition in instruction-response data. Data ratios are shown in the order of code/general/math/reasoning. EN and JA denote the average English and Japanese MT-Bench scores, respectively. The numbers in parentheses indicate standard deviations. The top two results across different data ratios are marked with **bold** and underlined.

	llmjp3		qwen2.5		llama-inhouse		llama3.2	
	en only	ja only	en only	ja only	en only	ja only	en only	ja only
MT-Bench (EN, coding)	4.58$_{-0.35}$	4.74$_{-0.19}$	5.39$_{+0.08}$	4.75$_{-0.56}$	4.21$_{-0.26}$	3.95$_{-0.52}$	4.23$_{-0.37}$	4.58$_{-0.02}$
MT-Bench (EN, general)	5.39$_{-0.28}$	6.19$_{+0.15}$	5.53$_{-0.44}$	5.71$_{-0.21}$	4.51$_{-0.85}$	5.33$_{-0.24}$	4.94$_{-0.44}$	5.50$_{-0.03}$
MT-Bench (EN, math)	3.92$_{-0.42}$	4.10$_{-0.24}$	4.87$_{-0.16}$	4.77$_{-0.26}$	3.08$_{-0.68}$	3.84$_{+0.08}$	4.79$_{+0.53}$	3.46$_{-0.80}$
MT-Bench (EN, reasoning)	3.61$_{-1.00}$	4.08$_{-0.53}$	3.93$_{-0.71}$	3.61$_{-1.03}$	3.74$_{-0.09}$	2.94$_{-0.89}$	3.21$_{-0.52}$	2.69$_{-1.04}$
HumanEval (EN)	0.27$_{-0.03}$	0.27$_{-0.03}$	0.31$_{-0.02}$	0.31$_{-0.02}$	0.24$_{-0.01}$	0.20$_{-0.05}$	0.27$_{-0.02}$	0.26$_{-0.03}$
GSM8K (EN)	0.31$_{-0.04}$	0.29$_{-0.06}$	0.35$_{-0.05}$	0.46$_{+0.06}$	0.16$_{-0.05}$	0.22$_{+0.01}$	0.24$_{-0.07}$	0.29$_{-0.02}$
MT-Bench (JA, coding)	4.35$_{-0.51}$	4.93$_{+0.07}$	4.78$_{-0.77}$	5.29$_{-0.26}$	4.00$_{-0.49}$	4.70$_{+0.21}$	4.32$_{-0.08}$	4.29$_{-0.11}$
MT-Bench (JA, general)	5.66$_{-0.94}$	6.68$_{+0.17}$	4.52$_{-1.87}$	6.73$_{+0.02}$	4.74$_{-1.49}$	7.16$_{+0.55}$	4.79$_{-1.39}$	6.23$_{+0.05}$
MT-Bench (JA, math)	5.14$_{+0.13}$	5.06$_{+0.05}$	4.98$_{-0.88}$	6.05$_{+0.19}$	3.98$_{-0.89}$	5.35$_{+0.48}$	4.13$_{-0.62}$	4.39$_{-0.36}$
MT-Bench (JA, reasoning)	4.21$_{-0.68}$	4.44$_{-0.45}$	3.87$_{-0.94}$	5.13$_{+0.32}$	3.31$_{-0.66}$	4.51$_{+0.54}$	3.33$_{-0.10}$	4.41$_{+0.98}$
HumanEval (JA)	0.24$_{+0.00}$	0.24$_{+0.00}$	0.26$_{-0.05}$	0.29$_{-0.02}$	0.21$_{+0.00}$	0.21$_{+0.00}$	0.23$_{+0.00}$	0.25$_{+0.02}$
GSM8K (JA)	0.29$_{-0.03}$	0.33$_{+0.01}$	0.34$_{-0.18}$	0.50$_{-0.02}$	0.18$_{-0.09}$	0.29$_{+0.02}$	0.20$_{-0.15}$	0.35$_{+0.00}$

Table 4: Impact of imbalanced language data composition in the instruction-response data. We compare two settings: English-only and Japanese-only instruction-response data. The subscript in each cell indicates the absolute difference from the baseline score, which was obtained by training on an instruction-response dataset containing approximately equal amounts of English and Japanese data. Improvements over the baseline are shown in green, while degradations are shown in red. For brevity, we group "extraction", "humanities", "roleplay", and "writing" in MT-Bench into the "general" category. The complete results are provided in Table 12.

from Japanese data. However, as the imbalance increases, overexposure to Japanese, which was underrepresented during VPT, may introduce interference (Stap et al., 2023), thereby impairing the model's ability to generalize to its dominant language. Similar interference effects have been reported in prior studies on multilingual continuous pre-training (Fujii et al., 2024).

Interestingly, unlike other models, llmjp3 does not exhibit a noticeable degradation in English performance, even when trained with 100% Japanese instruction-response data. This robustness can be attributed to its bilingual pre-training, during which the model was exposed to roughly equal amounts of English and Japanese tokens. As a result, llmjp3 appears more resilient to language distribution shifts during instruction pre-training.

These findings collectively suggest a cross-lingual trade-off between the two languages: increasing the amount of instruction-response data in a single language can enhance performance in that language but may harm performance in others, especially when the model lacks prior exposure. Models like llmjp3, which are pre-trained on balanced bilingual corpora, are more robust to such shifts and offer more stable performance across languages.

4.1.3 Categories

Table 3 reports the performance of models trained with varying proportions of instruction-response

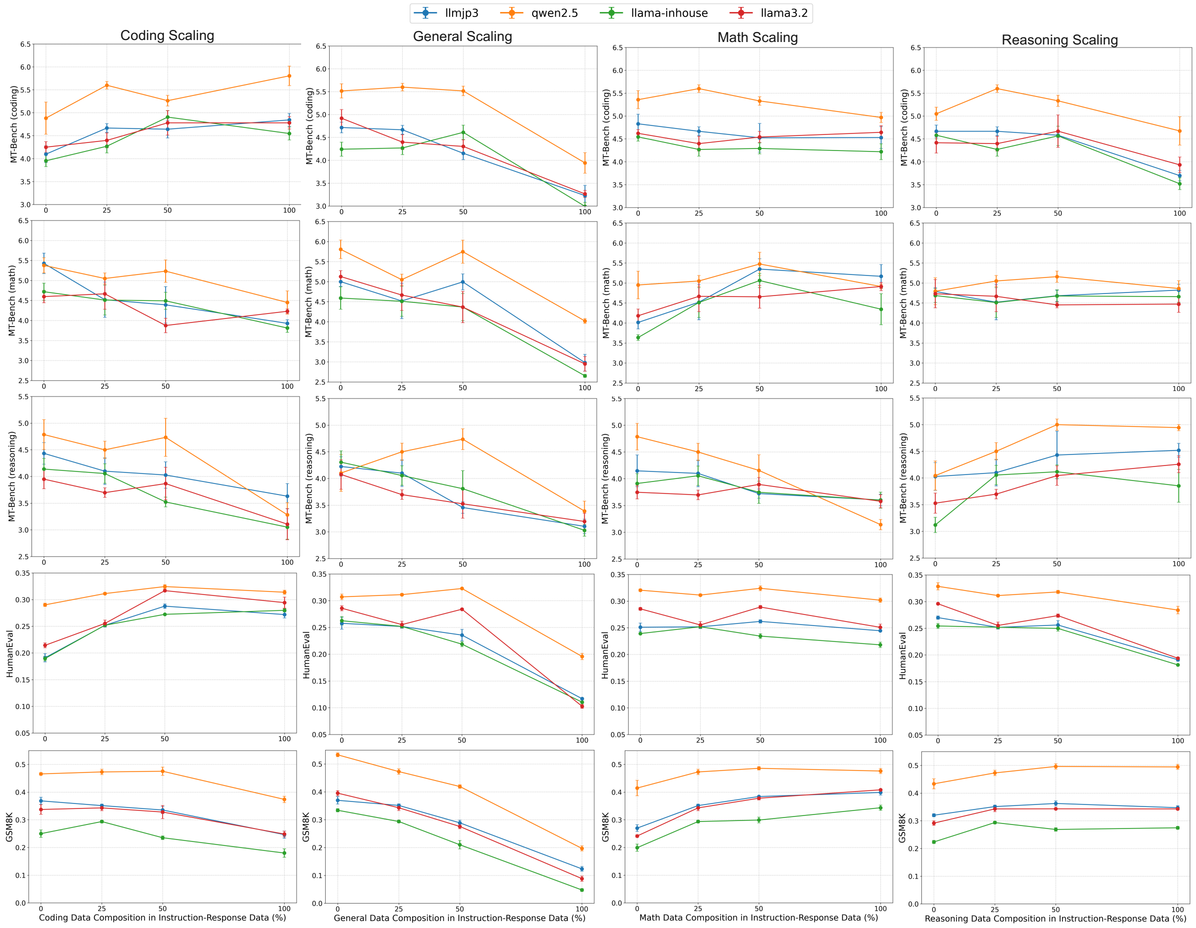

Figure 4: Impact of category composition in the instruction-response data on performance in coding, math, and reasoning tasks.

data across four categories: code, general, math, and reasoning. Although no single configuration consistently outperformed all others across every model, the 17/17/17/50 setting—where reasoning data accounts for half of the instruction-response data—yielded stronger overall results than the balanced configuration (25/25/25/25). To further validate these observations, we conducted t-tests comparing the two configurations. Across eight evaluation settings (four models × two languages in MT-Bench), 17/17/17/50 outperformed 25/25/25/25 in seven cases, and four of these differences were statistically significant. These findings suggest that increasing the share of reasoning data within the instruction-response data is beneficial to overall model performance.

4.2 Imbalanced Instruction-Response Data

We closely examine the performance of models trained with instruction-response data that is highly skewed in either language or category distribution.

4.2.1 Languages

Table 4 presents category-wise average scores on the English and Japanese MT-Bench under two extreme language configurations: 100% English and 100% Japanese instruction-response data.

For the English benchmarks (upper part of the table), models generally showed lower performance compared to the baseline with a balanced English-Japanese mix. Interestingly, even llama-inhouse, which was primarily pre-trained on English, showed a performance drop in the English-only setting. This suggests that Japanese instruction-response data contributes positively to English tasks, likely through cross-lingual transfer. Conversely, under the Japanese-only setting, the most significant decline was observed in the reasoning category, indicating that English instruction-response data is essential for maintaining performance on English reasoning tasks.

For the Japanese benchmarks (lower part of the table), performance tended to improve with Japanese-only instruction-response data. Cod-

Data Ratio (%)	MTBench (coding)		MTBench (math)		MTBench (reasoning)		HumanEval		GSM8k	
	EN	JA	EN	JA	EN	JA	EN	JA	EN	JA
Baseline	$2.82_{+134.6}$	$3.19_{+163.0}$	$2.25_{+103.5}$	$3.06_{+165.9}$	$3.39_{+161.0}$	$3.32_{+118.0}$	$0.13_{+1131.4}$	$0.14_{+939.3}$	$0.08_{+188.2}$	$0.12_{+385.6}$
25/25/25/25	$4.65_{+17.3}$	$4.81_{+6.8}$	$4.29_{+24.4}$	$5.09_{+16.2}$	$4.01_{+25.6}$	$4.16_{+36.4}$	$0.29_{+8.3}$	$0.24_{+9.6}$	$0.37_{+0.9}$	$0.36_{+7.9}$
0/33/33/33	$4.25_{+20.1}$	$4.35_{+20.9}$	$4.46_{+18.8}$	$5.60_{+22.6}$	$4.18_{+54.8}$	$4.47_{+36.2}$	$0.24_{+13.9}$	$0.20_{+7.2}$	$0.31_{+0.5}$	$0.40_{+9.3}$
50/17/17/17	$4.82_{+15.1}$	$4.97_{+12.6}$	$4.02_{+28.4}$	$4.97_{+23.4}$	$3.72_{+45.2}$	$4.36_{+29.7}$	$0.31_{+15.1}$	$0.29_{+15.4}$	$0.32_{-1.6}$	$0.36_{+4.8}$
100/0/0/0	$4.97_{+16.2}$	$5.02_{+10.0}$	$3.84_{+26.4}$	$4.37_{+17.3}$	$2.72_{+1.0}$	$3.81_{+34.4}$	$0.30_{+16.0}$	$0.28_{+11.7}$	$0.26_{+19.2}$	$0.27_{+20.8}$
33/0/33/33	$4.88_{+22.0}$	$4.81_{+12.5}$	$4.97_{+48.2}$	$5.29_{+15.7}$	$3.63_{+35.6}$	$4.72_{+54.4}$	$0.29_{+11.7}$	$0.26_{+9.5}$	$0.41_{+2.4}$	$0.41_{+4.5}$
17/50/17/17	$4.66_{+12.4}$	$4.63_{+15.1}$	$4.49_{+46.1}$	$5.25_{+20.1}$	$3.47_{+23.6}$	$4.29_{+29.6}$	$0.28_{+11.8}$	$0.24_{+10.2}$	$0.27_{+17.3}$	$0.33_{+5.0}$
0/100/0/0	$3.17_{+32.8}$	$3.54_{+27.7}$	$2.98_{+28.2}$	$3.32_{+26.3}$	$3.03_{+6.4}$	$3.32_{+3.4}$	$0.14_{+89.7}$	$0.12_{+104.8}$	$0.07_{+438.9}$	$0.16_{+38.9}$
33/33/0/33	$4.82_{+18.8}$	$4.86_{+13.2}$	$3.86_{+13.1}$	$4.53_{+5.6}$	$4.01_{+20.3}$	$4.29_{+35.1}$	$0.29_{+13.6}$	$0.26_{+5.0}$	$0.24_{+6.0}$	$0.33_{+8.0}$
17/17/50/17	$4.64_{+12.3}$	$4.70_{+10.3}$	$4.58_{+39.7}$	$5.69_{+18.8}$	$3.73_{+33.1}$	$4.04_{+31.9}$	$0.30_{+15.9}$	$0.25_{+12.3}$	$0.38_{-0.5}$	$0.40_{+4.8}$
0/0/100/0	$4.43_{+20.1}$	$4.75_{+14.3}$	$4.48_{+15.8}$	$5.19_{+13.2}$	$3.07_{+19.6}$	$3.89_{+31.8}$	$0.27_{+4.6}$	$0.23_{+10.0}$	$0.42_{-0.8}$	$0.39_{+3.5}$
33/33/33/0	$4.52_{+3.4}$	$4.84_{+12.3}$	$4.16_{+32.7}$	$5.34_{+27.8}$	$3.42_{+25.5}$	$3.94_{+22.3}$	$0.30_{+12.3}$	$0.28_{+11.4}$	$0.29_{-2.3}$	$0.35_{+8.7}$
17/17/17/50	$4.69_{+15.4}$	$4.88_{+16.6}$	$4.38_{+33.9}$	$5.10_{+13.9}$	$4.17_{+41.5}$	$4.63_{+33.2}$	$0.30_{+22.1}$	$0.25_{+9.0}$	$0.35_{-6.0}$	$0.39_{+5.8}$
0/0/0/100	$3.91_{+29.5}$	$4.00_{+20.6}$	$4.34_{+27.9}$	$5.07_{+32.2}$	$4.29_{+21.4}$	$4.50_{+29.6}$	$0.22_{+14.2}$	$0.20_{+18.4}$	$0.35_{+4.2}$	$0.37_{+6.9}$

Table 5: Benchmark scores on coding, math, and reasoning tasks, along with relative improvements during the SFT stage. Each model is instruction-pretrained with different category data composition. Data ratios are shown in the order of code/general/math/reasoning. The "Baseline" model is trained using only raw corpora in IPT. "EN" and "JA" indicate evaluation results in English and Japanese. The values shown in each cell denote the scores after SFT. Subscripts denote the relative improvement rate (%) from before to after SFT, computed as $(\text{after} - \text{before})/\text{before} \times 100$. Score improvements exceeding 20% are highlighted in green.

ing showed only minimal improvement, implying weaker dependence on language alignment. Models with limited Japanese exposure during VPT, such as llama-inhouse, appeared to benefit more substantially from Japanese instruction-response data. Conversely, English-only training tended to degrade Japanese performance across all models, likely due to insufficient adaptation to Japanese linguistic structures and formatting conventions.

4.2.2 Categories

Figure 4 presents the average English and Japanese benchmark scores for the coding, math, and reasoning categories under different proportions (0%, 25%, 50%, and 100%) of instruction-response data specific to each category. Across all three categories, increasing the proportion of category-specific data from 0% to 50% consistently improves performance in the corresponding category. However, further increasing the proportion to 100% generally yields no additional gains, and in some cases even leads to performance degradation, suggesting that excessive specialization may impair the model's ability to generalize.

4.3 Relationship between IPT and SFT

Table 5 reports benchmark scores for coding, math, and reasoning tasks, together with improvements from before to after SFT. Across all three categories, we observe a consistent trend: final performance tends to improve when the corresponding category is sufficiently included during the IPT phase. However, the magnitude of SFT gains varies by category. Results from MT-Bench and HumanEval indicate that coding performance is largely established during IPT, leaving limited room for further gains through SFT. In contrast, math and reasoning continue to benefit considerably from SFT, even when a substantial amount of relevant data has already been included during IPT.[13] However, GSM8K shows little improvement from SFT, unlike MT-Bench (math). This likely reflects that the instruction-response data already contains the GSM8K training split (see Table 17), enabling the model to sufficiently learn GSM8K tasks during IPT alone.

Configurations lacking target category data during IPT—such as 33/33/0/33 for math, and 100/0/0/0 or 0/100/0/0 for reasoning—consistently yield lower final scores. Moreover, these setups show only limited SFT improvements (less than 10%), suggesting that SFT alone cannot fully compensate for insufficient IPT exposure.

5 Conclusion

In this study, we systematically examined how the composition of IPT corpora affects the downstream performance and the effectiveness of subsequent SFT. Our results indicate that increasing the proportion of instruction-response data generally benefits model performance, particularly for

[13] We verified that our conclusions remain consistent even when using different SFT data compositions, as detailed in Appendix F.

models with extensive VPT budgets. We also underscore the importance of language balance: although cross-lingual transfer from Japanese to English is feasible, certain tasks, such as English reasoning, require direct exposure to English instruction-response data. Finally, we observe that coding performance is largely determined during IPT and shows minimal gains from subsequent SFT, whereas math and reasoning continue to benefit significantly from SFT even when these categories are well represented during the IPT phase.

Limitations

While our study provides valuable insights into the effects of IPT corpus composition, it is subject to several limitations.

First, due to the substantial computational cost of our large-scale instruction pre-training, which involves 100 IPT runs (25 configurations across 4 models), our experiments are limited to 1B-scale base models. Nevertheless, prior studies on scaling laws provide useful context for interpreting our results.

Kaplan et al. (2020) demonstrate that model performance follows power-law relationships with model size, dataset size, and compute budget, implying that larger models are more sample-efficient and achieve lower loss with fewer training tokens. Similarly, Hoffmann et al. (2022) (the Chinchilla scaling law) show that, under a fixed compute budget, the optimal balance is achieved when model parameters and training tokens scale proportionally. Additionally, Magnusson et al. (2025) propose *DataDecide*, which investigates how small-scale experiments on data composition can predict large-scale performance in vanilla pre-training. They train models up to 1B parameters across various data compositions and find that rankings derived from small models (e.g., $<$150M parameters) can predict the best data configurations at the 1B scale with approximately 80% accuracy.

Taken together, these findings suggest that the qualitative trends observed in our study—such as the effects of data composition and category balance—are likely to generalize to larger models, even though the absolute magnitude of improvements may vary as models become more data-efficient.

Second, our evaluation is mainly based on MT-Bench, a widely used benchmark that scores responses using GPT-4o. This introduces potential variability and misalignment with human judgment. To reduce this variability, we evaluate each data composition configuration nine times—three SFT runs with different seeds, each evaluated three times—but some degree of variance remains, and the results should be interpreted accordingly.

Third, to support large-scale IPT, we constructed a 90B-token instruction-response dataset, primarily sourced from existing corpora. Due to its massive scale, we were unable to conduct a comprehensive assessment of its quality or diversity, which may affect the generalizability of our conclusions.

Future work should explore whether these trends extend to larger models and further investigate how the quality and diversity of instruction-response data influence the effectiveness of IPT.

Acknowledgments

We used ABCI 3.0 provided by AIST and AIST Solutions with support from “ABCI 3.0 Development Acceleration Use.” We additionally used the “mdx: a platform for building data-empowered society.”

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, and 3 others. 2025. Smollm2: When smol goes big – data-centric training of a small language model. Preprint, arXiv:2502.02737.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20, Red Hook, NY, USA. Curran Associates Inc.

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

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, and 39 others.

2021. Evaluating large language models trained on code.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Andrey Kutuzov, Barry Haddow, and Kenneth Heafield. 2024. Monolingual or multilingual instruction tuning: Which makes a better alpaca. In Findings of the Association for Computational Linguistics: EACL 2024, pages 1347–1356, St. Julian's, Malta. Association for Computational Linguistics.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pretraining: Language models are supervised multitask learners. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2529–2550, Miami, Florida, USA. Association for Computational Linguistics.

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. 2021a. Training verifiers to solve math word problems. Preprint, arXiv:2110.14168.

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

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.

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.

Kazuki Fujii, Yukito Tajima, Sakae Mizuki, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Masanari Ohi, Masaki Kawamura, Taishi Nakamura, Takumi Okamoto, Shigeki Ishida, Kakeru Hattori, Youmi Ma, Hiroya Takamura, Rio Yokota, and Naoaki Okazaki. 2025. Rewriting pre-training data boosts llm performance in math and code. Preprint, arXiv:2505.02881.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In The Twelfth International Conference on Learning Representations.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. Preprint, arXiv:2407.21783.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, and 5 others. 2024. Qwen2.5-coder technical report. Preprint, arXiv:2409.12186.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Denis Kocetkov, Raymond Li, Loubna Ben allal, Jia LI, Chenghao Mou, Yacine Jernite, Margaret Mitchell, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro Von Werra, and Harm de Vries. 2023. The stack: 3 TB of permissively licensed source code. Transactions on Machine Learning Research.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip H. S. Torr, Fahad Shahbaz Khan, and Salman Khan. 2025. Llm post-training: A deep dive into reasoning large language models. Preprint, arXiv:2502.21321.

Chengpeng Li, Zheng Yuan, Hongyi Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. 2024. MuggleMath: Assessing the impact of query and response augmentation on math reasoning. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10230–10258, Bangkok, Thailand. Association for Computational Linguistics.

Zihao Li, Shaoxiong Ji, Hengyu Luo, and Jörg Tiedemann. 2025. Rethinking multilingual continual pretraining: Data mixing for adapting llms across languages and resources. Preprint, arXiv:2504.04152.

Geyu Lin, Bin Wang, Zhengyuan Liu, and Nancy F. Chen. 2025. CrossIn: An efficient instruction tuning approach for cross-lingual knowledge alignment. In Proceedings of the Second Workshop on Scaling Up Multilingual & Multi-Cultural Evaluation, pages 12–23, Abu Dhabi. Association for Computational Linguistics.

Elita Lobo, Chirag Agarwal, and Himabindu Lakkaraju. 2025. On the impact of fine-tuning on chain-of-thought reasoning. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 11679–11698, Albuquerque, New Mexico. Association for Computational Linguistics.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2024. Wizardcoder: Empowering code large language models with evol-instruct. In The Twelfth International Conference on Learning Representations.

Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. 2025. Datadecide: How to predict best pretraining data with small experiments. In Forty-second International Conference on Machine Learning.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. 2 olmo 2 furious. Preprint, arXiv:2501.00656.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024a. The fineweb datasets: Decanting the web for the finest text data at scale. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Martin Jaggi, Leandro von Werra, and Thomas Wolf. 2024b. Fineweb2: A sparkling update with 1000s of languages.

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, and 25 others. 2025. Qwen2.5 technical report. Preprint, arXiv:2412.15115.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality. In Findings of the Association for Computational Linguistics: ACL 2024, pages 2304–2317, Bangkok, Thailand. Association for Computational Linguistics.

Gerald Shen, Zhilin Wang, Olivier Delalleau, Jiaqi Zeng, Yi Dong, Daniel Egert, Shengyang Sun, Jimmy Zhang, Sahil Jain, Ali Taghibakhshi, Markel Sanz Ausin, Ashwath Aithal, and Oleksii Kuchaiev. 2024. Nemo-aligner: Scalable toolkit for efficient model alignment. Preprint, arXiv:2405.01481.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In The Eleventh International Conference on Learning Representations.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. CoRR, abs/1909.08053.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

David Stap, Vlad Niculae, and Christof Monz. 2023. Viewing knowledge transfer in multilingual machine translation through a representational lens. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 14973–14987, Singapore. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. Preprint, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. Preprint, arXiv:2307.09288.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. ReFT: Reasoning with reinforced fine-tuning. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7601–7614, Bangkok, Thailand. Association for Computational Linguistics.

Yejie Wang, Keqing He, Guanting Dong, Pei Wang, Weihao Zeng, Muxi Diao, Yutao Mou, Mengdi Zhang, Jingang Wang, Xunliang Cai, and Weiran Xu. 2024a. Dolphcoder: Echo-locating code large language models with diverse and multi-objective instruction tuning. Preprint, arXiv:2402.09136.

Zengzhi Wang, Xuefeng Li, Rui Xia, and Pengfei Liu. 2024b. Mathpile: A billion-token-scale pretraining corpus for math. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. 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.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, and 375 others. 2023. Bloom: A 176b-parameter open-access multilingual language model. Preprint, arXiv:2211.05100.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. Preprint, arXiv:2505.09388.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. Preprint, arXiv:2409.12122.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

A Base Model Architectures

Table 6 shows the architectures of the base models used in our experiments.

B Training of llama-inhouse Model

B.1 Datasets

We construct a training corpus for our llama-inhouse model by collecting text from a variety of sources. Table 7 summarizes the corpus statistics. The corpus comprises a total of 15.6 trillion tokens and includes texts in English, Japanese, Chinese, Korean, and programming code. English accounts for 91% of the corpus, while Japanese constitutes 2%.

B.2 Training

We trained the model using Megatron-LM (Shoeybi et al., 2019). The detailed hyperparameters are provided in Table 8. Training was conducted using up to 16 nodes, each equipped with eight NVIDIA H100 GPUs (80 GB).

C Instruction Pre-Training Details

C.1 Instruction-Response Dataset

Table 17 presents the instruction-response datasets used in our IPT experiments, categorized by dataset.

	llmjp3	qwen2.5	llama-inhouse	llama3.2
Params	1.8B	1.5B	1.3B	1.2B
Layers	24	28	16	16
Model Dimension	2,048	1,536	2,048	2,048
FFN Dimension	7,168	8,960	7,168	8,192
Attention Heads	16	12	16	32
Key Value Heads	16	2	8	8
Vocabulary Size	99,574	151,936	99,574	128,256

Table 6: Architecture specifications of the base models used in our experiments.

Language	Source	Subset	# of tokens [T]
English	Fineweb (sampled) (Penedo et al., 2024a)	-	14.0
	Dolma (Soldaini et al., 2024)	Gutenberg	0.005
	Dolma	peS2o	0.06
	Dolma	Reddit	0.08
	Dolma	Wikipedia	0.004
	gsm8k (Cobbe et al., 2021a)	-	0.000003
	FineMath-4+ (Allal et al., 2025)	-	0.01
	MathPile_Commercial (Wang et al., 2024b)	-	0.009
	olmo-mix-1124 (OLMo et al., 2025)	Algebraic Stack	0.01
	olmo-mix-1124	arXiv	0.02
	olmo-mix-1124	OpenWebMath	0.01
	dolmino-mix-1124 (OLMo et al., 2025)	StackExchange	0.001
	Wikipedia	-	0.005
Japanese	Fineweb2 (Penedo et al., 2024b)	-	0.3
	Wikipedia	-	0.001
Chinese	Fineweb2	-	0.8
	Wikipedia	-	0.0008
Korean	Fineweb2	-	0.05
	Wikipedia	-	0.0003
Code	olmo-mix-1124	StarCoder	0.1
	The Stack (Kocetkov et al., 2023)	-	0.1
Total	-	-	15.6

Table 7: Statistics of the raw corpora used for VPT in our llama-inhouse model. Token counts are based on segmentation with the llmjp3 tokenizer.

Hyperparameters	Value
Training steps	1.9M
Global Batch size	1,024
Max sequence length	8,192
Adam beta1	0.9
Adam beta2	0.95
Learning Rate (max)	3×10^{-4}
Learning Rate (min)	3×10^{-5}
Weight decay	0.1
Warmup steps	2,000
Decay style	cosine

Table 8: Hyperparameters for VPT in our llama-inhouse model.

C.2 Prompt for Category Labeling

Prompt for category labeling

You are an AI assistant that categorizes texts into one of the following four categories:

1. **Mathematics** - Texts related to mathematical calculations, proofs, equations.
2. **Coding** - Texts related to programming, scripting, debugging, or software development.
3. **Reasoning** - Texts that involve deductive or inductive reasoning, puzzles, or logical inference.
4. **General** - Any texts that do not fall into the above three categories.

Given the following text, classify it into the most appropriate category:

[Text]
{text}

Respond with only the category name.

C.3 Raw Corpora

Table 9 shows the statistics of the raw corpora used in our IPT experiments. To balance the number of training tokens between English and Japanese, some Japanese corpora were upsampled by a factor of two.

C.4 Model Training

We used Megatron-LM (Shoeybi et al., 2019) for our instruction pre-training experiments. The detailed hyperparameters are provided in Table 10.

D Supervised Fine-Tuning Details

We use Nemo-Aligner (Shen et al., 2024) for our supervised fine-tuning experiments. The detailed hyperparameters are listed in Table 11.

E Evaluation Details

For MT-Bench evaluation, we use llm-jp-judge v1.0.0[14]. For the evaluation of HumanEval, JHumanEval, GSM8K, and MGSM, we employ `swallow-evaluation`[15]. This tool extends `bigcode-evaluation-harness`[16] and `lm-evaluation-harness`[17], enabling integrated assessment of LLM performance in both English and Japanese. We adopt pass@1 as the evaluation metric for HumanEval and JHumanEval, and accuracy for GSM8K and MGSM.

F Ablation on SFT Category Composition

Our SFT dataset is less balanced across categories than the instruction-response dataset used for IPT, raising concerns that this imbalance might affect our conclusions. To address this issue, we constructed a more balanced SFT dataset by sampling from both the original SFT data and the instruction-response dataset. Table 13 shows the statistics of this dataset. Using this balanced SFT dataset, we repeated the experiments described in Section 4.3 as an ablation study. We used llmjp3 as the target model and evaluated performance with MT-Bench. The results, summarized in Table 14, are consistent with our main findings: coding performance is largely determined during IPT, whereas math and reasoning continue to benefit substantially from SFT.

G Scaling Study on Model Size

In this section, we conduct additional experiments using a 13B-scale model to partially examine whether the trends observed at the 1B scale persist for larger models. While our main experiments are limited to 1B-scale models due to computational constraints, previous studies (Kaplan et al., 2020; Hoffmann et al., 2022; Magnusson et al., 2025) suggest that qualitative trends in data composition may hold across model scales. Motivated by this, we compare model size (1.8B vs. 13B) and the number of IPT training steps (25k vs. 100k) to observe performance changes associated with scaling.

G.1 Experimental Setup

We used `llm-jp-3-13B`[18], which was vanilla pretrained on the same corpus and token count as llmjp3 (1.8B), as our 13B model. Table 15 shows the detailed architecture. The experimental configuration was almost identical to that described in Section 3.3, except that the number of IPT training steps was increased from 25k to 100k. The ratio between instruction-response data and raw corpora was fixed at approximately 60%, which corresponds to the setting that achieved the highest MT-Bench scores for the llmjp3 (1.8B) model in Section 4.1.1. For each model, we compared performance with and without SFT, evaluating them on both English and Japanese MT-Bench.

G.2 Results

Table 16 presents the results. For the 1.8B model, increasing the number of training steps from 25k to 100k slightly improved the final post-SFT score —from 4.52 to 4.64 on the English MT-Bench and from 5.95 to 6.09 on the Japanese MT-Bench— though the gain was modest. Regarding model size, even without SFT, the 13B model outperformed the 1.8B model by about +1.0 point on average, with particularly large improvements in the *Humanities*, *STEM*, and *Writing* categories. After applying SFT, the 13B model achieved an average MT-Bench score of 6.58 in English and 6.95 in Japanese, surpassing the 1.8B model and showing an overall performance improvement across most categories.

[14] https://github.com/llm-jp/llm-jp-judge/tree/v1.0.0

[15] https://github.com/swallow-llm/swallow-evaluation

[16] https://github.com/bigcode-project/bigcode-evaluation-harness

[17] https://github.com/EleutherAI/lm-evaluation-harness

[18] https://huggingface.co/llm-jp/llm-jp-3-13b

Language	Source	Raw tokens [B]	Upsampling Factor	Final Sampling Ratio [%]
English	Dolma	945.3	1	45.61
	Wikipedia	4.7	1	0.23
Japanese	Common Crawl	381.4	2	36.80
	Kaken	0.9	2	0.09
	NDL WARP HTML	1.4	2	0.13
	NDL WARP PDF (e0)	30.1	2	2.90
	NDL WARP PDF (e0.2)	177.2	1	8.55
	Wikipedia	1.3	2	0.12
Chinese	Wikipedia	0.8	1	0.04
Korean	Wikipedia	0.3	1	0.02
Code	The Stack	114.0	1	5.50

Table 9: Composition and sampling statistics of the raw corpora used in our IPT experiments. "Raw Tokens" refers to the original size before sampling. Following the LLM-jp method, some Japanese subsets are upsampled by a factor of 2 to balance language proportions. "Final Sampling Ratio" reflects upsampling and normalization across all components.

	llmjp3	qwen2.5	llama-inhouse	llama3.2
Training Steps	25k	25k	25k	25k
Global Batch size	1,024	1,024	1,024	1,024
Max Sequence Length	4,096	8,192	8,192	8,192
Adam Beta1	0.9	0.9	0.9	0.9
Adam Beta2	0.95	0.95	0.95	0.95
Learning Rate (max)	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Learning Rate (min)	3×10^{-5}	3×10^{-5}	3×10^{-5}	3×10^{-5}
Weight Decay	0.1	0.1	0.1	0.1
Warmup Steps	2,000	2,000	2,000	2,000
Decay Style	cosine	cosine	cosine	cosine

Table 10: Hyperparameters for IPT.

Hyperparameters	Value
Number of Epochs	2
Global Batch Size	64
Adam Beta1	0.9
Adam Beta2	0.98
Learning Rate (max)	2×10^{-5}
Learning Rate (min)	2×10^{-6}
Weight Decay	0.1
Warmup Steps	20
Warmup Style	linear
Decay Style	cosine

Table 11: Hyperparameters for SFT.

Overall, increasing the number of IPT training steps resulted in limited performance gains, whereas enlarging the model size yielded clear improvements. The performance gains attributed to incorporating instruction-response data, as observed at the 1.8B scale, were also reproduced at the 13B scale, which is consistent with theoretical predictions of scaling laws (Hoffmann et al., 2022).

H Potential Risks

While this work aims to improve the understanding of the design of instruction pre-training corpora, it also carries potential risks. Models trained with large-scale instruction-response data may inherit or even amplify biases present in the underlying datasets, such as cultural, linguistic, or demographic imbalances. Moreover, greater reliance on instruction-response data may propagate stylistic or behavioral biases originating from specific sources, potentially affecting downstream alignment and fairness.

	llmjp3		qwen2.5		llama-inhouse		llama3.2	
	en only	ja only	en only	ja only	en only	ja only	en only	ja only
MT-Bench (EN, coding)	$4.58_{-0.35}$	$4.74_{-0.19}$	$5.39_{+0.08}$	$4.75_{-0.56}$	$4.21_{-0.26}$	$3.95_{-0.52}$	$4.23_{-0.37}$	$4.58_{-0.02}$
MT-Bench (EN, extraction)	$4.17_{-0.67}$	$4.78_{-0.06}$	$5.58_{-1.04}$	$6.26_{-0.36}$	$2.85_{-1.4}$	$3.67_{-0.58}$	$3.95_{-0.56}$	$5.27_{+0.76}$
MT-Bench (EN, humanities)	$6.12_{+0.08}$	$6.17_{+0.13}$	$6.08_{-0.14}$	$6.18_{-0.04}$	$4.86_{-0.74}$	$5.43_{-0.17}$	$5.79_{-0.37}$	$5.58_{-0.58}$
MT-Bench (EN, math)	$3.92_{-0.42}$	$4.10_{-0.24}$	$4.87_{-0.16}$	$4.77_{-0.26}$	$3.08_{-0.68}$	$3.84_{+0.08}$	$4.79_{+0.53}$	$3.46_{-0.80}$
MT-Bench (EN, reasoning)	$3.61_{-1.00}$	$4.08_{-0.53}$	$3.93_{-0.71}$	$3.61_{-1.03}$	$3.74_{-0.09}$	$2.94_{-0.89}$	$3.21_{-0.52}$	$2.69_{-1.04}$
MT-Bench (EN, roleplay)	$5.78_{+0.04}$	$6.03_{+0.29}$	$5.61_{-0.22}$	$5.56_{-0.27}$	$4.57_{-0.54}$	$4.83_{-0.28}$	$5.29_{-0.29}$	$5.27_{-0.31}$
MT-Bench (EN, stem)	$4.68_{-0.28}$	$4.88_{-0.08}$	$5.43_{-0.35}$	$5.80_{+0.02}$	$4.23_{-0.79}$	$4.54_{-0.48}$	$4.55_{-0.54}$	$4.79_{-0.30}$
MT-Bench (EN, writing)	$5.39_{-0.57}$	$6.19_{+0.23}$	$5.53_{-0.37}$	$5.71_{-0.19}$	$4.51_{-0.73}$	$5.33_{+0.09}$	$4.94_{-0.53}$	$5.50_{+0.03}$
MT-Bench (EN, AVG)	$4.78_{-0.40}$	$5.12_{-0.06}$	$5.30_{-0.37}$	$5.33_{-0.34}$	$4.01_{-0.65}$	$4.32_{-0.34}$	$4.59_{-0.33}$	$4.64_{-0.28}$
HumanEval (EN)	$0.27_{-0.03}$	$0.27_{-0.03}$	$0.31_{-0.02}$	$0.31_{-0.02}$	$0.24_{-0.01}$	$0.20_{-0.05}$	$0.27_{-0.02}$	$0.26_{-0.03}$
GSM8K (EN)	$0.31_{-0.04}$	$0.29_{-0.06}$	$0.35_{-0.05}$	$0.46_{+0.06}$	$0.16_{-0.05}$	$0.22_{+0.01}$	$0.24_{-0.07}$	$0.29_{-0.02}$
MT-Bench (JA, coding)	$4.35_{-0.51}$	$4.93_{+0.07}$	$4.78_{-0.77}$	$5.29_{-0.26}$	$4.00_{-0.49}$	$4.70_{+0.21}$	$4.32_{-0.08}$	$4.29_{-0.11}$
MT-Bench (JA, extraction)	$4.96_{-1.11}$	$5.99_{-0.08}$	$5.46_{-0.90}$	$6.34_{-0.02}$	$4.31_{-0.69}$	$5.48_{+0.48}$	$4.64_{-0.87}$	$5.44_{-0.07}$
MT-Bench (JA, humanities)	$6.17_{-0.97}$	$7.53_{+0.39}$	$4.81_{-2.18}$	$7.13_{+0.14}$	$4.89_{-1.83}$	$7.51_{+0.79}$	$4.78_{-2.10}$	$6.78_{-0.10}$
MT-Bench (JA, math)	$5.14_{+0.13}$	$5.06_{+0.05}$	$4.98_{-0.88}$	$6.05_{+0.19}$	$3.98_{-0.89}$	$5.35_{+0.48}$	$4.13_{-0.62}$	$4.39_{-0.36}$
MT-Bench (JA, reasoning)	$4.21_{-0.68}$	$4.44_{-0.45}$	$3.87_{-0.94}$	$5.13_{+0.32}$	$3.31_{-0.66}$	$4.51_{+0.54}$	$3.33_{-0.10}$	$4.41_{+0.98}$
MT-Bench (JA, roleplay)	$6.21_{-0.57}$	$7.22_{+0.44}$	$4.69_{-2.04}$	$6.84_{+0.11}$	$4.74_{-1.76}$	$6.69_{+0.19}$	$5.02_{-1.01}$	$6.56_{+0.53}$
MT-Bench (JA, stem)	$5.13_{-0.99}$	$6.45_{+0.33}$	$3.83_{-2.71}$	$6.63_{+0.09}$	$4.23_{-1.45}$	$6.33_{+0.65}$	$4.32_{-1.63}$	$5.89_{-0.06}$
MT-Bench (JA, writing)	$5.66_{-1.09}$	$6.68_{-0.07}$	$4.52_{-2.36}$	$6.73_{-0.15}$	$4.74_{-1.69}$	$7.16_{+0.73}$	$4.79_{-1.6}$	$6.23_{-0.16}$
MT-Bench (JA, AVG)	$5.23_{-0.72}$	$6.04_{+0.09}$	$4.62_{-1.59}$	$6.27_{+0.06}$	$4.28_{-1.18}$	$5.97_{+0.51}$	$4.42_{-1.00}$	$5.50_{+0.08}$
HumanEval (JA)	$0.24_{+0.00}$	$0.24_{+0.00}$	$0.26_{-0.05}$	$0.29_{-0.02}$	$0.21_{+0.00}$	$0.21_{+0.00}$	$0.23_{+0.00}$	$0.25_{+0.02}$
GSM8K (JA)	$0.29_{-0.03}$	$0.33_{+0.01}$	$0.34_{-0.18}$	$0.50_{-0.02}$	$0.18_{-0.09}$	$0.29_{+0.02}$	$0.20_{-0.15}$	$0.35_{+0.00}$

Table 12: Impact of imbalanced language composition in the instruction-response data (full results). We compare two settings: English-only and Japanese-only instruction-response data. The subscript in each cell indicates the absolute difference from the baseline score, which was obtained by training on an instruction-response dataset containing approximately equal amounts of English and Japanese data. Improvements over the baseline are shown in green, while degradations are shown in red.

Language	Category	# of samples	# of tokens [M]
English	code	29,949	22.5
	general	13,288	23.1
	math	15,953	21.1
	reasoning	12,105	17.1
Japanese	code	33,933	23.6
	general	82,181	23.8
	math	37,498	20.8
	reasoning	43,846	23.4
Total		268,753	175.5

Table 13: Statistics of SFT dataset for ablation study. The number of tokens (response only) is calculated using llmjp3 tokenizer.

Data Ratio(%)	coding		math		reasoning	
	EN	JA	EN	JA	EN	JA
baseline	$2.57_{+148.9}$	$2.60_{+116.7}$	$2.09_{+67.1}$	$3.16_{+205.9}$	$2.70_{+134.8}$	$3.07_{+100.4}$
25/25/25/25	$4.13_{+5.8}$	$4.74_{+6.6}$	$3.57_{-4.7}$	$4.77_{+0.5}$	$3.48_{+9.4}$	$4.68_{+50.3}$
0/33/33/33	$3.97_{+9.2}$	$4.16_{+17.1}$	$4.61_{+32.9}$	$4.70_{-15.1}$	$5.36_{+77.5}$	$4.99_{+34.2}$
50/17/17/17	$4.17_{-1.4}$	$4.57_{+4.7}$	$3.86_{-6.2}$	$5.28_{+33.2}$	$3.51_{+9.5}$	$4.71_{+33.2}$
100/0/0/0	$4.81_{+21.8}$	$4.86_{+13.0}$	$3.56_{+2.6}$	$5.04_{+31.0}$	$3.14_{+17.2}$	$4.22_{+65.6}$
33/0/33/33	$4.66_{+2.3}$	$4.28_{-5.2}$	$4.69_{+18.8}$	$5.00_{+0.7}$	$4.39_{+15.0}$	$4.38_{+37.5}$
17/50/17/17	$4.47_{+2.8}$	$4.47_{+12.3}$	$4.03_{-6.0}$	$4.58_{-7.2}$	$3.34_{+4.0}$	$4.39_{+33.8}$
0/100/0/0	$3.02_{+14.8}$	$3.27_{+18.8}$	$2.36_{-0.5}$	$3.09_{+26.1}$	$3.23_{-4.0}$	$3.34_{+2.2}$
33/33/0/33	$4.52_{-12.0}$	$4.38_{+5.2}$	$3.87_{+13.7}$	$4.39_{+4.5}$	$4.29_{+26.8}$	$5.41_{+64.8}$
17/17/50/17	$4.01_{-5.0}$	$4.52_{-5.5}$	$4.53_{+39.5}$	$5.64_{+2.0}$	$3.78_{+4.6}$	$4.22_{+39.2}$
0/0/100/0	$4.68_{+9.6}$	$4.16_{+5.3}$	$4.45_{+9.0}$	$5.65_{+28.4}$	$3.21_{+6.5}$	$4.38_{+19.0}$
33/33/33/0	$4.49_{-0.5}$	$4.80_{+9.5}$	$4.26_{+43.4}$	$4.94_{+3.4}$	$2.89_{+4.4}$	$3.41_{-5.7}$
17/17/17/50	$4.84_{+24.2}$	$4.21_{-5.8}$	$4.09_{-1.5}$	$5.08_{-6.2}$	$4.76_{+34.6}$	$5.02_{+32.0}$
0/0/0/100	$3.64_{+9.2}$	$3.91_{+13.9}$	$4.27_{+18.1}$	$5.15_{+23.1}$	$4.68_{+43.2}$	$4.21_{+18.5}$

Table 14: Benchmark scores on coding, math, and reasoning tasks, along with relative improvements during the SFT stage. Each model is instruction-pretrained with different category data compositions. Data ratios are shown in the order of code/general/math/reasoning. The "Baseline" model is trained using only raw corpora in IPT. "EN" and "JA" indicate evaluation results in English and Japanese. The values shown in each cell denote the scores after SFT. Subscripts denote the relative improvement rate (%) from before to after SFT, computed as $(\text{after} - \text{before})/\text{before} \times 100$. Score improvements exceeding 20% are highlighted in green.

Params	1.8B	13B
Layers	24	40
Model Dimension	2,048	5,120
FFN Dimension	7,168	13,824
Attention Heads	16	40
Key Value Heads	16	40
Vocabulary Size	99,574	99,574

Table 15: Architecture specifications of llmjp3 1.8B and 13B models.

Model	Coding	Extraction	Humanities	Math	Reasoning	Roleplay	Stem	Writing	AVG
English MT-Bench									
1.8B 25k w/o SFT	4.58 (0.12)	4.33 (0.15)	5.67 (0.40)	3.33 (0.03)	3.30 (0.05)	5.17 (0.33)	4.93 (0.26)	4.82 (0.19)	4.52 (0.04)
1.8B 25k	4.93 (0.58)	4.84 (0.41)	6.04 (0.37)	4.34 (0.56)	4.61 (0.13)	5.74 (0.31)	4.96 (0.25)	5.96 (0.32)	5.18 (0.16)
1.8B 100k w/o SFT	4.50 (0.10)	4.05 (0.15)	5.83 (0.10)	3.57 (0.03)	3.58 (0.08)	4.92 (0.35)	5.43 (0.18)	5.22 (0.60)	4.64 (0.06)
1.8B 100k	5.05 (0.26)	5.18 (0.21)	6.44 (0.29)	4.80 (0.23)	3.63 (0.37)	6.13 (0.45)	5.34 (0.31)	6.11 (0.23)	5.34 (0.15)
13B 100k w/o SFT	5.35 (0.15)	5.93 (0.15)	6.88 (0.51)	3.90 (0.05)	4.62 (0.03)	6.13 (0.20)	5.97 (0.10)	6.52 (0.12)	5.66 (0.13)
13B 100k	5.59 (0.15)	6.82 (0.13)	8.24 (0.19)	5.47 (0.65)	5.00 (0.44)	6.89 (0.24)	6.97 (0.27)	7.70 (0.44)	6.58 (0.12)
Japanese MT-Bench									
1.8B 25k w/o SFT	4.10 (0.09)	4.38 (0.25)	6.37 (0.36)	4.23 (0.08)	3.43 (0.10)	6.37 (0.42)	5.50 (0.31)	6.32 (0.39)	5.09 (0.08)
1.8B 25k	4.86 (0.17)	6.07 (0.30)	7.14 (0.17)	5.01 (0.13)	4.89 (0.14)	6.78 (0.41)	6.12 (0.21)	6.75 (0.32)	5.95 (0.07)
1.8B 100k w/o SFT	4.63 (0.25)	5.45 (0.17)	6.88 (0.20)	5.58 (0.03)	3.57 (0.12)	5.82 (0.16)	5.75 (0.17)	6.78 (0.35)	5.56 (0.07)
1.8B 100k	5.12 (0.20)	6.12 (0.28)	7.61 (0.22)	5.38 (0.39)	4.51 (0.10)	7.34 (0.29)	5.91 (0.29)	6.69 (0.32)	6.09 (0.08)
13B 100k w/o SFT	5.07 (0.19)	6.77 (0.15)	7.63 (0.21)	5.40 (0.10)	4.58 (0.12)	7.07 (0.30)	6.47 (0.26)	6.93 (0.38)	6.24 (0.06)
13B 100k	5.59 (0.18)	6.26 (0.44)	8.46 (0.22)	6.42 (0.42)	5.88 (0.20)	7.96 (0.13)	7.33 (0.21)	7.74 (0.28)	6.95 (0.11)

Table 16: MT-Bench results (English and Japanese) for models of different scales and IPT budgets, evaluated with and without SFT. The numbers in parentheses indicate standard deviations.

Name	Lang.	Category	Source	# of tokens [B]
chatbot-arena-mixtral8x22b-ja	ja	general	https://huggingface.co/datasets/kanhatakeyama/ChatbotArenaJaMixtral8x22b	0.002
	ja	coding		0.001
	ja	math		0.0002
	ja	reasoning		0.0004
coding-en-jp	en	coding	https://huggingface.co/datasets/Aratako/Synthetic-JP-EN-Coding-Dataset-801k	0.5
	ja	coding		0.2
dolly-nemotron340b-ja	ja	general	https://huggingface.co/datasets/kanhatakeyama/databricks-dolly-15k-ja-regen-nemotron	0.002
flan	en	general	https://huggingface.co/datasets/allenai/dolmino-mix-1124	18.5
general-multiturn-calm3-ja	ja	general	https://huggingface.co/datasets/kanhatakeyama/0717-calm3-22b-random-genre-inst-sft-tsub-part,https://huggingface.co/datasets/kanhatakeyama/0717-calm3-22b-random-genre-inst-sft-tsub,https://huggingface.co/datasets/kanhatakeyama/0719-calm3-22b-random-genre-inst-sft-multiturn-tsub,https://huggingface.co/datasets/kanhatakeyama/0722-calm3-22b-random-genre-inst-sft-multiturn-tsub,https://huggingface.co/datasets/kanhatakeyama/0723-calm3-22b-random-genre-inst-sft-multiturn-clean-tsub	8.1
	ja	coding		4.9
	ja	math		1.8
	ja	reasoning		2.0
logical-mixtral8x22b-ja	ja	general	https://huggingface.co/datasets/kanhatakeyama/LogicalDatasetsByMixtral8x22b	0.01
	ja	coding		0.03
	ja	math		0.007
	ja	reasoning		0.007
logical-multiturn-calm3-ja	ja	reasoning	https://huggingface.co/datasets/kanhatakeyama/0804calm3-logical-multiturn-pretrain	5.7
logical-wizardlm7b-en	en	general	https://huggingface.co/datasets/kanhatakeyama/logical-wizardlm-7b	0.3
	en	code		11.8
	en	math		13.9
	en	reasoning		1.6
logical-wizardlm7b-ja	ja	general	https://huggingface.co/datasets/kanhatakeyama/logical-wizardlm-7b-ja,https://huggingface.co/datasets/kanhatakeyama/logical-wizardlm-7b-ja-0730,https://huggingface.co/datasets/kanhatakeyama/logical-wizardlm-7b-ja-0731,https://huggingface.co/datasets/kanhatakeyama/logical-wizardlm-7b-ja-0805	0.09
	ja	coding		5.6
	ja	math		6.3
	ja	reasoning		0.5
logical-wizardlm8x22b-en	en	general	https://huggingface.co/datasets/kanhatakeyama/logicaltext-wizardlm8x22b	0.1
	en	coding		0.008
	en	math		0.006
	en	reasoning		0.16
logical-wizardlm8x22b-ja	ja	general	https://huggingface.co/datasets/kanhatakeyama/logicaltext-wizardlm8x22b-Ja,https://huggingface.co/datasets/kanhatakeyama/logicaltext-wizardlm8x22b-api	0.09
	ja	coding		0.04
	ja	math		0.04
	ja	reasoning		0.4
open-math-inst-phi3-ja	ja	math	https://huggingface.co/datasets/kanhatakeyama/OpenMathInstruct-ja-phi3	0.02
open-orca-mixtral8x22b-ja	ja	general	https://huggingface.co/datasets/kanhatakeyama/OrcaJaMixtral8x22b	0.2
	ja	coding		0.002
	ja	math		0.002
	ja	reasoning		0.04
rc-multiturn-calm3-ja	ja	general	https://huggingface.co/datasets/kanhatakeyama/multiturn-Calm3-manual	0.2
wiki-qa-mixtral8x22b-ja	ja	general	https://huggingface.co/datasets/kanhatakeyama/AutoWikiQA	0.03
jaster	ja	general	https://github.com/llm-jp/llm-jp-eval	0.04
stack-exchange-en	en	general	https://huggingface.co/datasets/allenai/dolmino-mix-1124	0.3
	en	coding		1.0
	en	math		0.1
	en	reasoning		0.03
dolmino-synth-math-en	en	math	https://huggingface.co/datasets/allenai/dolmino-mix-1124	0.03
gsm8k	en	math	https://huggingface.co/datasets/allenai/dolmino-mix-1124	0.002
llmjp-magpie-sft-ja	ja	general	https://huggingface.co/datasets/llm-jp/magpie-sft-v1.0	0.03
	ja	coding		0.003
	ja	math		0.0006
	ja	reasoning		0.0009
daring-anteater-en	en	general	https://huggingface.co/datasets/nvidia/Daring-Anteater	0.2
	en	coding		0.03
	en	math		0.01
	en	reasoning		0.004
r1-distill-qwen-pseudo-qa-ja	ja	general	https://huggingface.co/datasets/hpprc/r1-distill-qwen-pseudo-qa	1.0
	ja	coding		0.009
	ja	math		0.01
	ja	reasoning		0.01
oasst-en	en	general	https://huggingface.co/datasets/llm-jp/oasst1-21k-en,https://huggingface.co/datasets/llm-jp/oasst2-33k-en	0.02
	en	coding		0.005
	en	math		0.0007
	en	reasoning		0.001
qwq-longcot-en	en	general	https://huggingface.co/datasets/amphora/QwQ-LongCoT-130K	0.03
	en	coding		0.009
	en	math		0.03
	en	reasoning		0.01
llm-jp-extraction-ja	ja	general	https://huggingface.co/datasets/llm-jp/extraction-wiki-ja	0.01
llm-jp-math-ja	ja	math	original	0.006
llm-jp-reasoning-mistral24b-en	en	reasoning	original	3.2
llm-jp-reasoning-qwen32b-en	en	reasoning	original	3.3

Table 17: Statistics of our instruction-response datasets. Token counts are based on segmentation by the llmjp3 tokenizer. “original” indicates that the datasets we created.