The Best Instruction-Tuning Data are Those That Fit

Dylan Zhang
University of Illinois Urbana–Champaign
shizhuo2@illinois.edu

Qirun Dai
University of Chicago
qirundai@uchicago.edu

Hao Peng
University of Illinois Urbana–Champaign
haopeng@illinois.edu

Abstract

High-quality supervised finetuning (SFT) data are essential for unlocking pretrained LLMs' capabilities. Typically, instructions are paired with responses from various sources by humans annotators or other LMs, which are often out of the distribution of the target model to be finetuned. This, at scale, can lead to diminishing returns and even hurt the models' performance and robustness. We hypothesize that SFT is most effective with data aligned to the model's pretrained distribution and propose GRAPE— a novel SFT framework that tailors supervision to the target model. For each instruction, it **g**athers **r**esponses from various sources, and selects the one that **a**ligns most closely to the target model's **pre**trained distribution, as measured by the normalized probability. We then proceed with standard SFT with these selected responses. We first evaluate GRAPE with a controlled experiment, where we sample various solutions for each question in UltraInteract from multiple models and finetune on GRAPE-selected data using LMs from different families including LLaMA.1-8B, Mistral-7B and Qwen2.5-7B. GRAPE significantly outperforms strong baselines, including distilling from the strongest model with absolute gain up to 13.8% averaging across benchmarks, and a baseline trained on $3\times$ more data with maximum 17.3% performance improvements. GRAPE's strong performance generalizes to off-the-shelf SFT data. We use GRAPE to subsample responses from the post-training data used for Tulu3 and Olmo-2. GRAPE can outperform strong baselines with 4.5 times the data by 6.1% and state-of-the-art data selection approaches by 3.9% on average performance. Remarkably, using 1/3 data and half number of epochs, GRAPE allows LLaMA.1-8B to surpass the performance of Tulu3-SFT by 3.5%. Our findings highlight that aligning supervision with the pretrained distribution offers a simple yet powerful way to improve SFT efficiency and performance.

1 Introduction

High-quality, large-scale supervised data is crucial for supervised fine-tuing(SFT; Databricks, 2023; Köpf et al., 2023; Zhao et al., 2024a; Zheng et al., 2024). A common practice of collecting SFT data involves sampling responses from strong language models, predominantly focusing on expanding the size of the dataset and improving the overall quality of the responses (Sun et al., 2023; Taori et al., 2023; Wang et al., 2023; Xu et al., 2024c; Chen et al., 2024). However, recent research suggests that there is more complex dynamics involved (Xu et al., 2024d). A plateau effect in synthetic data scaling, where performance either stagnates or even declines as the size of the synthetic data increases beyond a certain point, has been widely observed. This phenomenon arises due to issues such as diminishing diversity (Padmakumar & He, 2024; Guo et al., 2023) and distortion in the data

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

distribution (LeBrun et al., 2021), which ultimately undermine the base model's performance and robustness (Alemohammad et al., 2024; Gerstgrasser et al., 2024; Shumailov et al., 2023; Dohmatob et al., 2024; Hataya et al., 2023; Martínez et al., 2023a,b; Bohacek & Farid, 2023; Briesch et al., 2023).

Thus, effective SFT requires more than scaling up the data; it often needs "tailoring" the data to the unique characteristics of the target model. Existing works focus on enhancing the model's existing knowledge and capabilities (Du et al., 2023) and optimizing the curriculum progression for instruction tuning (Zhao et al., 2024b; Lee et al., 2024; Feng et al., 2023; Setlur et al., 2024). They typically find questions the model should best learn, instead of what answers the model should imitate.

Meanwhile, tailoring the responses to the target model has been a crucial ingredient for the success of later phases of LLM development, particularly through on-policy preference learning (Tajwar et al., 2024; Zhang et al., 2024c,a; Miao et al., 2024; Gulcehre et al., 2023; Azar et al., 2023; Tang et al., 2024a; Zhuang et al., 2023), and on-policy/online reinforcement learning (Guo et al., 2024b; Liu et al., 2024d; Zhou et al., 2024c).

Inspired by these insights, we hypothesize that SFT can similarly benefit from aligning data with the model, the core idea behind GRAPE. For each instruction, GRAPE gathers and selects response(s) from various sources that are closest to the target model's pretrained distribution. This is achieved by calculating the probability of each response using the target model and selects the one with the highest length-normalized probability(§3).After obtaining these more "in-distribution" responses, GRAPE proceeds with standard SFT without any modification to the training.

Unlike existing datasets that usually contains one-size-fits-all responses for each instruction without customization (Yu et al., 2024; Yuan et al., 2024b; Lian et al., 2023b; Teknium, 2023), GRAPE curates model-dependent SFT datasets that better matches the base model's distribution, better mitigating the risks associated with distribution shift like spurious correlations (Zhou et al., 2024d) and catastrophic forgetting (Luo et al., 2025; Kotha et al., 2024), while posing minimum overhead of single forward pass over the candidate set. In return, GRAPE allows better downstream performance with reduced training compute.

To GRAPE's advantage, many existing datasets share overlapping instructions but contain different high-quality responses, e.g., the instructions in Flan (Longpre et al., 2023), GSM-8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), and the post-training recipes that re-use SFT instructions for preference learning (Lambert et al., 2024; OLMo et al., 2025). Therefore, GRAPE can directly select, for each model, the best fit(s) among the off-the-shelf responses without having to produce new responses, which proves effective in our experiments (§ 5).

We first validate our hypothesis through extensive controlled experiments on a reasoning dataset with chain-of-thoughts, UltraInteract-SFT (Yuan et al., 2024b), and demonstrate the importance of supervising base models with in-distribution responses(§4). We experimented on 4 popular pretrained LMs from Mistral (MistralAI, 2024c), Llama3.1 (Dubey et al., 2024) and Qwen2.5 (Hui et al., 2024) families. Notably, models fine-tuned with GRAPE-selected responses outperform those trained on a $3\times$ larger datasets up to 17.3% absolute gain on average performances even with less compute, even surpass models trained on responses from the strongest teacher under consideration—LLAMA3.1-405B-INSTRUCT—by significant margins.

We then experiment with a more realistic setting for general-domain instruction-tuning, collecting responses from post-training data for Tulu3 and Olmo-v2. Again, GRAPE demonstrates its effectiveness by outperforming state-of-the-art data selection approaches by avg. 4.6% and a strong baseline trained on all these available data that is 4.5 times larger by up to 6.1%. Remarkably, GRAPE allows finetuning a Llama3.1-8B base model to exceed the performance of Tulu-8B-SFT using 1/3 data and half of the epochs.

Our results reveal that distributional alignment is a crucial and previously underappreciated dimension of effective instruction tuning. GRAPE introduces this perspective with a simple, scalable algorithm that consistently outperforms strong baselines with far less data and compute for the actual training.

2 Background and Motivation

2.1 Data Engineering for Instruction Tuning

Data is central to the success of effective instruction tuning, (Xu et al., 2023; Xia et al., 2024; Chan et al., 2024), featuring both automated data synthesis (Xu et al., 2024a; Zeng et al., 2024; Yu et al., 2024; Wei et al., 2023) and selection (Xia et al., 2024; Chen et al., 2023a; Parkar et al., 2024; Li et al., 2024e). Some selection approaches focus on high-quality data by leveraging LLMs (Chen et al., 2023a; Parkar et al., 2024; Li et al., 2024b) or employing principled metrics (Kang et al., 2024; Mekala et al., 2024; Xia et al., 2024), while others, such as Yang et al. (2024b); Das & Khetan (2023), aim to identify diversity-optimized subsets for greater efficiency. An emerging trend is the customization of training data based on the characteristics of the base models. (Li et al., 2024c; Du et al., 2023; Li et al., 2024a). These methods typically reweight or filter ***instructions***. However, a key overlooked aspect is ***response*** selection—specifically, choosing responses that align with the model's pretrained distribution, which may be critical for preserving useful behaviors and ensuring effective fine-tuning.

2.2 Toward Distribution-Aligned Supervised Fine-Tuning

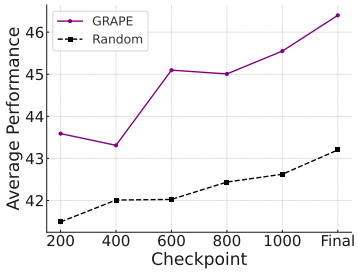

Figure 1: Average performance curve of LLAMA3.1-8B on TULU-3-OLMO-2 combo of GRAPE against random selection

An Analogy from Reinforcement Learning and Preference Learning The investigation of this work into the distribution match between the pre-trained LM and supervised fine-tuning (SFT) data is inspired by recent findings on policy optimization for LM alignment with RL (Ouyang et al., 2022) and preference learning (Rafailov et al., 2023; Ethayarajh et al., 2024). While the importance of matching training data distribution with the policy has been well noted in both traditional RL (Shi et al., 2023; Fujimoto et al., 2018; Kumar et al., 2019; Peng et al., 2019; Wang et al., 2021; Arora & Goyal, 2023; Jiang & Li, 2016; Tang & Abbeel, 2010) and LM settings (Xiong et al., 2024), preference learning algorithms like DPO (Rafailov et al., 2023), IPO (Azar et al., 2023) and KTO (Ethayarajh et al., 2024) first emerged as off-policy algorithms. However, subsequent research has highlighted the performance gap between on-policy and off-policy training due to distribution shifts (Xu et al., 2024b; Tang et al., 2024b) and proposed various mitigation strategies (Zhuang et al., 2023; Zhou et al., 2024c; Zhang et al., 2024a; Xiong et al., 2024; Guo et al., 2024b), showing that training models on data more closely aligned with their policy distribution can significantly improve performance, while failing to do so can yield sub-optimal policies or those that are harder to generalize. Works like SPIN (Yuan et al., 2024a) echo the intuition by gradually improving policy through self-play to mitigate distribution shift.

Hypothesis: Supervised Fine-tuning Benefits From Data That Better Matches Base Distribution The base distribution of pre-trained language models—shaped by extensive training on vast and diverse datasets—is inherently robust and generalizable (Brown et al., 2020; Saunshi et al., 2021). Therefore, during supervised fine-tuning phase, the pre-trained distribution should be carefully preserved (Kumar et al., 2022; Cohen-Wang et al., 2024; He et al., 2023; Yang et al., 2024d; Ding et al., 2023), to best retain the knowledge and capabilities that emerge during pre-training (Zhou et al., 2023). If the proximity between the pre-trained distribution and the fine-tuning data is not maintained, the limited number of training examples available during SFT, compared to the vast scale of pre-training data, can increase the risk of distribution distortion. This misalignment can lead to issues such as catastrophic forgetting (Aghajanyan et al., 2020; Yang et al., 2024d) and the emergence of spurious correlations (Feldman, 2021).

The central premise of our work is that by using responses closely aligned to the pre-trained distribution, we can minimize distribution shift during SFT and therefore achieve better data efficiency and stronger performance.

From On-Policy Alignment To Distribution-Aligned SFT We build on principles of on-policy alignment techniques with key distinctions tailored for SFT. Yet, given that SFT represents an earlier

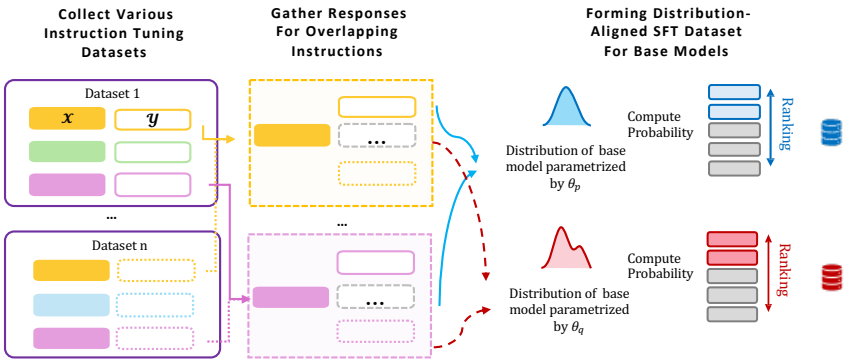

Figure 3: An overview of GRAPE. GRAPE takes multiple off-the-shelf existing datasets and optionally generates new responses, finds overlapping instructions with multiple different responses and selects responses that align with the base model's distribution. This dataset is then used for standard supervised fine-tuning.

stage of post-training than RL, sampling responses from the base model itself alone can lead to model collapse, as noted in (Shumailov et al., 2024). Prior studies have documented similar risks like instability, bias reinforcement, knowledge stagnation, and overfitting (Herel & Mikolov, 2024; Mobahi et al., 2020; Allen-Zhu & Li, 2023; Ghosh et al., 2024; Dong et al., 2025; Zhang et al., 2024d). To address this, we advocate for an approach that stays more in-distribution while delivering effective supervision to the base model. To this end, we propose to gather and select responses from various sources, and select one that is closest to the target model's pretrained distribution, which we name GRAPE.

3 Methodology

We introduce GRAPE, a surprisingly simple yet effective methodology to enhance supervised fine-tuning (SFT) by customizing the training data for the base model. The key idea is to find a response, among a candidate pool, for each instruction x_i that aligns closely with the base model's pretrained distribution π_{θ_0}.

As diagrammed in Figure 3, GRAPE consists of two main steps, followed by standard SFT:
Response Collection (§3.1) Collect a pool of high-quality candidate responses from various sources.
Customization (§3.2): For the target model to be fine-tuned π_{θ_0}, find the response(s), for each instruction, that are closest to the pretrained distribution of π_{θ_0}.

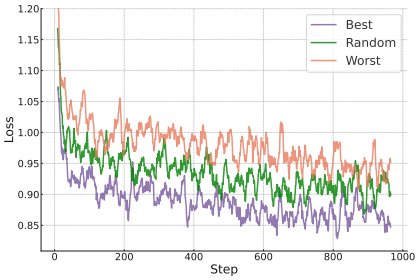

Figure 2: Training loss curve of Llama3.1-8B on TULU-3-OLMO-2. **Best** (highest-probability, selected by GRAPE), **Random**, and **Worst** (lowest-probability) responses show a clear loss ordering: **Best** < **Random** < **Worst**.

3.1 Collecting Responses from Existing Resources

For instruction-tuning of language models, high-quality instructions are more difficult to collect than responses (Xu et al., 2024c; Liu et al., 2024a). Therefore, it is a common practice to reuse existing instruction-tuning prompts while generating diverse responses using various methods tailored to specific requirements. For instance, instructions from Flan (Longpre et al., 2023), OpenOrca (Lian et al., 2023a), ShareGPT (Team, 2023), and the training splits of GSM-8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), and CodeContests (Li et al., 2022) are frequently reused in datasets like Olmo (OLMo et al., 2025), Tulu (Lambert et al., 2024), OpenHermes (Teknium, 2023), OpenOrca (Lian et al., 2023a), MetaMath (Yu et al., 2024), MathInstruct (Yue et al., 2023), UltraFeedback (Cui et al., 2024), and UltraInteract (Yuan et al., 2024b), whether for SFT or preference learning. The solutions are generated using different

models or follow varying styles depending on the specific needs. This naturally leads to a situation where a single instruction with multiple responses becomes a readily available resource. GRAPE therefore leverages these pre-existing response candidates to tailor training dataset that better aligns with the base model's distribution. When such resources are unavailable or insufficient, practitioners can generate new responses and apply GRAPE.

For each instruction, we collect multiple responses from various datasets. These responses form a candidate set associated with the instruction.

3.2 Customize Dataset For Models

We then compute the conditional probability of each response $\pi_{\theta_0}(y_i^j \mid x_i)$. Practically, we format each example using a simple prompt template: `Question:` **{instruction}** `\n Answer:` **{response}**. For each instruction, we rank its candidate responses based on the conditional log-probability assigned by the base model, normalized by response length. This is equivalent to ranking them from lowest to highest perplexity where $\texttt{Perplexity} = \exp\left(-\frac{1}{N}\sum_{t=1}^{N}\log P(x_t \mid x_{<t})\right)$. We then select the responses with the **highest** normalized probability (i.e., lowest perplexity) for supervision. Figure 4 shows a clear difference in models' choices among the same candidate pool, indicating that GRAPE-selected datasets are highly customized towards different models.

Since GRAPE only involves forward-pass log-probability computation (no gradients or optimization), it is highly efficient and simple to integrate into any SFT pipeline with minimal overhead compared with model-based data selection approaches (Xia et al., 2024; Yang et al., 2024b; Liu et al., 2024b; Zhao et al., 2021; Zhang et al., 2024b; Pan et al., 2024). Additionally, it is important to distinguish GRAPE from the other perplexity-based data selection and curriculum planning methods (Wu et al., 2024; Li et al., 2024b; Liu et al., 2024c). Existing approaches focus on selecting **instructions** by using perplexity as a difficulty measure, which differs from GRAPE that uses probability to select for each instruction in a fixed instruction set, responses that better matches with the base model's distribution. Our experiments in §5 demonstrate that low-probability responses with fixed set of instructions are detrimental to performance, further emphasizing the fundamental difference in the two processes.

4 Controlled Experiments Show the Benefits of Distributional Alignment

We conduct controlled experiments using UltraInteract-SFT to test whether selecting responses aligned with a base model's distribution improves fine-tuning outcomes more than relying on stronger generators or scaling data size. This setup—focused on verifiable tasks in coding, logic, and math—isolates the effect of distribution matching. Results show that GRAPE-selected responses consistently outperform alternatives, validating distributional alignment as a key supervision signal. These findings motivate our broader evaluations in §5.

4.1 Experimental Setup

Training Data Curation In this controlled experiment, we focus on chain-of-thought reasoning (Wei et al., 2022; Wang et al., 2024a; Luo et al., 2024; Cobbe et al., 2021; Li et al., 2023; Lightman et al., 2023). Different models may follow different reasoning paths to solve a problem, while their final solutions can be easily verified.

We use UltraInteract-SFT (Yuan et al., 2024b), which contains approximately $80,800$ unique instructions covering coding, math (chain-of-thought and program-aided) and logic reasoning domains , where each instruction is paired with varying numbers (avg. 3.5/instruction) of different responses to contain a total $> 280,000$ training examples. The responses in the dataset are strictly in step-wise format. For each instruction, we construct a response pool consisting of both (i) original UltraInteract-SFT responses and (ii) additional responses generated from a diverse set of LLMs. GRAPE is then applied to select the most in-distribution response per instruction to this enlarged candidate pool and ensure the number of responses matches the original UltraInteract-SFT dataset for fair comparisons.

We collect responses from a diverse set of models of various sizes across model families, including MIXTRAL-7X7B-INSTRUCT (Jiang et al., 2024), CODESTRAL-22B (MistralAI, 2024a), MISTRAL-SMALL (MistralAI, 2024b), LLAMA-3.1-70B-INSTRUCT and LLAMA-3.1-405B-

INSTRUCT (Dubey et al., 2024), and QWEN2.5-72B-CHAT (Yang et al., 2024a), resulting in approximately 10x additional responses per instruction. The responses are then filtered based on the answers to ensure their validity following Yuan et al. (2024b).

Base Models To demonstrate the generalizability of GRAPE, we evaluate its performance across multiple LLMs, including LLAMA-3.1-8B and LLAMA-3.2-3B from LLAMA-3 (Grattafiori et al., 2024) family, MISTRAL-7B (Jiang et al., 2023) and QWEN2.5-7B (Hui et al., 2024). We ensure that all training configurations (GRAPE, baselines) use the same number of instructions and responses per instruction as original UltraInteract-SFT, unless otherwise stated.

Evaluation We evaluate the model on coding and math reasoning benchmarks. For coding tasks, we consider HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), LeetCode (Guo et al., 2024a); for math datasets, we consider **MATH** dataset (Hendrycks et al., 2021b), **GSM-Plus** (Li et al., 2024d) and **TheoremQA** (Chen et al., 2023b) dataset. **HumanEval** and **MBPP** are natural-language-to-code benchmarks testing language models' ability to produce functionally correct programs. **LeetCode** contains interview-level programming problems that are more challenging. **MATH** contains high-school level math competition problems, whereas **GSM-Plus** is a more challenging variant of GSM-8k (Cobbe et al., 2021) and **Theorem-QA** contains complex math reasoning problems.

Baselines We compare GRAPE against several baselines to isolate the impact of response selection: **Original Dataset** performs standard SFT on the unmodified UltraInteract-SFT dataset, which includes verified correct responses. **Strongest-Model Responses** uses only responses from the most powerful generator available—LLAMA3.1-405B-INSTRUCT. This represents a strong upper bound on data quality and helps determine whether GRAPE offers additional gains beyond simply choosing a strong model. **3×Data:** use 3 times the number of distinct, validated responses per instruction relative to UltraInteract, while keeping all training hyperparameters (learning rate, number of epochs, etc.) fixed. It directly tests whether GRAPE's gains stem from strategic data selection rather than merely scaling up data volume. [1] We train all models for 1 epoch with a learning rate of 10^{-5}.

Model	Data	HE	LC	MBPP	MATH		GSMPlus		TheoremQA		Avg.	Abs.
					CoT	PoT	CoT	PoT	CoT	PoT		Δ
MISTRAL-7B	Original-UI	46.3	15.6	50.1	21.6	32.6	45.9	45.3	16.8	20.1	32.7	*3.6*
	Llama3.1-405B	44.5	12.2	46.9	24.5	33.8	48.0	50.0	17.5	16.8	32.7	*3.6*
	3x Data	48.1	13.3	52.8	25.1	26.1	50.0	49.4	16.8	12.4	32.7	*3.6*
	GRAPE	**52.4**	**15.6**	**53.4**	**28.9**	**34.6**	**50.5**	**52.8**	**17.8**	**20.6**	**36.3**	-
LLAMA3.1-8B	Original-UI	54.3	11.1	58.9	29.7	31.0	53.7	51.6	20.0	20.8	36.8	*4.7*
	Llama3.1-405B	56.7	15.0	60.0	34.8	38.1	51.4	55.4	16.6	21.0	38.8	*2.7*
	3x Data	48.8	7.8	57.9	25.5	11.2	48.6	45.1	20.6	19.6	31.7	*9.8*
	GRAPE	**57.3**	**19.4**	**63.8**	**34.8**	**39.2**	**56.6**	**56.1**	**22.5**	**23.9**	**41.5**	-
LLAMA3.2-3B	Original-UI	32.9	3.9	41.6	12.8	16.1	30.8	19.5	14.6	10.5	20.3	*3.8*
	Llama3.1-405B	31.7	5.0	43.3	6.6	6.6	30.8	20.6	15.1	10.8	18.9	*5.1*
	3x Data	42.6	6.7	42.9	8.7	5.1	17.8	19.5	14.6	**12.8**	19.0	*5.1*
	GRAPE	**42.6**	**13.3**	**44.6**	**16.4**	**17.6**	**34.9**	**20.6**	**15.1**	11.4	**24.1**	-
QWEN2.5-7B	Original-UI	67.0	41.2	60.0	51.0	38.3	64.1	59.5	14.6	10.5	45.1	*11.6*
	Llama3.1-405B	71.3	45.0	62.0	31.5	35.1	40.1	36.4	33.3	31.2	42.9	*13.8*
	3x Data	75.6	48.3	62.9	32.8	24.5	47.1	22.8	21.6	19.0	39.4	*17.3*
	GRAPE	**77.4**	**48.9**	**70.7**	**56.4**	**45.3**	**67.7**	**66.3**	**37.4**	**40.1**	**56.7**	-

Table 1: Result of synthetic experiment on UltraInteract-SFT. The last column, **Abs.** Δ is GRAPE's absolute improvement on average performance over that row.

4.2 Results and Analysis

Table 1 summarizes the performance of GRAPE across benchmarks. Our approach consistently outperforms the various baselines across the board, including the original UltraInteract-SFT dataset. GRAPE-selected solutions can outperform those directly sampled from the strongest model under consideration (LLAMA3.1-405B-INSTRUCT) up to 13.8% absolute improvement. This implies that customization for base models should be prioritized over identifying the presumably highest-quality responses. This verifies our central premise that being in-distribution with each base model is an important ingredient for the responses we supervise the base models on, to boosting downstream performance. Furthermore, we demonstrate that merely adding more responses does not always lead

[1] For instance, if UltraInteract contains 3 validated responses for instruction x, this setting uses 9.

to continuous improvement in model performance, which aligns with findings in prior studies (Li et al., 2024c; Du et al., 2023). By properly aligning with models' base distributions, GRAPE outperforms those trained with 3x responses with at least 3.6% and up to 17.3% absolute improvement. These results reinforce the notion that scaling data without considering its alignment with the base model's initial distribution risks diminishing returns and, in some cases, even performance degradation.

5 GRAPE-Picking From Real-World SFT Datasets

In this section, we leverage the findings from the earlier experiments and demonstrate the effectiveness of GRAPE to customize training data for each base model by selecting from available datasets with overlapping instructions. Here, we do *not* generate any new responses for the instructions; it only selects from existing ones. We evaluate GRAPE on the fully open dataset used in post-training phases of TULU-3 (Lambert et al., 2024) and OLMO-2 (OLMo et al., 2025). The details are presented below. We discuss additional results in Appendix B

5.1 Data Mixture Details

TULU-3 (Lambert et al., 2024) is a fully open-source collection of post-training recipes, including supervised fine-tuning and preference alignment data. OLMO-2 (OLMo et al., 2025) is a fully open-source language model. Both TULU-3 and OLMO-2 use the same data mixture during the supervised fine-tuning stage, but different data mixtures and source models for generating preference data for different sizes of their models: Tulu-3-8B/70B and Olmo-2-7B/13B. To demonstrate the effectiveness of GRAPE, we collected the overlapping instructions from both models and gather their corresponding responses. From the preference data, we retained only the winning responses. We formed our candidate pool with those instructions with at least two distinct responses, resulting in a dataset of 350.4K unique instructions and about 1.03 million total instruction-response pairs for evaluation with GRAPE. We do *not* apply further processing of these data or any filtering on top of GRAPE.

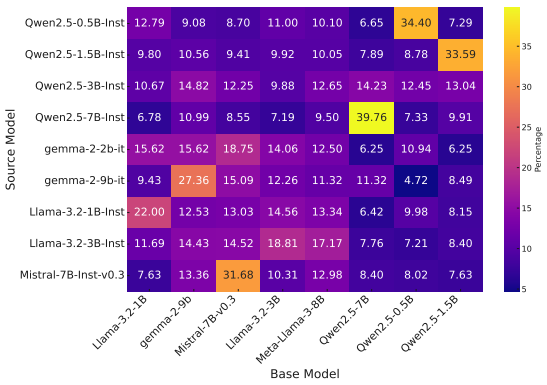

Figure 4: Breakdown of GRAPE-selected responses for 1K Tulu instructions vary significantly across base models, reflecting its highly model-oriented nature over responses. Details in Appendix H

5.2 Evaluation

We evaluate on a set of commonly used benchmarks spanning over coding, math, knowledge and instruction-following. We evaluated on LeetCode (Guo et al., 2024a), MATH (Hendrycks et al., 2021b), BigBenchHard(BBH) (Suzgun et al., 2022), MMLU (Hendrycks et al., 2021a), and AlpacaEval-V2 (Dubois et al., 2024). LeetCode, MATH, BBH and MMLU are evaluated the in the same way as in (Yuan et al., 2024b), where we use zero-shot for MATH and MMLU, 3-shot example for BBH. We use the same AlpacaEval-v2 as in OpenInstruct.

5.3 Baselines

We extensively compare against three baseline types: controlled baselines with fixed instructions, scaling baselines with increased data, and state-of-the-art selection methods. Additional details in Appendix C

Controlled Baselines We include three baselines to isolate the effect of GRAPE's response selection. **SFT-only** replaces GRAPE-selected responses with those from the original SFT dataset, pairing each instruction with a standard reference response to measure improvement over presumably good

SFT responses. **Random** selects candidate responses uniformly at random from the pool for the same set of instructions, establishing a noise-tolerant baseline. **Reverse-GRAPE** instead selects responses with the highest perplexity. We test if responses diverging the most from the base model's distribution degrade performance, providing a contrast that sharpens the effectiveness of GRAPE.

Scaling Baselines To assess how GRAPE compares under larger-scale training, we consider three scaling-oriented baselines. **Tulu3-SFT** uses all 939K SFT instances from the Tulu3 training mixture to test whether GRAPE can still outperform despite using only a subset of the instruction pool. **All Responses** trains over the entire 1.04M-instance candidate pool to demonstrate the effect of selection versus brute-force inclusion. **All Available Data** uses all 1.58M instruction-response pairs under consideration, roughly 4.5× the data used by GRAPE, to test whether data volume alone suffices.

State-Of-The-Art SFT Data Selection Approaches We compare GRAPE against recent state-of-the-art data selection methods. **LESS** (Xia et al., 2024) selects data based on influence scores on validation tasks. We follow the implementation setup from Dai et al. (Dai et al., 2025). **Emb-NV** selects training data close to validation in embedding space using NV-Embed-V2 (Lee et al., 2025), following (Ivison et al., 2025). **S2L** (Yang et al., 2024b) clusters data via loss trajectories from small reference models (LLAMA-3.2-1B, QWEN2.5-0.5B, MISTRAL-V0.3-7B) and samples uniformly across clusters to match GRAPE's data budget.

Model	Data	Num. Instances	AlpacaEval2		BBH	MMLU	MATH	LeetCode	Avg.	Abs. Δ
			LC	WR						
LLAMA 3.1-8B	**Highest**	350.4k	10.1	6.2	68.9	63.2	22.9	13.3	30.8	5.2
	Random	350.4k	12.8	10.8	68.6	63.1	27.9	13.3	32.8	3.2
	SFT-Only	350.4k	7.1	5.5	68.9	64.1	20.2	17.2	30.5	5.4
	Tulu3-SFT	939k	12.4	8.0	67.9	65.9	31.5*	7.8	32.4	3.5
	All Responses	1.03M	12.9	11.4	68.7	62.8	32.1	17.2	34.2	1.8
	S2L	350.4k	8.5	7.6	68.6	63.1	26.5	16.1	31.7	4.2
	Emb-NV	350.4k	8.4	7.0	67.9	63.7	32.1	17.2	32.7	3.2
	LESS	350.4k	7.3	6.0	68.2	63.3	25.1	16.1	31.0	4.0
	All Available Data	1.58M	8.8	10.1	**69.8**	62.1	**32.5**	16.1	33.2	2.7
	GRAPE	350.4k	**14.8**	**15.2**	69.6	**64.5**	32.1	**19.4**	**35.9**	-
MISTRAL -7B	**Highest**	350.4k	7.0	5.5	58.8	56.1	15.1	10.6	25.5	6.4
	Random	350.4k	10.6	9.2	60.0	57.8	19.6	11.1	28.1	3.9
	SFT-Only	350.4k	7.1	5.3	53.9	57.0	14.4	12.0	24.9	7.0
	Full-SFT-Data	939k	11.5	10.5	59.0	55.2	**25.8**	15.6	29.9	2.0
	All Responses	1.03M	10.5	11.5	61.0	57.9	24.2	14.4	29.9	2.0
	S2L	350.4k	10.5	11.9	61.9	57.1	22.4	13.9	29.6	2.3
	Emb-NV	350.4k	6.0	5.2	60.7	56.5	23.9	11.7	27.3	4.6
	LESS	350.4k	6.4	4.8	59.2	55.4	16.0	8.3	25.0	6.9
	All Available Data	1.58M	8.0	7.0	55.3	53.7	25.4	12.3	26.9	5.0
	GRAPE	350.4k	**13.6**	**13.9**	**62.3**	**59.2**	24.2	**18.3**	**31.9**	-
QWEN2.5 -7B	**Highest**	350.4k	8.0	10.7	72.2	73.2	49.4	42.2	42.6	5.9
	Random	350.4k	16.1	14.9	**73.3**	73.1	56.0	43.3	46.1	2.4
	SFT-Only	350.4k	9.9	7.8	71.2	74.1	51.1	46.6	43.4	5.1
	Full-SFT-Data	939k	9.5	7.1	71.4	73.1	47.0	**48.3**	42.7	5.8
	All Responses	1.03M	16.0	14.5	71.4	72.1	51.7	43.3	44.8	3.7
	S2L	350.4k	13.4	14.9	72.7	73.1	53.4	40.6	44.7	3.9
	Emb-NV	350.4k	11.9	10.6	72.1	72.5	53.3	43.3	44.0	4.6
	LESS	350.4k	7.8	5.9	71.3	72.9	48.3	41.1	41.2	7.4
	All Available Data	1.58M	13.3	12.3	70.3	71.8	44.0	42.8	42.4	6.1
	GRAPE	350.4k	**20.0**	**20.4**	73.2	**73.3**	**60.0**	44.4	**48.6**	-

Table 2: GRAPE on the Tulu-Olmo collection. For Llama3.1-8B base model, we included Tulu3-SFT model's results. The "*"-marked number for MATH is obtained using 4-shot prompting. We train all the models (except that we took Tulu3-SFT numbers directly) for 1 epoch with a learning rate of 10^{-5}. Abs.Δ is GRAPE's absolute improvement on average performance over that row.

5.4 Results

As shown in Table 2, models fine-tuned on responses selected by GRAPE outperforms the strong baselines we constructed, especially the one that trains over all available data by significant margins across the 3 models.

	GRAPE	QWEN2.5 -72B	LLAMA3.1 -405B	GEMMA -IT-9B
LC	**28.1**	25.8	16.3	26.9
WR	**33.1**	24.3	17.0	20.2

Table 3: Alpaca-Eval2 On Magpie-Zoo Xu et al. (2024d).

Remarkably, using roughly 1/6 training computation (Tulu3-8B-SFT was trained for 2 epochs on 3 times of data), our performance exceeds that of TULU3-8B-SFT.

Also, GRAPE outperforms state-of-the-art data-selection approaches like S2L, despite its simplicity and efficiency, further highlighting its effectiveness in diverse real-world scenarios and making it a practical option for real-world SFT setups with minimal data engineering effort. Without the need to synthesize any new data, one can easily leverage established datasets sourced from the web to customize a dataset for each base model that yields better fine-tuning outcome.

These results highlight GRAPE as an effective and efficient selection strategy for real-world SFT.

5.5 Ablations

Remains Effective Even with a Single Response Source In Section 4, we demonstrated how GRAPE enables practitioners to refine model-generated responses for improved training outcomes. Beyond that, GRAPE can optimize responses from a single generator. The Magpie-Zoo (Xu et al., 2024d) dataset contains a fixed set of instructions and multiple versions of response sets each generated by different language models. Using the Magpie-Zoo instruction set, we sample 10 responses per instruction from Qwen2.5-72B-Instruct, select in-distribution responses with GRAPE, to train a Mistral-v0.3-7B model. We compare with replicas that achieve top-3 performance from Magpie-Zoo.

As shown in Table 3, GRAPE-selected responses further boost the performance. Given that batch sampling from a strong model introduces minimal latency (Zhong et al., 2024; Zhou et al., 2024e), this result positions GRAPE as a practical and efficient data curation strategy—achieving strong results with just a single generator and no additional engineering overhead.

Model	Data	Avg.
MISTRAL-7B	Self-Distilled	28.4(-)
	Original-UI	32.7
LLAMA3.1-8B	Self-Distilled	29.4(-)
	Original-UI	36.8
LLAMA3.2-3B	Self-Distilled	15.1(-)
	Original-UI	20.3

Table 4: Performance Degradation From Self-Distillation On UI.

Why GRAPE Outperforms Self-Generated Responses To probe the effectiveness of GRAPE, we ablate it against a degenerate alternative: self-generation, where the model is fine-tuned on its own outputs.

Dataset	Model	Response Generator	N	Acc.
MATH	MISTRAL	Llama3.1-70B	10	18.2
MATH	MISTRAL	FT-Mistral	10	15.9 (-)
MATH	LLEMMA	Llama3.1-70B	10	26.2
MATH	LLEMMA	FT-Llemma	10	23.6 (-)
MATH	MISTRAL	MM-AnsAug	–	22.3
MATH	MISTRAL	FT-Mistral	10	20.6 (-)
MATH	LLEMMA	MM-AnsAug	–	28.1
MATH	LLEMMA	FT-Llemma	10	21.4 (-)

Table 5: Self-generation on MATH dataset. FT-MISTRAL refers to the model right above that row finetuned from either MM-AnsAug or Llama3.1-70B-Instruct produced solutions. N stands for the number of responses sampled.

We fine-tune a base model using responses generated by its previously fine-tuned variant. This setup consistently degrades performance (Table 4). We further confirm this on the MATH dataset (Hendrycks et al., 2021b): responses from strong models like LLAMA3.1-70B-INSTRUCT or MetaMathQA (Yu et al., 2024) are used to fine-tune a model that then generates new solutions for another round of fine-tuning. Again, performance drops (Table 5).

This failure arises from distributional collapse (see § 2.2): self-generated responses become increasingly narrow and repetitive, reinforcing biases and reducing exposure to diverse reasoning. Correctness alone is insufficient—external diversity is essential for generalization.

GRAPE avoids this collapse by selecting external responses that are both diverse and distribution-aligned, preserving semantic breadth and stylistic variability while staying true to the model's pretraining. This enables more stable and generalizable fine-tuning.

6 Conclusion

We present GRAPE, a simple yet effective method for improving supervised fine-tuning by selecting responses aligned with the base model's pretrained distribution. GRAPE requires only a forward pass over candidate responses, making it highly efficient and easy to integrate. Despite its simplicity, GRAPE consistently outperforms stronger baselines using significantly larger datasets and surpasses more complex, costly data selection methods. Our study affirms that carefully aligning SFT data with a model's pretrained distribution yields substantial performance and efficiency gains.

7 Discussion and Limitations

Response versus Instance Level Selection GRAPE selects responses: it begins with a fixed set of instructions and evaluates multiple candidate responses for each. This differs from instance-level selection approaches (e.g., Kung et al. (2023); Li et al. (2024c); Wang et al. (2024b)), which focus on choosing which instructions to include based on factors such as coverage, skill-balancing or difficulty. In contrast, GRAPE focuses on the quality of supervision—that is, selecting responses to provide the most effective learning signal. Importantly, this response-centric perspective is complementary, not contradictory, to instance-level or instruction-based selection methods; both might be combined to enhance overall data quality and training efficiency.

Limitations Like many other data selection algorithms (Du et al., 2023; Li et al., 2024c; Xia et al., 2024; Das & Khetan, 2023; Kang et al., 2024; Mekala et al., 2024; Yang et al., 2024c; Zhang et al., 2024b; Pan et al., 2024; Dai et al., 2025; Ivison et al., 2025; Bhatt et al., 2024; Yin & Rush, 2024; Liu et al., 2024b), GRAPE assumes a quality-controlled candidate pool from which to select samples. Furthermore, because GRAPE relies on the base model itself, its selection effectiveness may be influenced by the model's inherent capabilities.

8 Acknowledgment

This project is partly supported by NSF under award No. 2019897. This research used the DeltaAI advanced computing and data resource, which is supported by the National Science Foundation (award OAC 2320345) and the State of Illinois. DeltaAI is a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications.

References

Aghajanyan, A., Shrivastava, A., Gupta, A., Goyal, N., Zettlemoyer, L., and Gupta, S. Better fine-tuning by reducing representational collapse, 2020. URL https://arxiv.org/abs/2008.03156.

Alemohammad, S., Casco-Rodriguez, J., Luzi, L., Humayun, A. I., Babaei, H., LeJeune, D., Siahkoohi, A., and Baraniuk, R. Self-consuming generative models go MAD. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ShjMHfmPs0.

Allen-Zhu, Z. and Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning, 2023. URL https://arxiv.org/abs/2012.09816.

Ankner, Z., Blakeney, C., Sreenivasan, K., Marion, M., Leavitt, M. L., and Paul, M. Perplexed by perplexity: Perplexity-based data pruning with small reference models, 2024. URL https://arxiv.org/abs/2405.20541.

Arora, S. and Goyal, A. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*, 2023.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., and Sutton, C. Program synthesis with large language models, 2021.

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences, 2023.

Bhatt, G., Chen, Y., Das, A. M., Zhang, J., Truong, S. T., Mussmann, S., Zhu, Y., Bilmes, J., Du, S. S., Jamieson, K., Ash, J. T., and Nowak, R. D. An experimental design framework for label-efficient supervised finetuning of large language models, 2024. URL https://arxiv.org/abs/2401.06692.

Bohacek, M. and Farid, H. Nepotistically trained generative-ai models collapse, 2023.

Briesch, M., Sobania, D., and Rothlauf, F. Large language models suffer from their own output: An analysis of the self-consuming training loop, 2023.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Chan, Y.-C., Pu, G., Shanker, A., Suresh, P., Jenks, P., Heyer, J., and Denton, S. Balancing cost and effectiveness of synthetic data generation strategies for llms, 2024. URL https://arxiv.org/abs/2409.19759.

Chen, J., Qadri, R., Wen, Y., Jain, N., Kirchenbauer, J., Zhou, T., and Goldstein, T. Genqa: Generating millions of instructions from a handful of prompts. *arXiv preprint arXiv:2406.10323*, 2024.

Chen, L., Li, S., Yan, J., Wang, H., Gunaratna, K., Yadav, V., Tang, Z., Srinivasan, V., Zhou, T., Huang, H., et al. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023a.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021.

Chen, W., Yin, M., Ku, M., Lu, P., Wan, Y., Ma, X., Xu, J., Wang, X., and Xia, T. Theoremqa: A theorem-driven question answering dataset, 2023b. URL https://arxiv.org/abs/2305.12524.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Cohen-Wang, B., Vendrow, J., and Madry, A. Ask your distribution shift if pre-training is right for you, 2024. URL https://arxiv.org/abs/2403.00194.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley-Interscience, Hoboken, NJ, USA, 2nd edition, 2006. ISBN 978-0-471-24195-9. URL https://onlinelibrary.wiley.com/doi/book/10.1002/047174882X.

Cui, G., Yuan, L., Ding, N., Yao, G., He, B., Zhu, W., Ni, Y., Xie, G., Xie, R., Lin, Y., Liu, Z., and Sun, M. Ultrafeedback: Boosting language models with scaled ai feedback, 2024. URL https://arxiv.org/abs/2310.01377.

Dai, Q., Zhang, D., Ma, J. W., and Peng, H. Improving influence-based instruction tuning data selection for balanced learning of diverse capabilities, 2025. URL https://arxiv.org/abs/2501.12147.

Das, D. and Khetan, V. Deft: Data efficient fine-tuning for large language models via unsupervised core-set selection. *arXiv preprint arXiv:2310.16776*, 2023.

Databricks. Databricks dolly-15k, 2023. URL https://huggingface.co/datasets/databricks/databricks-dolly-15k.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

Dohmatob, E., Feng, Y., Subramonian, A., and Kempe, J. Strong model collapse, 2024. URL https://arxiv.org/abs/2410.04840.

Dong, Q., Dong, L., Zhang, X., Sui, Z., and Wei, F. Self-boosting large language models with synthetic preference data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=7visV100Ms.

Du, Q., Zong, C., and Zhang, J. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*, 2023.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra,

S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Tan, X. E., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Grattafiori, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Vaughan, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Franco, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Wyatt, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Ozgenel, F., Caggioni, F., Guzmán, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Thattai, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Damlaj, I., Molybog, I., Tufanov, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Prasad, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Huang, K., Chawla, K., Lakhotia, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Tsimpoukelli, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Laptev, N. P., Dong, N., Zhang, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Li, R., Hogan, R., Battey, R., Wang, R., Maheswari, R., Howes, R., Rinott, R., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Kohler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Albiero, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wang, X., Wu, X., Wang, X., Xia, X., Wu, X., Gao, X., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Hao, Y., Qian, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., and Zhao, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dubois, Y., Galambosi, B., Liang, P., and Hashimoto, T. B. Length-controlled alpacaeval: A simple way to debias automatic evaluators, 2024. URL https://arxiv.org/abs/2404.04475.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Kto: Model alignment as prospect theoretic optimization, 2024.

Face, H. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Feldman, V. Does learning require memorization? a short tale about a long tail, 2021. URL https://arxiv.org/abs/1906.05271.

Feng, T., Wang, Z., and Sun, J. Citing: Large language models create curriculum for instruction tuning, 2023. URL https://arxiv.org/abs/2310.02527.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:54457299.

Gerstgrasser, M., Schaeffer, R., Dey, A., Rafailov, R., Korbak, T., Sleight, H., Agrawal, R., Hughes, J., Pai, D. B., Gromov, A., Roberts, D., Yang, D., Donoho, D. L., and Koyejo, S. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=5B2K4LRgmz.

Ghosh, S., Evuru, C. K. R., Kumar, S., S, R., Aneja, D., Jin, Z., Duraiswami, R., and Manocha, D. A closer look at the limitations of instruction tuning, 2024. URL https://arxiv.org/abs/2402.05119.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard,

J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Gulcehre, C., Paine, T. L., Srinivasan, S., Konyushkova, K., Weerts, L., Sharma, A., Siddhant, A., Ahern, A., Wang, M., Gu, C., Macherey, W., Doucet, A., Firat, O., and de Freitas, N. Reinforced self-training (rest) for language modeling, 2023. URL https://arxiv.org/abs/2308.08998.

Guo, D., Zhu, Q., Yang, D., Xie, Z., Dong, K., Zhang, W., Chen, G., Bi, X., Wu, Y., Li, Y. K., Luo, F., Xiong, Y., and Liang, W. Deepseek-coder: When the large language model meets programming – the rise of code intelligence, 2024a.

Guo, S., Zhang, B., Liu, T., Liu, T., Khalman, M., Llinares, F., Rame, A., Mesnard, T., Zhao, Y., Piot, B., Ferret, J., and Blondel, M. Direct language model alignment from online ai feedback, 2024b. URL https://arxiv.org/abs/2402.04792.

Guo, Y., Shang, G., Vazirgiannis, M., and Clavel, C. The curious decline of linguistic diversity: Training language models on synthetic text, 2023.

Hanawa, K., Yokoi, S., Hara, S., and Inui, K. Evaluation of similarity-based explanations, 2021. URL https://arxiv.org/abs/2006.04528.

Hataya, R., Bao, H., and Arai, H. Will large-scale generative models corrupt future datasets? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20555–20565, October 2023.

He, G., Chen, J., and Zhu, J. Preserving pre-trained features helps calibrate fine-tuned language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NI7StoWHJPT.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021a. URL https://arxiv.org/abs/2009.03300.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset, 2021b. URL https://arxiv.org/abs/2103.03874.

Herel, D. and Mikolov, T. Collapse of self-trained language models, 2024. URL https://arxiv.org/1abs/2404.02305.

Huang, S. C., Piqueres, A., Rasul, K., Schmid, P., Vila, D., and Tunstall, L. Open hermes preferences. https://huggingface.co/datasets/argilla/OpenHermesPreferences, 2024.

HuggingFace-H4. Openhermes-2.5-preferences-v0-deduped, 2024. URL https://huggingface.co/datasets/HuggingFaceH4/OpenHermes-2.5-preferences-v0-deduped.

Hui, B., Yang, J., Cui, Z., Yang, J., Liu, D., Zhang, L., Liu, T., Zhang, J., Yu, B., Lu, K., Dang, K., Fan, Y., Zhang, Y., Yang, A., Men, R., Huang, F., Zheng, B., Miao, Y., Quan, S., Feng, Y., Ren, X., Ren, X., Zhou, J., and Lin, J. Qwen2.5-coder technical report, 2024. URL https://arxiv.org/abs/2409.12186.

ichi Amari, S. *Information Geometry and Its Applications*, volume 194 of *Applied Mathematical Sciences*. Springer, Tokyo, Japan, 1st edition, 2016. ISBN 978-4-431-55977-3. doi: 10.1007/978-4-431-55978-0. URL https://doi.org/10.1007/978-4-431-55978-0.

Ivison, H., Zhang, M., Brahman, F., Koh, P. W., and Dasigi, P. Large-scale data selection for instruction tuning, 2025. URL https://arxiv.org/abs/2503.01807.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning, 2016. URL https://arxiv.org/abs/1511.03722.

Kang, F., Just, H. A., Sun, Y., Jahagirdar, H., Zhang, Y., Du, R., Sahu, A. K., and Jia, R. Get more for less: Principled data selection for warming up fine-tuning in llms. *arXiv preprint arXiv:2405.02774*, 2024.

Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z. R., Stevens, K., Barhoum, A., Nguyen, D., Stanley, O., Nagyfi, R., ES, S., Suri, S., Glushkov, D., Dantuluri, A., Maguire, A., Schuhmann, C., Nguyen, H., and Mattick, A. Openassistant conversations - democratizing large language model alignment. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 47669–47681. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/949f0f8f32267d297c2d4e3ee10a2e7e-Paper-Datasets_and_Benchmarks.pdf.

Kotha, S., Springer, J. M., and Raghunathan, A. Understanding catastrophic forgetting in language models via implicit inference. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VrHiF2hsrm.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c2073ffa77b5357a498057413bb09d3a-Paper.pdf.

Kumar, A., Raghunathan, A., Jones, R., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution, 2022. URL https://arxiv.org/abs/2202.10054.

Kung, P.-N., Yin, F., Wu, D., Chang, K.-W., and Peng, N. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks, 2023. URL https://arxiv.org/abs/2311.00288.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafjord, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tulu 3: Pushing frontiers in open language model post-training, 2024. URL https://arxiv.org/abs/2411.15124.

LeBrun, B., Sordoni, A., and O'Donnell, T. J. Evaluating distributional distortion in neural language modeling. In *International Conference on Learning Representations*, 2021.

Lee, B. W., Cho, H., and Yoo, K. M. Instruction tuning with human curriculum, 2024. URL https://arxiv.org/abs/2310.09518.

Lee, C., Roy, R., Xu, M., Raiman, J., Shoeybi, M., Catanzaro, B., and Ping, W. Nv-embed: Improved techniques for training llms as generalist embedding models, 2025. URL https://arxiv.org/abs/2405.17428.

Li, M., Chen, L., Chen, J., He, S., Gu, J., and Zhou, T. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 16189–16211, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.958. URL https://aclanthology.org/2024.findings-acl.958.

Li, M., Zhang, Y., He, S., Li, Z., Zhao, H., Wang, J., Cheng, N., and Zhou, T. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning, 2024b. URL https://arxiv.org/abs/2402.00530.

Li, M., Zhang, Y., Li, Z., Chen, J., Chen, L., Cheng, N., Wang, J., Zhou, T., and Xiao, J. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7602–7635, Mexico City, Mexico, June 2024c. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.421. URL https://aclanthology.org/2024.naacl-long.421.

Li, Q., Cui, L., Zhao, X., Kong, L., and Bi, W. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers, 2024d. URL https://arxiv.org/abs/2402.19255.

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Dal Lago, A., Hubert, T., Choy, P., de Masson d'Autume, C., Babuschkin, I., Chen, X., Huang, P.-S., Welbl, J., Gowal, S., Cherepanov, A., Molloy, J., Mankowitz, D. J., Sutherland Robson, E., Kohli, P., de Freitas, N., Kavukcuoglu, K., and Vinyals, O. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, December 2022. ISSN 1095-9203. doi: 10.1126/science.abq1158. URL http://dx.doi.org/10.1126/science.abq1158.

Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., and Chen, W. Making language models better reasoners with step-aware verifier. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5315–5333, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.291. URL https://aclanthology.org/2023.acl-long.291.

Li, Z., Hua, Y., Vu, T.-T., Zhan, H., Qu, L., and Haffari, G. Scar: Efficient instruction-tuning for large language models via style consistency-aware response ranking, 2024e. URL https://arxiv.org/abs/2406.10882.

Lian, W., Goodson, B., Pentland, E., Cook, A., Vong, C., and "Teknium". Openorca: An open dataset of gpt augmented flan reasoning traces. https://https://huggingface.co/Open-Orca/OpenOrca, 2023a.

Lian, W., Wang, G., Goodson, B., Pentland, E., Cook, A., Vong, C., and "Teknium". Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification, 2023b. URL https://https://huggingface.co/Open-Orca/SlimOrca.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Liu, R., Wei, J., Liu, F., Si, C., Zhang, Y., Rao, J., Zheng, S., Peng, D., Yang, D., Zhou, D., et al. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*, 2024a.

Liu, W., Zeng, W., He, K., Jiang, Y., and He, J. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning, 2024b. URL https://arxiv.org/abs/2312.15685.

Liu, Y., Liu, J., Shi, X., Cheng, Q., Huang, Y., and Lu, W. Let's learn step by step: Enhancing in-context learning ability with curriculum learning, 2024c. URL https://arxiv.org/abs/2402.10738.

Liu, Z., Lu, M., Zhang, S., Liu, B., Guo, H., Yang, Y., Blanchet, J., and Wang, Z. Provably mitigating overoptimization in rlhf: Your sft loss is implicitly an adversarial regularizer, 2024d. URL https://arxiv.org/abs/2405.16436.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., and Roberts, A. The flan collection: Designing data and methods for effective instruction tuning, 2023. URL https://arxiv.org/abs/2301.13688.

Luo, L., Liu, Y., Liu, R., Phatale, S., Guo, M., Lara, H., Li, Y., Shu, L., Zhu, Y., Meng, L., Sun, J., and Rastogi, A. Improve mathematical reasoning in language models by automated process supervision, 2024. URL https://arxiv.org/abs/2406.06592.

Luo, Y., Yang, Z., Meng, F., Li, Y., Zhou, J., and Zhang, Y. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025. URL https://arxiv.org/abs/2308.08747.

Marion, M., Üstün, A., Pozzobon, L., Wang, A., Fadaee, M., and Hooker, S. When less is more: Investigating data pruning for pretraining llms at scale, 2023a. URL https://arxiv.org/abs/2309.04564.

Marion, M., Üstün, A., Pozzobon, L., Wang, A., Fadaee, M., and Hooker, S. When less is more: Investigating data pruning for pretraining llms at scale, 2023b. URL https://arxiv.org/abs/2309.04564.

Martínez, G., Watson, L., Reviriego, P., Hernández, J. A., Juarez, M., and Sarkar, R. Combining generative artificial intelligence (ai) and the internet: Heading towards evolution or degradation? *arXiv preprint arxiv: 2303.01255*, 2023a.

Martínez, G., Watson, L., Reviriego, P., Hernández, J. A., Juarez, M., and Sarkar, R. Towards understanding the interplay of generative artificial intelligence and the internet. *arXiv preprint arxiv: 2306.06130*, 2023b.

Mekala, D., Nguyen, A., and Shang, J. Smaller language models are capable of selecting instruction-tuning training data for larger language models. *arXiv preprint arXiv:2402.10430*, 2024.

Miao, Y., Gao, B., Quan, S., Lin, J., Zan, D., Liu, J., Yang, J., Liu, T., and Deng, Z. Aligning codellms with direct preference optimization, 2024. URL https://arxiv.org/abs/2410.18585.

Mindermann, S., Brauner, J., Razzak, M., Sharma, M., Kirsch, A., Xu, W., Höltgen, B., Gomez, A. N., Morisot, A., Farquhar, S., and Gal, Y. Prioritized training on points that are learnable, worth learning, and not yet learnt, 2022. URL https://arxiv.org/abs/2206.07137.

MistralAI. Codestral-22b-v0.1. https://huggingface.co/mistralai/Codestral-22B-v0.1, 2024a. Accessed: 2024-12-13.

MistralAI. Mistral-small-instruct-2409. https://huggingface.co/mistralai/Mistral-Small-Instruct-2409, 2024b. Accessed: 2024-12-13.

MistralAI. Codestral-22b-v0.1, 2024c. URL https://huggingface.co/mistralai/Codestral-22B-v0.1. Accessed: 2024-09-28.

Mobahi, H., Farajtabar, M., and Bartlett, P. L. Self-distillation amplifies regularization in hilbert space, 2020. URL https://arxiv.org/abs/2002.05715.

OLMo, T., Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., Lambert, N., Schwenk, D., Tafjord, O., Anderson, T., Atkinson, D., Brahman, F., Clark, C., Dasigi, P., Dziri, N., Guerquin, M., Ivison, H., Koh, P. W., Liu, J., Malik, S., Merrill, W., Miranda, L. J. V., Morrison, J., Murray, T., Nam, C., Pyatkin, V., Rangapur, A., Schmitz, M., Skjonsberg, S., Wadden, D., Wilhelm, C., Wilson, M., Zettlemoyer, L., Farhadi, A., Smith, N. A., and Hajishirzi, H. 2 olmo 2 furious, 2025. URL https://arxiv.org/abs/2501.00656.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Padmakumar, V. and He, H. Does writing with language models reduce content diversity? In *International Conference on Learning Representations (ICLR)*, 2024.

Pan, R., Zhang, J., Pan, X., Pi, R., Wang, X., and Zhang, T. Scalebio: Scalable bilevel optimization for llm data reweighting, 2024. URL https://arxiv.org/abs/2406.19976.

Parkar, R. S., Kim, J., Park, J. I., and Kang, D. Selectllm: Can llms select important instructions to annotate? *arXiv preprint arXiv:2401.16553*, 2024.

Paul, M., Ganguli, S., and Dziugaite, G. K. Deep learning on a data diet: Finding important examples early in training, 2023. URL https://arxiv.org/abs/2107.07075.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019. URL https://arxiv.org/abs/1910.00177.

Qin, Y., Yang, Y., Guo, P., Li, G., Shao, H., Shi, Y., Xu, Z., Gu, Y., Li, K., and Sun, X. Unleashing the power of data tsunami: A comprehensive survey on data assessment and selection for instruction tuning of language models, 2024. URL https://arxiv.org/abs/2408.02085.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model, 2023.

Rubin, O., Herzig, J., and Berant, J. Learning to retrieve prompts for in-context learning, 2022. URL https://arxiv.org/abs/2112.08633.

Saunshi, N., Malladi, S., and Arora, S. A mathematical exploration of why language models help solve downstream tasks, 2021. URL https://arxiv.org/abs/2010.03648.

Setlur, A., Garg, S., Geng, X., Garg, N., Smith, V., and Kumar, A. Rl on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold, 2024. URL https://arxiv.org/abs/2406.14532.

Shi, L., Dadashi, R., Chi, Y., Castro, P. S., and Geist, M. Offline reinforcement learning with on-policy q-function regularization, 2023. URL https://arxiv.org/abs/2307.13824.

Shumailov, I., Shumaylov, Z., Zhao, Y., Gal, Y., Papernot, N., and Anderson, R. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arxiv:2305.17493*, 2023.

Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R. J., and Gal, Y. Ai models collapse when trained on recursively generated data. *Nat.*, 631(8022):755–759, July 2024. URL https://doi.org/10.1038/s41586-024-07566-y.

Sun, Z., Shen, Y., Zhou, Q., Zhang, H., Chen, Z., Cox, D., Yang, Y., and Gan, C. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2023.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., and Wei, J. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Tajwar, F., Singh, A., Sharma, A., Rafailov, R., Schneider, J., Xie, T., Ermon, S., Finn, C., and Kumar, A. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*, 2024.

Tang, J. and Abbeel, P. On a connection between importance sampling and the likelihood ratio policy gradient. pp. 1000–1008, 01 2010.

Tang, Y., Guo, D. Z., Zheng, Z., Calandriello, D., Cao, Y., Tarassov, E., Munos, R., Pires, B. Á., Valko, M., Cheng, Y., et al. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*, 2024a.

Tang, Y., Guo, D. Z., Zheng, Z., Calandriello, D., Cao, Y., Tarassov, E., Munos, R., Ávila Pires, B., Valko, M., Cheng, Y., and Dabney, W. Understanding the performance gap between online and offline alignment algorithms, 2024b. URL https://arxiv.org/abs/2405.08448.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Team, O. T. Open Thoughts, January 2025.

Team, V. D. Vicuna llm: An open-source chatbot developed by fine-tuning the llama model on user-shared conversations, achieving performance comparable to other advanced chatbots. https://lmsys.org/blog/2023-03-30-vicuna/, 2023. Accessed: 2025-01-27.

Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023. URL https://huggingface.co/datasets/teknium/OpenHermes-2.5.

Wang, P., Li, L., Shao, Z., Xu, R. X., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. Math-shepherd: Verify and reinforce llms step-by-step without human annotations, 2024a.

Wang, P., Shen, Y., Guo, Z., Stallone, M., Kim, Y., Golland, P., and Panda, R. Diversity measurement and subset selection for instruction tuning datasets, 2024b. URL https://arxiv.org/abs/2402.02318.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, 2023. Association for Computational Linguistics.

Wang, Z., Novikov, A., Zolna, K., Springenberg, J. T., Reed, S., Shahriari, B., Siegel, N., Merel, J., Gulcehre, C., Heess, N., and de Freitas, N. Critic regularized regression, 2021. URL https://arxiv.org/abs/2006.15134.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Wei, Y., Wang, Z., Liu, J., Ding, Y., and Zhang, L. Magicoder: Source code is all you need. *arXiv preprint arXiv:2312.02120*, 2023.

Wu, B., Meng, F., and Chen, L. Curriculum learning with quality-driven data selection, 2024. URL https://arxiv.org/abs/2407.00102.

Xia, M., Artetxe, M., Zhou, C., Lin, X. V., Pasunuru, R., Chen, D., Zettlemoyer, L., and Stoyanov, V. Training trajectories of language models across scales, 2023. URL https://arxiv.org/abs/2212.09803.

Xia, M., Malladi, S., Gururangan, S., Arora, S., and Chen, D. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*, 2024.

Xiong, W., Dong, H., Ye, C., Wang, Z., Zhong, H., Ji, H., Jiang, N., and Zhang, T. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=c1AKcA6ry1.

Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., Lin, Q., and Jiang, D. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=CfXh93NDgH.

Xu, S., Fu, W., Gao, J., Ye, W., Liu, W., Mei, Z., Wang, G., Yu, C., and Wu, Y. Is dpo superior to ppo for llm alignment? a comprehensive study. In *ICML*, 2024b. URL https://openreview.net/forum?id=6XH8R7YrSk.

Xu, Y., Yao, Y., Huang, Y., Qi, M., Wang, M., Gu, B., and Sundaresan, N. Rethinking the instruction quality: Lift is what you need, 2023. URL https://arxiv.org/abs/2312.11508.

Xu, Z., Jiang, F., Niu, L., Deng, Y., Poovendran, R., Choi, Y., and Lin, B. Y. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*, 2024c.

Xu, Z., Jiang, F., Niu, L., Lin, B. Y., and Poovendran, R. Stronger models are not stronger teachers for instruction tuning, 2024d. URL https://arxiv.org/abs/2411.07133.

Yan, J., Li, Y., Hu, Z., Wang, Z., Cui, G., Qu, X., Cheng, Y., and Zhang, Y. Learning to reason under off-policy guidance, 2025. URL https://arxiv.org/abs/2504.14945.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., and Fan, Z. Qwen2 technical report, 2024a. URL https://arxiv.org/abs/2407.10671.

Yang, Y., Mishra, S., Chiang, J. N., and Mirzasoleiman, B. Smalltolarge (s2l): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models, 2024b. URL https://arxiv.org/abs/2403.07384.

Yang, Y., Mishra, S., Chiang, J. N., and Mirzasoleiman, B. Smalltolarge (s2l): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. *arXiv preprint arXiv:2403.07384*, 2024c.

Yang, Z., Pang, T., Feng, H., Wang, H., Chen, W., Zhu, M., and Liu, Q. Self-distillation bridges distribution gap in language model fine-tuning, 2024d. URL https://arxiv.org/abs/2402.13669.

Yin, J. O. and Rush, A. M. Compute-constrained data selection, 2024. URL https://arxiv.org/abs/2410.16208.

Yu, L., Jiang, W., Shi, H., YU, J., Liu, Z., Zhang, Y., Kwok, J., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=N8N0hgNDRt.

Yuan, H., Chen, Z., Ji, K., and Gu, Q. Self-play fine-tuning of diffusion models for text-to-image generation. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 73366–73398. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/860c1c657deafe09f64c013c2888bd7b-Paper-Conference.pdf.

Yuan, L., Cui, G., Wang, H., Ding, N., Wang, X., Deng, J., Shan, B., Chen, H., Xie, R., Lin, Y., Liu, Z., Zhou, B., Peng, H., Liu, Z., and Sun, M. Advancing llm reasoning generalists with preference trees, 2024b.

Yue, X., Qu, X., Zhang, G., Fu, Y., Huang, W., Sun, H., Su, Y., and Chen, W. Mammoth: Building math generalist models through hybrid instruction tuning, 2023. URL https://arxiv.org/abs/2309.05653.

Zeng, W., Xu, C., Zhao, Y., Lou, J.-G., and Chen, W. Automatic instruction evolving for large language models. *arXiv preprint arXiv:2406.00770*, 2024.

Zhang, D., Diao, S., Zou, X., and Peng, H. **PLUM**: Improving code lms with execution-guided on-policy preference learning driven by synthetic test cases, 2024a. URL https://arxiv.org/abs/2406.06887.

Zhang, J., Qin, Y., Pi, R., Zhang, W., Pan, R., and Zhang, T. Tagcos: Task-agnostic gradient clustered coreset selection for instruction tuning data, 2024b. URL https://arxiv.org/abs/2407.15235.

Zhang, S., Yu, D., Sharma, H., Zhong, H., Liu, Z., Yang, Z., Wang, S., Hassan, H., and Wang, Z. Self-exploring language models: Active preference elicitation for online alignment, 2024c. URL https://arxiv.org/abs/2405.19332.

Zhang, Y., Schwarzschild, A., Carlini, N., Kolter, J. Z., and Ippolito, D. Forcing diffuse distributions out of language models. In *First Conference on Language Modeling*, 2024d. URL https://openreview.net/forum?id=9JY1QLVFPZ.

Zhao, B., Mopuri, K. R., and Bilen, H. Dataset condensation with gradient matching, 2021. URL https://arxiv.org/abs/2006.05929.

Zhao, W., Ren, X., Hessel, J., Cardie, C., Choi, Y., and Deng, Y. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Zhao, Y., Yu, B., Hui, B., Yu, H., Huang, F., Li, Y., and Zhang, N. L. A preliminary study of the intrinsic relationship between complexity and alignment, 2024b. URL https://arxiv.org/abs/2308.05696.

Zheng, L., Chiang, W.-L., Sheng, Y., Li, T., Zhuang, S., Wu, Z., Zhuang, Y., Li, Z., Lin, Z., Xing, E., Gonzalez, J. E., Stoica, I., and Zhang, H. LMSYS-chat-1m: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=BOfDKxfwt0.

Zhong, Y., Liu, S., Chen, J., Hu, J., Zhu, Y., Liu, X., Jin, X., and Zhang, H. Distserve: Disaggregating prefill and decoding for goodput-optimized large language model serving, 2024. URL https://arxiv.org/abs/2401.09670.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., and Levy, O. Lima: Less is more for alignment, 2023. URL https://arxiv.org/abs/2305.11206.

Zhou, H., Liu, T., Ma, Q., Yuan, J., Liu, P., You, Y., and Yang, H. Gauging learnability in supervised fine-tuning data, 2024a. URL https://openreview.net/forum?id=KpC3dPumJj.

Zhou, H., Liu, T., Ma, Q., Zhang, Y., Yuan, J., Liu, P., You, Y., and Yang, H. Davir: Data selection via implicit reward for large language models, 2024b. URL https://arxiv.org/abs/2310.13008.

Zhou, W., Agrawal, R., Zhang, S., Indurthi, S. R., Zhao, S., Song, K., Xu, S., and Zhu, C. Wpo: Enhancing rlhf with weighted preference optimization, 2024c. URL https://arxiv.org/abs/2406.11827.

Zhou, Y., Xu, P., Liu, X., An, B., Ai, W., and Huang, F. Explore spurious correlations at the concept level in language models for text classification, 2024d. URL https://arxiv.org/abs/2311.08648.

Zhou, Z., Ning, X., Hong, K., Fu, T., Xu, J., Li, S., Lou, Y., Wang, L., Yuan, Z., Li, X., et al. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294*, 2024e.

Zhuang, Z., LEI, K., Liu, J., Wang, D., and Guo, Y. Behavior proximal policy optimization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3c13LptpIph.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See experiments in Section 4 and 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See section 5.6, where we show pursuing in-distribution answers in the wrong way can lead to performance degradations.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: We do not have theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: See appendices.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We use publicly available datasets and models. Our method only requires computing the normalized probability of training data, which can be easily done with any open-sourced machine learning codebase.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See Experiments Sections and Appendices.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We follow standard evaluation paradigms by computing pass@1 accuracy with greedy sampling, which does not explicitly involve randomness.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See appendicies, where we provide information about GPU resources.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: See paper.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification: This paper presents work whose goal is to advance the field of Machine Learning. It investigates fundamental aspects of instruction-tuning of language models and should not have direct societal impacts or implications that should be discussed here specifically, to the best of the authors' knowledge.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: We do not release data or models that have a high risk for misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Yes, we cite all the datasets and pretrained models used in the paper, which are all open-sourced for research use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: We do not have crowdsourcing experiments or research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: We do not have crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: We do not involve LLMs as a core component in implementing or developing the method of this work.

    Guidelines:

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

A Parameter Distance

We measure the L2 norm of model parameter difference between fine-tuned and pre-trained checkpoints, as a signal of how much the distribution has drifted during SFT ichi Amari (2016); Cover & Thomas (2006). We notice that training over well-matched distribution shifts the parameter less than training over those ill-matched.

	Mistral-7B-v0.3	**Llama3.1-8B**	**Qwen2.5**
GRAPE	8.006	8.196	8.426
Worst	8.029	8.202	8.467

Table 6: Performance comparison across different models

B Further Experiments

B.1 Comparing with Reward Based Selection

We compare GRAPE with purely reward-based selection. where for each instruction, we select the response with the highest scalar reward as determined by a reward model - Skywork-Reward-Llama3.1-8B-v0.2. Once the top-ranked response for each instruction is selected, we proceed with standard supervised fine-tuning on the resulting instruction-response pairs.The RFT setup provides a natural contrast to our proposed GRAPE method by emphasizing reward alignment over base-model alignment, thereby enabling us to disentangle the effects of distribution matching versus reward optimization in SFT data selection.

As shown in Table B.1, GRAPE outperforms reward-based selection across both models and all benchmarks. These results suggest that aligning supervision with the base model's own distributional preferences—rather than relying on external reward models—can yield better task performance.

Model	**Method**	**AE WR**	**AE WR (LC)**	**LeetCode**	**MATH**	**MMLU**	**BBH**	**Avg**
LLaMA3.1-8B	GRAPE	15.2	14.8	19.4	32.1	64.5	69.6	**35.9**
	Reward	14.0	14.5	17.2	31.3	63.3	69.0	34.9
Mistral-v0.3-7B	GRAPE	13.9	13.6	18.3	24.2	59.2	62.3	**31.9**
	Reward	12.5	13.9	13.3	22.4	58.5	62.2	30.5

Table 7: Performance comparison between GRAPE and reward-based selection across benchmarks for LLaMA3.1-8B and Mistral-v0.3-7B. Metrics are benchmark-specific scores (higher is better).

B.2 Experiment on OpenHermes

To test the generality of our findings beyond the UltraInteract and Tulu-Olmo settings, we conduct additional experiments on the OPENHERMES-2.5 (Teknium, 2023) dataset—a large-scale, high-quality instruction-tuning corpus with approximately 1 million distinct instructions.

Item	**Metric**	**Data**	**Llama 3.1-8B**	**Mistral -7B-v0.3**	**Qwen 2.5-7B**
Alpaca -Eval2	LC	Subset	8.6	5.9	7.6
		Random	8.0	6.2	9.0
		GRAPE	**11.3**	**8.2**	**10.8**
	WR	Subset	6.2	3.9	5.2
		Random	6.4	4.8	7.2
		GRAPE	**9.4**	**7.5**	**9.6**
Truthful -QA	MC2	Subset	51.6	49.0	54.4
		Random	51.4	49.9	55.2
		GRAPE	**52.7**	**51.6**	**56.4**

Table 8: Results on OpenHermes-2.5. The Subset row refers to training exclusively on the SFT responses over the subset.

Following the setup from §5, we apply GRAPE to select from responses aggregated across sources, including Huang et al. (2024) and HuggingFace-H4 (2024). For preference-based datasets, we retain only the winning responses to ensure quality, mirroring our earlier selection protocol. This results in 575K unique instructions and 1.34M instruction-response pairs.

As shown in Table 8, GRAPE continues to outperform naive combination strategies. The consistent gains across diverse data sources and model families strengthen our central claim: GRAPE is a general-purpose, model-aligned response selection strategy that reliably improves SFT performance in real-world, large-scale instruction tuning.

B.3 Data Selection For Long Chain-of-Thoughts

O1-/R1-style long chain-of-thoughts have drawn increasing attention. This paradigm, exemplified by models like OpenAI's O1 and DeepSeek-R1, has shown remarkable success in challenging domains such as mathematics and coding. We further experiment with the use of GRAPE in long chain-of-thought distillation. We generate multiple candidate trajectories using R1-Distill-Qwen-1.5B DeepSeek-AI et al. (2025) for a subset of OpenR1-Math (Face, 2025) dataset — LUFFY (Yan et al., 2025), and verify the correctness of each, retaining the correct ones.

We compare the results on lowest versus highest perplexity instances below in table 9.

Model	Perplexity	Acc.
Qwen2.5-1.5B	Highest	0.330
Qwen2.5-1.5B	Lowest (GRAPE)	0.396
Qwen2.5-3B	Highest	0.524
Qwen2.5-3B	Lowest (GRAPE)	0.544

Table 9: Performance metrics on MATH dataset

B.4 Token-level GRAPE

To further investigate the alternative uses of our insight, we conduct experiments beyond data selection by incorporating token-level likelihoods directly into the training objective. Specifically, we modified the loss function to weigh each token proportionally to its likelihood. We trained this variant on the OpenThoughts-114k Team (2025) dataset, a curated collection of 114k high-quality reasoning samples spanning domains such as math, science, code etc. Evaluation followed the same benchmark suite as LUFFY Yan et al. (2025), including competition-level math datasets (AIME24/25, AMC, MATH-500, Minerva, OlympiadBench) and general reasoning tests (ARC-c, GPQA-Diamond, MMLU-Pro). Our result in Table 10 show that our token-level likelihood-weighted training yields consistent improvements across several benchmarks.

Benchmark	Baseline (%)	Token-level GRAPE (%)
MATH	55.60	60.20
OLYMPIADBENCH	21.04	28.30
MINERVA	15.07	19.85
AIME-24	3.12	4.90
AMC	23.76	29.74
AIME-25	4.38	3.23
ARC-C	60.84	76.62
GPQA-DIAMOND	7.07	19.70
MMLU-PRO	27.02	34.35

Table 10: Qwen2.5-3B performance (in %) on various benchmarks under Baseline vs. Token-level GRAPE.

C Further Details On Baselines

This section details the experimental setup for our data selection baselines: **S2L**,**LESS** and **NV-Embed**.

C.1 S2L

S2L, a state-of-the-art unsupervised data selection baseline, operates through two key steps: training a reference model to capture training dynamics and clustering the resulting trajectories to form a

diverse, balanced subset of training data. The reference models used in our setup are specifically selected to enhance S2L's performance, adhering to the theoretical underpinnings from the original paper that training dynamics remain consistent across models of varying sizes within the same family.

For our experiments, we train small reference models corresponding to the final target models. Specifically, we pair Llama-3.1-8B with Llama-3.2-1B, Qwen-2.5-7B with Qwen-2.5-0.5B, and Mistral-v0.3-7B with itself due to the lack of smaller models in the Mistral family. To minimize computational costs, LoRA is applied when training the Mistral reference model. **This choice of reference models are better compared to original S2L setup**, which employed a Pythia-70M proxy, thereby improving the fidelity of the selected subset.

Following S2L, the reference models are trained on a random 5% subset of the dataset over four epochs. This reduced training requirement is justified by prior work, which demonstrates that only partial data is sufficient for the proxy model to learn meaningful training dynamics. During trajectory collection, we record the training loss of all examples at intervals of 500 iterations. The batch size and learning rate schedules are set as batch size of 128 and a learning rate warmup of 3%, followed by a cosine decay to 2e-5.

We then perform K-means clustering using the Faiss library to efficiently partition the trajectory space into 100 clusters. The number of iterations is set to 20, and we use the Euclidean distance metric to ensure convergence to well-separated clusters. From each cluster, an equal number of examples are sampled to maintain a balanced subset distribution.

C.2 LESS

LESS is a state-of-the-art model-based and supervised data selection method that leverages gradient-based influence estimation. Given a small set of validation examples per task, LESS computes the influence of each training example by measuring the weighted cosine similarity between their LoRA gradients across multiple warmup checkpoints. It then aggregates these influence scores by averaging over validation examples within each task, followed by taking the maximum across tasks to obtain a scalar utility score per training example. Training examples are selected greedily based on these scores. In our experiments, we use the same base model for both selection and training, and follow the original LESS setup: 5% warmup training for 4 epochs and a gradient projection dimension of 8192.

C.3 NV-Embed

Embedding-based data selection as detailed in (Ivison et al., 2025) is a supervised data selection method that ranks training examples by computing cosine similarity between their embeddings and those of validation examples. Unlike model-aware methods like LESS, embedding-based data selection is model-agnostic: it relies on fixed, pretrained embedding models (in our case, we used NV-embed-v2, the state-of-the-art embedding model) rather than the target model. Instead of aggregating similarity scores into a single utility value per training example, embedding-based data selection uses a round-robin strategy that iteratively selects the highest-scoring example for each validation instance, ensuring diverse coverage across tasks. We follow the original setup from Ivison et al. (2025) in our experiments.

D Further Training Details

We train our models on a 4-GPU Nvidia-GH200 node, with batch size 256 and micro batch size 2.

E Further Ablations on UltraInteract.

See Tables 11 and 4.

F Additional Related Works On Model Dependent Data Selection Approaches

Model-dependent data selection methods leverage internal signals from a target model—such as gradients, embeddings, or log-probabilities—to identify training examples that are most useful for

Data		Full UI	Closest-1	Random-1
Num. Instances		**280K**	**80K**	**80K**
HumanEval		46.3	42.1	41.5 (-)
LeetCode		15.6	13.9	11.1 (-)
MBPP		50.1	52.1	49.1 (-)
MATH	CoT	21.6	19.2	15.5 (-)
	PoT	32.6	24.9	15.1 (-)
GSMPlus	CoT	45.9	44.1	35.3 (-)
	CoT	45.3	43.2	45.2 (-)
TheoremQA	CoT	16.8	15.8	15.8
	PoT	20.1	12.9	15.3
Avg.		***32.7***	***29.8***	***27.1(-)***

Table 11: Ablations on data selection with MISTRAL-7B-V0.3 by selecting within UltraInteract-SFT (since it contains varying numbers of responses per-instruction). Closest-1 denotes the one closest to the base model's initial distribution. Random-1 is sampled from the entire enlarged dataset formed by both original and generated responses. We use (-) to denote **Random-1** underperforming **Closest-1**.

Model	Data	HE	LC	MBPP	MATH		GSMPlus		TheoremQA		Avg.
					CoT	PoT	CoT	PoT	CoT	PoT	
Mistral-7B	Self-Distill	46.3	13.3	49.6	17.3	18.5	43.3	33.2	16.8	17.4	28.4
	Original-UI	46.3	15.6	50.1	21.6	32.6	45.9	45.3	16.8	20.1	32.7
	Ours	52.4	15.6	53.4	28.9	34.6	50.5	52.8	17.8	20.6	36.3
Llama3.1-8B	Self-Distilled	47.6	6.7	51.7	22.9	12.7	47.2	35.3	18.8	21.5	29.4
	Original-UI	54.3	11.1	58.9	29.7	31.0	53.7	51.6	20.0	20.8	36.8
	Ours	57.3	19.4	63.8	34.8	39.2	56.6	56.1	22.5	23.9	41.5
Llama3.2-3B	Self-Distilled	32.3	5.6	41.9	8.8	7.0	12.1	12.1	5.9	10.5	15.1
	Original-UI	32.9	3.9	41.6	12.8	16.1	30.8	19.5	14.6	10.5	20.3
	Ours	42.6	13.3	44.6	16.4	17.6	34.9	20.6	15.1	11.4	24.1

Table 12: The detailed comparison across benchmarks for self-distillation discussed in Section 5.5

fine-tuning. These approaches have led to strong empirical results across various settings. However, many of them involve substantial computational costs, such as repeated gradient computations or auxiliary model training, which can limit their scalability. We discuss these approaches and the costs they incur in this section.

F.1 Notations

1. A training dataset $D = \{x_i\}_{i=1}^{N}$ of size N; the final language model to be trained on the selected data θ.
2. We denote **the average cost of one forward pass** of model θ on a training example as F_θ. As one backward pass is approximately the cost of two forward passes, **the average cost of one "gradient pass"** (i.e., one forward + one backward) is thus $3F_\theta$.
3. Another important source of computational cost in data selection comes from the training of additional models. We use $C(\theta, D, T)$ to denote the cost of training model θ on dataset D for T epochs (i.e., $N \cdot T$ examples are seen in total).
4. Therefore, we unify the computational cost of most data selection approaches into two parts:
   (a) **The training of additional models**. For example, gradient-based influence requires training an additional model on part of the training dataset for T epochs to obtain the checkpoints for gradient computation.
   (b) **The computation of per-sample features**. For example, for each training example, gradient-based influence requires computing its gradient for each saved checkpoint, which means T gradient passes are needed.
5. Note that some algorithms may have additional computational costs other than the two parts above, such as clustering or a greedy algorithm for the final data selection. Since the two parts above constitute the majority of computation for almost all the data selection approaches, **we omit the other cost and only focus on these two.**

F.2 TLDR: The Final Table

For GRAPE, we assume that in the training dataset D, various responses to the same instruction are already available, thus no additional cost is incurred in the *Response Collection* step of GRAPE. So the computational cost analysis of GRAPE under our framework is:

- **Additional Training**: 0, as GRAPE directly evaluates data using the base model.
- **Per-sample conditional probability**: NF_θ, as for a given target model θ, we only need to compute conditional probability for each response (example) once.

The table below shows that our method, GRAPE, achieves superior performance with minimal computational cost compared with other model-based data selection approaches.

	Additional Training	**Per-Sample Feature Computation**
GRAPE (ours)	0	NF_θ
Gradient-based influence (LESS)	$C(\theta_{\text{lora}}, D_{\text{warmup}}, T)$	$3T \cdot NF_\theta$
In-run gradient-based influence	$C(\theta, D, 1)$	0
Gradient matching	$C(\theta_{\text{lora}}, D_{\text{warmup}}, T)$	$3T \cdot NF_\theta$
Gradient norm	$m \cdot C(\theta, D, 1)$	$3m \cdot NF_\theta$
Embedding-based	0	NF_θ
Simple uncertainty indicators	0	NF_θ
Perplexity	$C(\theta_{\text{ref}}, D_{\text{ref}}, 1)$	$NF_{\theta_{\text{ref}}}$
Learnability	$C(\theta, D, 1)$	$2 \cdot NF_\theta$
Loss trajectory (S2L)	$C(\theta_{\text{ref}}, D, T)$	$T \cdot NF_{\theta_{\text{ref}}}$

Table 13: Computational cost comparison of data selection methods.

F.3 Gradient-based Methods

Gradients have long been an important source of information for training data selection, as they directly affect the whole optimization process of language models. Three kinds of model-based gradient-based data selection approaches have been proposed:

1. Gradient-based influence
2. Gradient matching
3. Gradient norm

F.3.1 Gradient-based Influence

Gradient-based influence computes the pairwise influence scores between each pair of training and validation examples. Training data with the highest influence are selected, as training on them leads to the theoretically largest decrease in model loss on validation data. LESS Xia et al. (2024) formulates the pairwise influence scores as the cosine similarity between the gradients of training and validation data, and computes these gradient features using the following two steps:

1. LoRA-train the final model on part of the whole training dataset, denoted as D_{warmup}, for T epochs, and save the T model checkpoints.
2. For each data point, compute its LoRA gradient with each of the T checkpoints, and later aggregate these T gradients together in the cosine similarity expression.

Therefore, the computational cost of gradient-based influence is:

- **Additional training**: $C(\theta_{\text{lora}}, D_{\text{warmup}}, T)$.
- **Per-sample gradient for each checkpoint**: $NT \cdot 3F_\theta = 3T \cdot NF_\theta$.

In order to reduce the cost incurred by per-sample gradient computation, recent work has developed *in-run gradient-based influence* that directly computes the dot product between gradients without

the need for separate gradient computations. However, this approach incorporates the dot product computations into the standard training process, which means in order to obtain pairwise influence scores for **the whole training set**, **a full training run** has to be done on all the training data. This incurs inefficiency when we do not actually need full dataset training. Moreover, the pairwise scores here only show the model's **"dynamic preference"**: scores computed at the t-th iteration only reflect the model's preference at this specific iteration. It is not theoretically guaranteed that these scores reflect the model's preference from the beginning of training. Thus, the cost of in-run gradient-based influence is:

- **Additional Training**: $C(\theta, D, 1)$.
- **Per-sample gradient**: 0.

F.3.2 Gradient Matching

Gradient matching also requires per-sample gradients, but utilizes their information in a different way. It performs clustering based on these gradient features to group similar data, and then applies an iterative greedy selection algorithm. **In order to scale to LLM-level gradient computation and clustering, TAGCOS Zhang et al. (2024b) completely follows the warmup training and gradient computation pipeline of LESS Xia et al. (2024).** As the computational bottleneck here is still the gradient computation instead of clustering or iterative selection, Zhang et al. (2024b) also shares the same computational cost as Xia et al. (2024):

- **Additional training**: $C(\theta_{\text{lora}}, D_{\text{warmup}}, T)$.
- **Per-sample gradient for each checkpoint**: $NT \cdot 3F_\theta = 3T \cdot NF_\theta$.

F.3.3 Gradient Norm

The L_2-norms of gradient vectors can also serve as effective indicators for data selection. Paul et al. (2023) proposes GraNd, which obtains a utility score for each training point based on its gradient norm early in the training. More specifically, it starts from m different model weight initializations, trains each model on the whole dataset to obtain per-sample gradient norms, and finally averages the m gradient norms for each training point to obtain the final GraNd score. Therefore, the computational cost of GraNd is shown below:

- **Additional training**: $m \cdot C(\theta, D, 1)$.
- **Per-sample gradient for each weight initialization**: $Nm \cdot 3F_\theta = 3m \cdot NF_\theta$.

F.4 Embedding-based Methods

Embedding-based methods project the whole training set into an embedding space to quantify the information of each data point and their interactions. For model-based embedding-based selection methods, the embeddings are usually computed by the final model θ to align with its preference.

Under a supervised data selection setup where validation data representing target task distributions are available, **Representation-based Data Selection** (RDS; Rubin et al. (2022); Hanawa et al. (2021)) computes the embedding similarity between training and validation data, and selects training points that are most similar to the target distribution in the embedding space.

For an unsupervised setup where only the embeddings of training data are accessible, **geometry-based coreset sampling** methods are widely used Qin et al. (2024). Grounded on the intuition that close samples in the embedding space often share similar properties, a **diverse** subset can be obtained by controlling the minimum distance between any two selected data points. Among them, using K-center greedy sampling to select embedding-based **facility locations** has been proven especially effective for instruction fine-tuning of LLMs Bhatt et al. (2024).

These embedding-based approaches share similar computational costs: they do not need any additional model training and can directly extract useful per-sample embeddings using the last-layer hidden states of the pretrained final model θ. Thus, their computational cost is shown below:

- **Additional training**: 0.
- **Per-sample embedding computation**: NF_θ.

F.5 LogProb-based Methods

LogProb-based methods also directly utilize the target LLM to evaluate the utility of each training data point.

4.1 Simple Uncertainty-based Indicators

Some simple model-based indicators inspired by the notion of uncertainty have been shown effective for a long time and recently extended to data selection for LLM instruction tuning (Marion et al., 2023b; Bhatt et al., 2024). Bhatt et al. (2024) demonstrates the effectiveness of various indicators including **mean entropy, least confidence, mean margin**, etc. These simple indicators do not require additional training and can also be directly obtained with the pretrained final model θ. Their computational cost is shown below:

- **Additional training**: 0.
- **Per-sample per-token logits computation**: NF_θ.

F.5.1 Perplexity (PPL)

PPL is also a long-standing data selector and has been shown effective for LLM-scale data selection. Typically, a split of the training dataset, D_{ref}, is needed to train θ_{ref}, a reference model that will be used to compute PPL for the whole training set.

A common approach is to use the final model θ as the reference model θ_{ref} to ensure the alignment in PPL patterns Marion et al. (2023a), but prior work Ankner et al. (2024) also shows that a reference model much smaller than the final model can also be an effective PPL-based data selector. The computational cost for PPL-based selection is shown below:

- **Additional training**: $C(\theta_{\text{ref}}, D_{\text{ref}}, 1)$.
- **Per-sample PPL computation**: $NF_{\theta_{\text{ref}}}$.

F.5.2 Learnability

In addition, **learnability (Mindermann et al., 2022; Zhou et al., 2024a,b)** is a more effective metric than pure uncertainty or PPL, as it excludes uncertain but unlearnable points (e.g., noisy or less task-relevant) by considering the decrease in per-sample loss before and after the model is fully trained. More specifically, it trains the final model θ on the full training dataset to obtain a strong reference model θ_{ref}, and then computes the difference of loss on each training example between θ and θ_{ref}. In this way, it requires two forward passes for per-sample computation:

- **Additional training**: $C(\theta, D, 1)$.
- **Per-sample learnability computation**: $2 \cdot NF_\theta$.

F.5.3 Loss Trajectory

Moreover, logprob-based methods can also obtain finer-grained information from the **training dynamics** of LLMs. S2L Yang et al. (2024b) obtains a feature vector for **each** training point by collecting their **training loss trajectories** over T-epoch training on a small reference model θ_{ref}, and then applies K-means clustering to equally sample data points from each trajectory cluster. Prior work shows its superiority over other logprob-based indicators, but it also comes with significant computational cost:

- **Additional training**: $C(\theta_{\text{ref}}, D, T)$. Here the choice of θ_{ref} is especially important, as prior work Xia et al. (2023) shows that reference models that come from the same model family as the final model tend to have similar loss trajectories of training data, so they can preserve more fidelity in their loss trajectory patterns.
- **Per-sample loss trajectory computation**: $T \cdot NF_{\theta_{\text{ref}}}$. Note that T here is typically much larger than that in gradient-based influence computation, so the computational cost of this gradient-free approach can be even higher than gradient-based methods.

G Multi-Round GRAPE

To evaluate whether GRAPE continues to provide benefits after an initial round of fine-tuning, we conducted a second-round experiment using LLAMA3.1-8B. In the first round, we selected the best responses per instruction using GRAPE and fine-tuned the model accordingly. Then, we re-applied GRAPE on the newly fine-tuned model to select a fresh set of responses and conducted another round of fine-tuning. This second iteration led to further performance improvements across multiple benchmarks, including AlpacaEval, WizardEval (both original and LeetCode), and real-world tasks such as MATH, MMLU, and BBH. Notably, GRAPE Round 2 improved average benchmark scores from 35.9% to 37.3%, demonstrating that the method remains effective and even compounding when iteratively applied.

Method	AE	WR	WR (LC)	MATH	MMLU	BBH
GRAPE Round 1	15.2	14.8	19.4	32.1	64.5	69.6
GRAPE Round 2	**17.6**	**19.6**	18.9	**33.2**	64.5	**70.0**

Table 14: Performance of LLAMA3.1-8B after two rounds of GRAPE fine-tuning.

H Details Of Correlation Analysis

To analyze how well different models align with the training distribution, we conducted a perplexity-based study on the Tulu-v3 training set. Specifically, we randomly sampled 1,000 instances from the training data and used a wide array of generator models to produce responses for each instruction. For each response, we computed its perplexity under each base model and ranked the responses by perplexity, with lower perplexity indicating better matching of distribution. We then identified the top-1 response per instance for each base model and visualized the overall rankings using a heatmap (Figure 4).