

Balancing the Budget: Understanding Trade-offs Between Supervised and Preference-Based Finetuning

Mohit Raghavendra, Junmo Kang, Alan Ritter

Georgia Institute of Technology

{mraghavendra6, junmo.kang, aritter34}@gatech.edu

Abstract

Post-training of Large Language Models often involves a pipeline of Supervised Finetuning (SFT) followed by Preference Finetuning (PFT) using methods like Direct Preference Optimization. Both stages require annotated data that are very different in structure and costs. We study how to optimally allocate a fixed training data budget between the two stages, through extensive experiments spanning four diverse tasks, multiple model sizes and various data annotation costs. Our findings reveal that just SFT on the base model dominates performance in low-data regimes ($< 1,000$ annotated examples). With larger data-budgets, we observe that a combination of SFT and PFT, often with increasing portions allocated towards preference data yields optimal performance. However, completely eliminating SFT and running PFT directly on the base model yields sub-optimal performance, described as the cold start problem on tasks like mathematics. We observe that this is due to the distribution shift arising from using DPO directly on the base model to elicit step-by-step reasoning. This limitation can be effectively addressed by allocating even a small portion ($< 10\%$) of the budget to SFT first, resulting in performance improvements of $15 - 20\%$ on analytical benchmarks like GSM8k. These results provide actionable insights for researchers and practitioners optimizing model development under budget constraints, where high-quality data curation often represents a significant portion of the total costs of model development.

1 Introduction

The effectiveness of Large Language Models at instruction following, reasoning, safety, and many other capabilities is largely driven by post-training after an initial pre-training phase (Achiam et al., 2023; Team et al., 2023; Chung et al., 2024). Two prominent approaches have emerged: Supervised Fine-Tuning (SFT), which uses direct instruction-

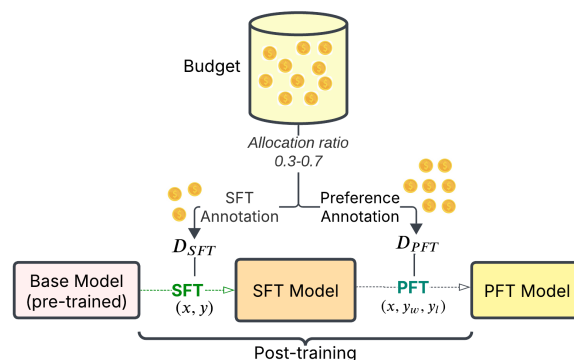


Figure 1: An illustration of the choices that introduce the data-allocation trade-off in LLM post-training. Given a fixed limited budget, one has to decide how much to allocate for annotating SFT data and how much for preference annotation (PFT data).

response pairs, and Reinforcement Learning (RL) based methods like RLHF (Christiano et al., 2017; Wei et al., 2022; Bai et al., 2022b,c). SFT and RLHF are usually applied sequentially after pre-training, as part of the post-training pipeline for building many state-of-the-art models (Dubey et al., 2024; Lambert et al., 2025). However, recent work in alignment methods (Liu et al., 2024b; Ethayarajh et al., 2024a) show that models can be aligned for some tasks without extensive SFT. Recent reasoning models (Guo et al., 2025) have also improved a pre-trained model’s analytical abilities with RL without SFT, raising questions about the necessity, role and extent of SFT in the post-training pipeline. Hence, the optimal allocation of limited training resources between SFT and RL-based approaches in post-training is an open question. Many RL methods involve training a separate reward model to iteratively generate preferences online (Christiano et al., 2017, 2023). However, approaches based on preference finetuning (PFT) (Rafailov et al., 2024; Ethayarajh et al., 2024b) can be used with offline preference-annotated data without training a reward model, while being competitive in perfor-

mance. (Lambert et al., 2025; Allal et al., 2025). Understanding how these methods perform under resource constraints is crucial for researchers and practitioners seeking to maximize their performance under strict finetuning budgets. In particular, we focus on measuring their effectiveness under fixed data-annotation budgets, since the cost of collecting and annotating human data is substantial as compared to the compute costs of efficiently finetuning models on the collected data. See appendix A.3 for an estimate of this from our experiments. Efforts aimed at data collection for SFT and PFT methods also present an interesting practical trade-off. SFT typically requires expert demonstrations, which can be expensive and time-consuming to collect but provide a direct signal of the desired model behavior. PFT requires preference annotated data that consists of comparative judgments, which may be easier to obtain but potentially offer less direct supervision (Bai et al., 2022a). We illustrate this decision-making process in Figure 1, framing it through the lens of consumer choice theory, which examines how individuals maximize utility by selecting an optimal combination of options within resource constraints (Becker, 1965; Lancaster, 1966). Based on this, we formulate three research questions to understand this trade-off:

1. *For a given data-budget, what ratio should we allocate towards SFT and PFT?*
2. *Is SFT a necessary precursor to PFT?*
3. *How would the optimal allocation change under different costs of annotating SFT and PFT data?*

We examine four tasks commonly used to evaluate finetuning methods in prior works: Summarization, Helpfulness, Instruction Following, and Grade School Math. Our analysis spans training budgets from 100 to 20,000 examples per task, with different allocation ratios between supervised and preference data, over multiple model families and sizes, to examine the Pareto-optimal frontier of performance versus the budget. This comprehensive evaluation allows us to identify optimal strategies for allocating resources between these fine-tuning approaches. Key takeaways from our extensive analysis (which includes over 1,000 unique finetuning runs) across different tasks, model sizes and training data budgets:

- SFT demonstrates superior performance in low-data regimes (under 1,000 examples), while the benefits of combining it with preference tuning become more apparent in large-data regimes.
- Across various configurations of model sizes, budgets and tasks, we find that devoting a higher ratio of the budget towards preference data (e.g., 3:1) with respect to SFT data yields the best performance while being cost effective.
- We also analyze the cold-start problem while running preference finetuning directly on the base model, and find that the presence of even minimal SFT data (100-1,000 examples) is highly beneficial.

These findings have important implications for researchers and practitioners developing LLMs under budget constraints, suggesting that while devoting most resources to preference finetuning is quite cost effective, a hybrid approach with a small amount of high-quality SFT data complemented by larger amounts of preference data may be optimal.

2 Experimental Design

In this section, we formulate our problem of studying resource allocation under a fixed budget (described in Section 2.1), using the two methods detailed in Sections 2.2 and 2.3, using the experimental setup from Section 2.4.

2.1 Problem formulation

The primary focus of this study is to understand and evaluate two finetuning stages SFT and PFT used in the post-training of text-only LLMs, in terms of their data-annotation costs. Given a fixed budget, the goal is to identify which method is cost-effective, and in what ratio should the resources be allocated between the two methods. We limit our focus to models under 10 Billion parameters and leave the study of larger models, which might incur compute-data annotation cost trade-offs (Bai et al., 2021) for future work. We also assume that the set of prompts for the task are available, and the annotation process involves generating responses for them. Other resources like researcher/engineer labor costs for model training are assumed to be equal for both methods, for simplicity.

Data budget is a common variable along which we compare post-training methods. This can be

Category	SFT	PFT
Summarization	Reddit TL/DR (Völske et al., 2017)	Reddit comparison dataset (Stiennon et al., 2020)
Helpfulness	HelpSteer (Wang et al., 2023b) HelpSteer (Wang et al., 2023b)	HelpSteer2 (Wang et al., 2024) HelpSteer2 (Wang et al., 2024)
Instruction Following	Tülu3 Persona IF (Lambert et al., 2025)	Tülu3 Persona IF (Lambert et al., 2025)
Grade School Math	GSM8k (Cobbe et al., 2021) [↓]	Tülu3 Grade School Math (Lambert et al., 2025) [↓]

Table 1: Overview of the task-specific datasets used. Datasets marked with [↓] use prompts from the original datasets, but the annotations are synthetically created by us. Refer to appendix A.2 for more information on how the datasets were curated and processed.

Task	Benchmark
Summarization	Win Rate vs Reference
Helpfulness	Win Rate vs Reference
Instruction Following	IFEval Instruction Accuracy (Zhou et al., 2023)
Grade School Math	GSM8k Test Accuracy

Table 2: Overview of the benchmarks and evaluation metrics for the tasks.

measured in terms of the number of training examples or the monetary cost of annotating the dataset. We assume that no task-specific labeled data is available at the start, and simulate the creation of SFT and PFT annotations using open-source datasets for each target task/skill when available, or synthetically curating data ourselves. Since we intend to study improvements on a task using targeted data annotation, we leave out task-agnostic general-purpose conversational datasets that are commonly used for Preference Finetuning, like UltraFeedback (Cui et al., 2023) and Chatbot Arena preferences (Chiang et al., 2024).

2.2 Supervised Finetuning

The loss function for SFT is defined as:

$$\mathcal{L}_{\text{SFT}}(\pi) = -\mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{SFT}}} [\log \pi(y|x)]$$

where $\mathcal{L}_{\text{SFT}}(\pi)$ is parameterized by the model as π (the model’s parameters), and $\pi(y|x)$ is the probability of generating the response y by the model π given the prompt x , where (x, y) is a prompt-response pair sampled from the SFT dataset \mathcal{D}_{SFT} .

2.3 Preference Finetuning

For RLHF, we use offline Preference Finetuning (PFT) methods, as this simplifies our analysis without the need for iterative training or reward models. This also allows for a more controlled comparison of various methods, as the data used for post-

training is exactly the same in each fine-tuning run. We use Direct Preference Optimization (DPO) (Rafailov et al., 2024) in our main experiments, which has been used extensively in building state-of-the-art LLMs (Dubey et al., 2024; Lambert et al., 2025). The objective of DPO is defined as:

$$\mathcal{L}_{\text{DPO}}(\pi_{\theta}; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}_{\text{PFT}}} [\log \sigma(\beta \Delta)]$$

$$\Delta = \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$$

$\mathcal{L}_{\text{DPO}}(\pi_{\theta}; \pi_{\text{ref}})$ is the DPO objective, π_{θ} is the model policy π_{θ} , π_{ref} is the reference policy, β is a scalar parameter controlling the strength of the preference modeling and (x, y_w, y_l) is a triplet of prompt, chosen response and rejected response sampled from an offline preference dataset \mathcal{D}_{PFT} . We also include experiments using the KTO objective by Ethayarajh et al. (2024b) which is defined as follows:

$$\mathcal{L}_{\text{KTO}}(\pi_{\theta}; \pi_{\text{ref}}) = \mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{FB}}} [\lambda_y - v(x, y)]$$

$$v(x, y) = \begin{cases} \lambda_D \sigma(\beta [r_{\theta}(x, y) - z_0]), & y \in y_w, \\ \lambda_U \sigma(\beta [z_0 - r_{\theta}(x, y)]), & y \in y_l, \end{cases}$$

$$r_{\theta}(x, y) = \log \frac{\pi_{\theta}(y | x)}{\pi_{\text{ref}}(y | x)},$$

$$z_0 = \text{KL}(\pi_{\theta}(\cdot | x) \parallel \pi_{\text{ref}}(\cdot | x)),$$

where λ_U and λ_D are hyperparameters.

2.4 Models and Post-Training Data

We primarily finetune the Llama3.1 8B and Qwen2.5-7B model families for our experiments. Both SFT and PFT are run for 2 epochs each. Since the study involves hundreds of finetuning runs, we use LoRA (Hu et al., 2021) to accelerate finetuning.

Additional hyperparameter details are in appendix A.4. We consider four tasks that represent a diverse variety of LLM post-training goals - *Helpfulness*, *Summarization*, *Instruction Following*, and *Grade School Mathematics*. These test models for capabilities like elementary reasoning, precise response generation, and alignment to human preferences. Table 1 and 2 describe the datasets used in finetuning for each task and the evaluation metrics considered. Since human-annotated data and synthetically generated data could have significant differences in quality and content, we ensured that when we compared models finetuned using the two methods on a task, the training data used by both methods were either both human-generated or both synthetically generated data. More details about dataset curation, processing and the conversational chat template used are in appendix A.2.

3 Analysis of SFT and PFT

3.1 *For a given data-budget, what ratio should we allocate towards SFT and PFT?*

In this section, we study the following challenge that an NLP researcher or practitioner would face - to improve performance on a *target task or skill*, given a fixed budget to spend on post-training data annotation, what would be an optimal budget allocation between SFT and PFT. Prior work has shown that improvements by scaling up SFT data follows a power law, indicating that training data requirements increase drastically for subsequent performance gains through SFT (Raghavendra et al., 2024). We study how this changes when we shift some of the data annotation efforts into Preference Finetuning. To eliminate the unobserved effect of multitask finetuning, we simulate annotating, finetuning, and evaluating the improvements for each skill separately.

Given a pretrained model and a fixed data-budget, our objective is to finetune using a pipeline of SFT followed by PFT, and empirically determine the optimal performance while scaling the post-training data size. We increase data-budgets from 0 to 20,000 instances, covering over four orders of magnitude. We experimented with SFT data allocation ratios $\in \{1.0, 0.75, 0.5, 0.25, 0.0\}$, with the rest allocated for preference data annotation. For instance, for a data budget of 10,000 if the SFT ratio is 0.25, then the SFT and PFT training data have 2,500 and 7,500 examples each. Note that an SFT ratio of 1.0 and 0.0 denotes pure SFT and pure

PFT on the base model respectively. By measuring data budgets and allocation in terms of number of examples, this section also implicitly assumes that SFT and PFT data have equal data annotation costs. This is because data annotation costs vary substantially - the price of human annotation used in the training data of leading LLMs is usually not publicly disclosed, and synthetic annotation uses LLM API whose prices change significantly over time.¹ So, defining the data budget in terms of the number of training examples makes the results agnostic to the exact dollar cost at any point in time. In section 3.3, we relax this equal cost assumption and investigate the sensitivity of the results under different SFT-PFT data annotation costs using publicly available information in the literature as well as our estimated annotation costs.

We show results for Llama3.1 8B and Qwen2.5 7B in Figure 2 using DPO as the Preference Finetuning method. We also show results using an alternate PFT method of Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024b) in Figure 3. Additional results to test for generality, using other model sizes like Llama3.2 {3B, 1B} and Qwen2.5 {3B, 1.5B} are in Figure 9 and 10 in appendix A.1.

Our experiments reveal key insights about the interplay between SFT and PFT across different data regimes. In several cases, finetuning on a dataset with optimal SFT-PFT budget allocation ratio performed better than finetuning with **2-5X larger** suboptimally allocated training data budget. For instance, finetuning with 5000 examples with a 25% SFT allocation is on par with training with 20,000 examples with 75% SFT allocation on Summarization, Helpfulness and Grade School Math. However, there are variations in the relative performance gains from PFT over SFT. The performance improvement from PFT is outsized on tasks like Summarization and Helpfulness Alignment and more modest in tasks like Mathematics. We believe that future work could explore this further, to understand if the difference in relative improvement is a result of the training data, nature of the task, evaluation metrics or the offline-online RL setting.

Takeaway: *In low-data scenarios ($n < 1000$), pure SFT demonstrates superior performance compared to other mixed allocation approaches. As the data budget increases, we observe a consistent*

¹<https://www.nebuly.com/blog/openai-gpt-4-api-pricing>

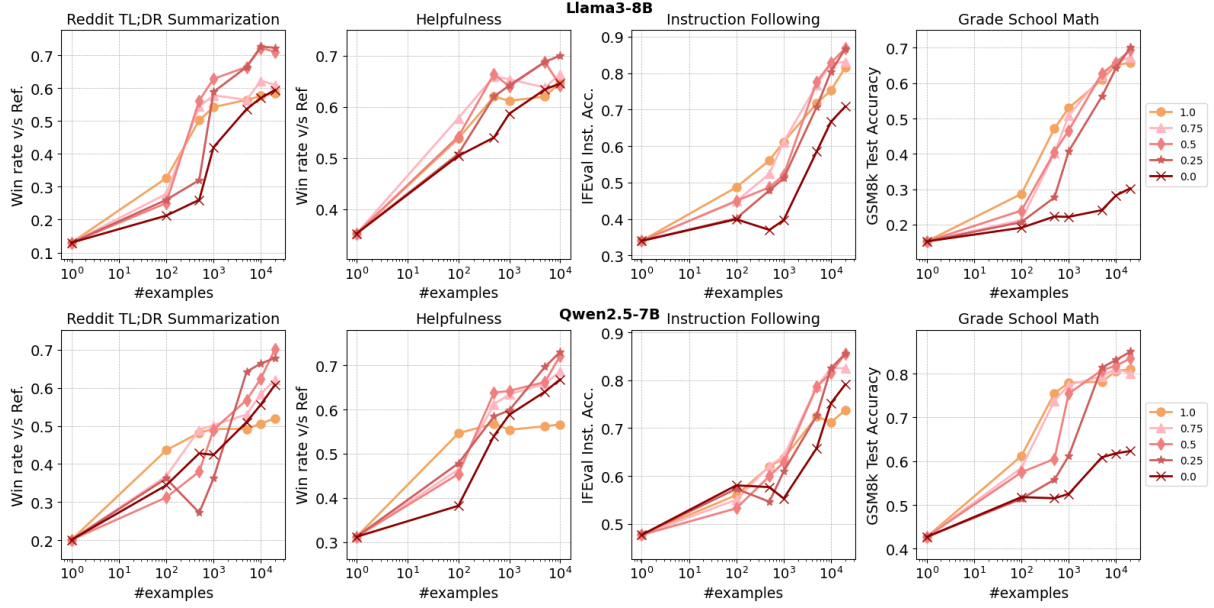


Figure 2: Effect of varying the of SFT-PFT data mix on the performance of Llama3.1-8B (top) and Qwen2.5-7B (bottom) base models. The x-axis represents the number of training examples (data budget), and the y-axis represents performance, measured using different task-specific metrics. The ratios represent the fraction of the training data allocated for SFT, and the rest is for Preference Finetuning. The orange line shows the performance when trained using only SFT data (1.0 ratio). The subsequent darkening red-shaded lines indicate decreasing proportions of SFT data in the training set, all the way till using only PFT data directly on the base model (0.0 ratio).

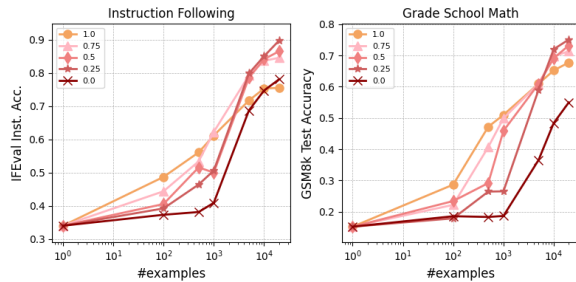


Figure 3: Comparison of SFT against KTO method of PFT. We notice similar relative performance scaling patterns as SFT vs DPO highlighting the general trends in preference data-based finetuning against SFT.

trend of higher proportions of preference data in the training set leading to improved performance. (Figure 2, 3)

3.2 Is SFT a necessary precursor to PFT?

In the previous section, we observed that lowering the proportion of SFT gave better results. However, the complete absence of SFT data led to suboptimal performance. In Figure 4, we study how model performance scales with training data size when PFT is run directly on the base model across model sizes, and contrast it with SFT on the base model. We observe that SFT on the base model shows a consistent improvement on the task across

all model sizes with more data, unlike Preference Finetuning.

Takeaway: *Directly applying preference finetuning on the base model shows little or no improvements on smaller models across all tasks. As model sizes grow, stylistic tasks like Instruction Following and TL;DR Summarization improve eventually with more data, but mathematics only shows modest improvements. (Figure 4)*

This hints at the *cold-start problem* in applying preference-based RL methods directly on the base models for tasks like math, similar to what was recently observed in other RL methods like GRPO in Shao et al. (2024); Guo et al. (2025). To study this further, we finetune Llama3.1-8B model with 10,000 examples on Instruction Following and Math data, with diminishing SFT ratios of 0.1 (1000 SFT examples), 0.01 (100 SFT examples), and 0.0 (0 SFT examples), and try both DPO and KTO on these SFT checkpoints. The results from this experiment in Figure 5 show the effects of diminishing SFT data on PFT results.

We see that allocating even a minimal amount of budget to SFT data in tasks like math (as low as < 10%) provides a substantial performance boost, whereas the improvements in stylistic tasks are more gradual.

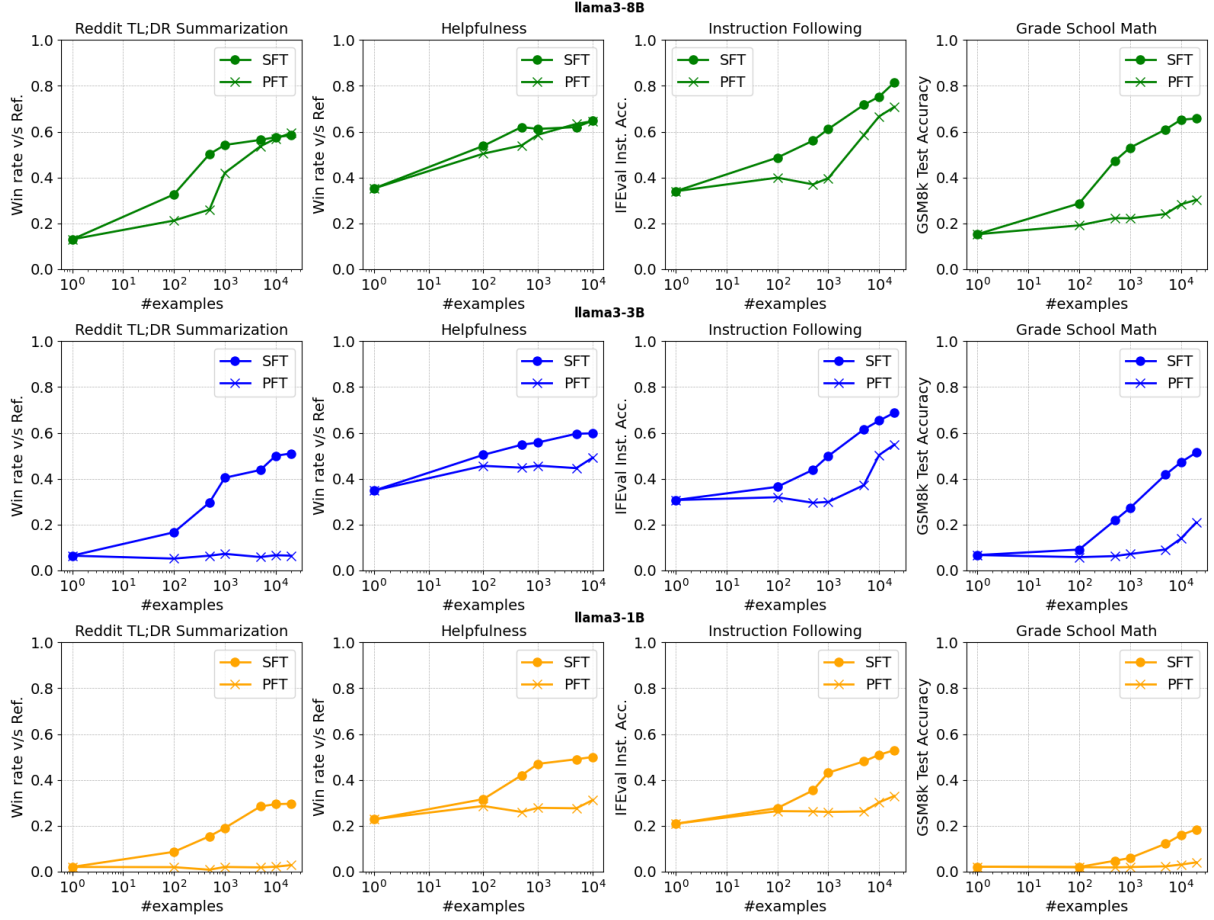


Figure 4: Scaling patterns of SFT and PFT (using DPO) directly on the Llama3 models - 8B(**top**), 3B (**middle**) and 1B(**bottom**). We observe that SFT shows a consistent improvement on the task across all model sizes. However, directly applying PFT shows improvements only in large data-regimes, and only in larger model sizes.

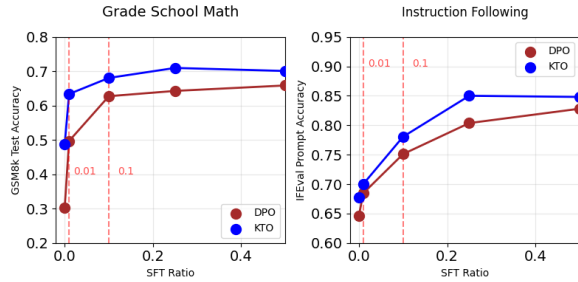


Figure 5: Performance of DPO and KTO models with decreasing SFT data ratio of 0.1, 0.01 (highlighted in red dotted lines) and 0 (Pure PFT), for the same total data budget. We see that even a minimal amount of SFT can have outsized benefits in both cases, with the improvements being more drastic in analytical tasks like math compared to gradual improvements in stylistic tasks like instruction following.

We also analyze this cold-start phenomenon qualitatively by manually annotating and studying a random subset of 50 responses from the GSM8k test set. We check for correct answer, whether the

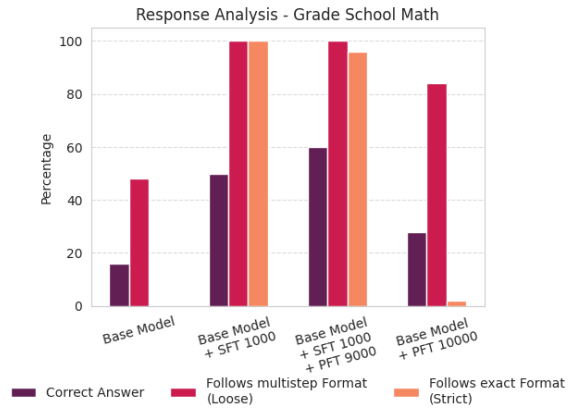


Figure 6: Manual analysis of model responses fine-tuned with "cold-start" SFT data, on the GSM8k test set. Directly performing PFT even with 10,000 examples, although encourages some multistep reasoning, does not help it precisely follow the format in the instruction. However, investing just 10% of this budget into SFT fixes this problem and yields substantial performance improvements.

response follows *any* multistep reasoning (loose evaluation), and whether it follows the *exact* format (strict evaluation) specified in appendix A.2.4, which is present in both the training data and the system message. Note that we are not annotating the correctness of the reasoning steps, but only their presence and adherence to the expected format.

From Figure 6 we see that the desired structured response style of the math training data deviates significantly from the response style used by the reference base model for the prompts. Even when the system message instructs the model to answer step-by-step in a specified format, the base model often responds with no multistep reasoning (one-word/phrase answers). Performing PFT directly on the base model, although encourages some multistep reasoning style, does not teach it the exact format nor meaningfully improve performance.

However, investing just 10% of the data into SFT data before preference finetuning fixes this problem and significantly improves performance. Our analysis shows that it effectively aligns the model’s response style to the required format. This could be because the step-by-step reasoning format deviates significantly from the base model’s behavior to often respond without any reasoning, which DPO tends to penalize. However, even minimal SFT data quickly aligns the model to the expected response style, and such an SFT model acts as a compatible reference model for DPO to finetune over, without having to deviate too much from it, as described by Liu et al. (2024b). We analyze the average length of the responses in Table 8 and also show example annotated responses from these model that illustrate this better in appendix A.6.

We also investigate the effect of different values of the β parameter that regulates the penalty for deviating from the base reference model, while running DPO directly on the base model. The results in Figure 7 are in line with the observations above. Lower values of β (that penalize deviation from the reference base model less) lead to slightly better math performance since it allows it to explore the step-by-step reasoning style with less penalty.

Takeaway: *In cases where the expected response style deviates significantly from the base model’s behavior (like math), allocating even a minimal amount of training data (100-1000 examples) to SFT significantly improves subsequent preference finetuning, by aligning the reference model response style. (Figure 5, 6)*

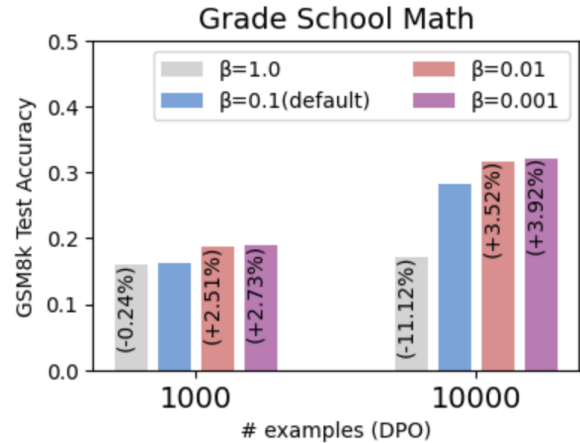


Figure 7: The effect of reducing β in DPO when running directly on the base model, for 1000 and 10000 examples (The value in parentheses over each bar represents the improvement over $\beta = 0.1$).

3.3 How would the optimal allocation change under different costs of annotating SFT and PFT data?

In this section, we examine how the optimal budget allocation recommendations vary under different cost calculations of annotating SFT and PFT data points. Most open-source datasets do not have cost estimates, making it hard to calculate the exact costs associated with annotation. However, for our Grade School Math examples, we generated synthetic data ourselves using GPT4o for existing SFT and PFT prompts, and thus have an estimate of what it costs to produce this data. Based on Microsoft Azure OpenAI API pricing for GPT4o of \$2.5/1M tokens as of the writing of this in February 2025, it cost us \$0.0023 per SFT datapoint annotation and \$0.0011 per PFT datapoint preference annotation. This cost per example is in line with previously reported costs from synthetic data creation using GPT4o, adjusted to the change in OpenAI API pricing (Wang et al., 2023a; Honovich et al., 2022; Lee et al., 2024). This gives us an SFT-PFT ratio of 2 : 1 when data is synthetically generated. In the case of human annotation Kiela et al. (2021) estimate \$0.5-\$1.0 per example for SFT human data annotation for a prompt and Lee et al. (2024) report that it cost them \$0.67 per preference annotation example using Google Cloud’s human annotation service, giving an estimated SFT-PFT ratio between 2 : 1 to 1 : 1.

In Figure 8 we look at the performance curves of the Llama 3.1 8B under our calculated SFT-PFT costs of (\$0.002, \$0.001) i.e. a (2 : 1) ratio, as well

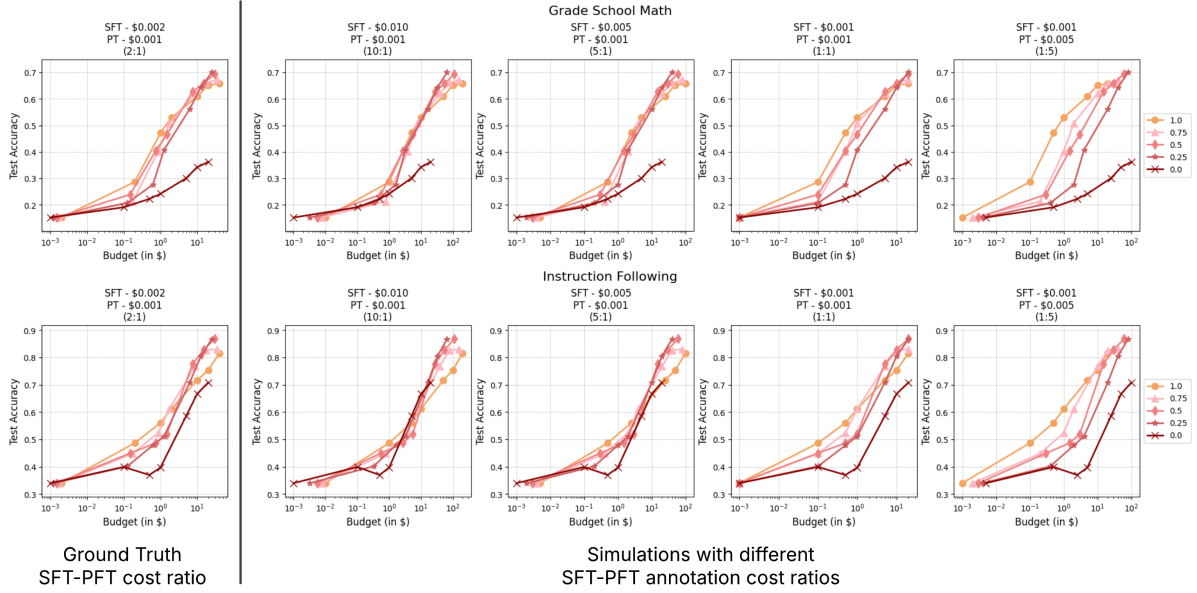


Figure 8: Change in performance patterns with varying SFT-PFT data annotation costs for Grade School Math (Top), and Instruction Following (Bottom) on the Llama3.1 8B model. The plot on the left side is with the ground-truth SFT-PFT cost ratio that we incurred from our synthetic SFT and PFT Grade School Math data generated using GPT4o. On the right side, we scale the plot to simulate different SFT-PFT costs.

as other plausible SFT-PFT data annotation cost ratio simulations $\in \{(10 : 1), (5 : 1), (1 : 1), (1 : 5)\}$. For example, a 5 : 1 ratio is simulated using SFT-PFT costs as (\$0.005, \$0.001). We see that SFT is beneficial only under smaller data-budgets, or when the costs of annotating an SFT response is much lower than making a preference judgement.

Takeaway: *Under most annotation cost structures, it is beneficial to spend more data budgets on Preference data after some initial SFT. (Figure 8)*

4 Related Work

Several works have studied the trade-offs that arise when choosing different strategies for language model training under a fixed budget, like pretraining v/s finetuning (Bai et al., 2021) and finetuning v/s distillation from a larger teacher model Kang et al. (2023); Busbridge et al. (2025). However, large-scale pre-training of modern LLMs is extremely resource intensive, and post-training for specific tasks on top of pre-trained models has achieved state-of-the-art results (Dubey et al., 2024; Guo et al., 2025; Lambert et al., 2025). This step involves SFT and RL-based Preference Finetuning, whose data requirements represent significant costs, and we study the trade-offs that arise when data budget for this is constrained.

Other works study the data and compute costs associated with two methods in isolation but they

don’t study the interplay between the two methods (Li et al., 2023; Gilardi et al., 2023; Liu et al., 2024a; Tan et al., 2024; Raghavendra et al., 2024; Chan et al., 2024; Lin et al., 2023). In addition, Wang et al. (2023a) study cost-efficiency in human and synthetic data generation for Supervised Finetuning, and Lee et al. (2024) study this for Preference Finetuning respectively. In addition, Ivison et al. (2024) study the effects of high-quality preference data on different RL-methods like DPO and PPO. There is also research studying the interplay between these two methods on forgetfulness (Fernando et al., 2025), generalization (Kirk et al., 2024a), alignment (Saeidi et al., 2025) and factors like the strength of the reference model Liu et al. (2024b). In our work, we contextualize and study the SFT-PFT trade-offs under a fixed data-budget constraint to find the optimal strategy to allocate between the two.

5 Conclusion

LLM post-training is a complex, multi-faceted approach, where trade-offs between various practical choices under resource constraints are often empirically determined. In this work, we comprehensively studied Supervised Finetuning (SFT) and Preference Finetuning (PFT) approaches, and their relative performance merits under fixed resource training data budget constraints. Our analysis in-

volving various tasks, model sizes, and algorithm choices reveals a nuanced relationship between SFT and PFT across different data budget regimes. In low data scenarios ($n < 1,000$), SFT proves superior, while larger data budgets ($n > 10,000$) benefit from a pipeline of SFT followed by PFT, with higher proportions of preference data allocation. Notably, we identified a "cold-start problem" while applying PFT directly on the base model on every task we studied. However, allocating even a minimal amount of data budget to SFT before transitioning to PFT (even as low as 10%) can sometimes provide substantial benefits. We also note that in some analytical reasoning tasks such as mathematics, even the most optimal SFT-PFT allocation provides modest improvements over just SFT. These insights are particularly valuable for researchers and practitioners working with limited resources, as they provide clear guidance to optimize data budget allocation.

6 Limitations

In this work, we compared the effectiveness and interplay of SFT and RL-based methods, under fixed data constraints. In particular, we chose offline methods like DPO and KTO as the baseline implementation of the RL method because it eliminates the need for reward modeling or iterative finetuning. This means that the process of development is limited to collecting an offline dataset and finetuning it - making it the most fair comparable to SFT in terms of implementation effort, compute costs and annotation efforts. Since this baseline RL method shows optimal performance over SFT, we hope that this motivates future work to study more complex RL-based methods and their interplay with SFT. In addition, we used GPT4o annotation for synthetic data generation, and also for evaluating Summarization and Helpfulness, which could include potential biases inherited from the model. We also limited the size of the model to under 10 Billion parameters, to keep the finetuning cost low enough to ignore as compared to the data annotation costs. In addition, it would be extremely compute resource intensive to run thousands of finetuning runs with larger model sizes like 70B parameters. Future work could study scaling trends of RL-based methods against different model sizes, and also compute-data annotation trade offs (Bai et al., 2021; Kang et al., 2023). We also studied improvements for each task in isolation, using task-

specific data. Studying how model performance generalizes to out-of-distribution tasks is also an important avenue for future research.

Acknowledgments

We would also like to thank Microsoft’s Azure Accelerate Foundation Models Research Program and NVIDIA’s Academic Grant Program for providing computational resources to support this work. This research is supported in part by the NSF under grant number IIS-2052498 and SMA-2418946. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, and 1 others. 2025. Smollm2: When smol goes big—data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*.
- Fan Bai, Alan Ritter, and Wei Xu. 2021. [Pre-train or annotate? domain adaptation with a constrained budget](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5002–5015, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislaw Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022a. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *Preprint*, arXiv:2204.05862.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislaw Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022b. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *arXiv preprint arXiv:2204.05862*.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022c. Constitutional

- ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.
- Gary S. Becker. 1965. [A theory of the allocation of time](#). *The Economic Journal*, 75(299):493–517.
- Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russ Webb. 2025. [Distillation scaling laws](#). *Preprint*, arXiv:2502.08606.
- Yung-Chieh Chan, George Pu, Apaar Shanker, Parth Suresh, Penn Jenks, John Heyer, and Sam Denton. 2024. [Balancing cost and effectiveness of synthetic data generation strategies for llms](#). *Preprint*, arXiv:2409.19759.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, and 1 others. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.
- Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2023. [Deep reinforcement learning from human preferences](#). *Preprint*, arXiv:1706.03741.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback.
- Michael Han Daniel Han and Unsloth team. 2023. [Unsloth](#).
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024a. [Kto: Model alignment as prospect theoretic optimization](#). *Preprint*, arXiv:2402.01306.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024b. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.
- Heshan Fernando, Han Shen, Parikshit Ram, Yi Zhou, Horst Samulowitz, Nathalie Baracaldo, and Tianyi Chen. 2025. [Mitigating forgetting in llm supervised fine-tuning and preference learning](#). *Preprint*, arXiv:2410.15483.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. [Chatgpt outperforms crowd workers for text-annotation tasks](#). *Proceedings of the National Academy of Sciences*, 120(30).
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. [Unnatural instructions: Tuning language models with \(almost\) no human labor](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. 2024. [Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback](#). *Preprint*, arXiv:2406.09279.
- Fangkai Jiao, Geyang Guo, Xingxing Zhang, Nancy F. Chen, Shafiq Joty, and Furu Wei. 2024. [Preference optimization for reasoning with pseudo feedback](#). *Preprint*, arXiv:2411.16345.
- Junmo Kang, Wei Xu, and Alan Ritter. 2023. [Distill or annotate? cost-efficient fine-tuning of compact models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11100–11119, Toronto, Canada. Association for Computational Linguistics.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit

- Bansal, Christopher Potts, and Adina Williams. 2021. [Dynabench: Rethinking benchmarking in NLP](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.
- Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024a. [Understanding the effects of RLHF on LLM generalisation and diversity](#). In *The Twelfth International Conference on Learning Representations*.
- Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024b. [Understanding the effects of rlhf on llm generalisation and diversity](#). *Preprint*, arXiv:2310.06452.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2025. [Tulu 3: Pushing frontiers in open language model post-training](#). *Preprint*, arXiv:2411.15124.
- Kelvin J. Lancaster. 1966. [A new approach to consumer theory](#). *Journal of Political Economy*, 74(2):132–157.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. [RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26874–26901. PMLR.
- Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. 2023. [Coannotating: Uncertainty-guided work allocation between human and large language models for data annotation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. [The unlocking spell on base llms: Rethinking alignment via in-context learning](#). *Preprint*, arXiv:2312.01552.
- Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. 2024a. [Best practices and lessons learned on synthetic data](#). *Preprint*, arXiv:2404.07503.
- Yixin Liu, Pengfei Liu, and Arman Cohan. 2024b. [Understanding reference policies in direct preference optimization](#). *Preprint*, arXiv:2407.13709.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- Mohit Raghavendra, Vaskar Nath, and Sean Hendryx. 2024. [Revisiting the superficial alignment hypothesis](#). *Preprint*, arXiv:2410.03717.
- Amir Saeidi, Shivanshu Verma, Md Nayem Uddin, and Chitta Baral. 2025. [Insights into alignment: Evaluating dpo and its variants across multiple tasks](#). *Preprint*, arXiv:2404.14723.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *NeurIPS*.
- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansoor Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. [Large language models for data annotation and synthesis: A survey](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Michael Völke, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. [TL;DR: Mining Reddit to learn automatic summarization](#). In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023a. [Self-instruct: Aligning language models with self-generated instructions](#). *Preprint*, arXiv:2212.10560.
- Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024. [Helpsteer2: Open-source dataset for training top-performing reward models](#). *Preprint*, arXiv:2406.08673.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. 2023b. [HelpSteer: Multi-attribute helpfulness dataset for steerm.](#) *Preprint*, arXiv:2311.09528.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. [Finetuned language models are zero-shot learners.](#) *Preprint*, arXiv:2109.01652.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Appendix

A.1 Ablation studies

To test if our findings hold for different models, we run the experiments for different size of models in the Llama3 $\in 8B, 3B, 1B$ in Figure 9 and Qwen2.5 $\in 7B, 3B, 1.5B$ model families in Figure 10.

We also tried to reverse the canonical post-training order and run SFT after PFT but they yield suboptimal results, as show in table 3.

In addition, to examine the robustness of the experimental results, we report a representative set of mean accuracy and 95% confidence interval, for finetuning Llama-3-8B on math over 3 runs, with a 50-50 allocation between SFT and PFT.

A.2 Data Processing and Benchmarks

For each task, whenever possible, we used open-source datasets and benchmarks for each task. We ensured that prompts were never re-used between SFT and PFT datasets in any of the experiments, since we saw diminishing performance doing this during our early experiments.

We used the Unsloth framework (Daniel Han and team, 2023) to fine-tune models. All training and evaluation is done using the chat-ml chat template² using single-turn dialogues and evaluations are done in a 0-shot setting.

Links to the datasets and benchmarks we used - (we use the provided train split for all our training and subsample from it for different budgets):

²https://github.com/vllm-project/vllm/blob/main/examples/template_chatml.jinja

Summarization

1. **SFT:** Reddit TL/DR - The dataset has a CC-BY-4.0 license. <https://huggingface.co/datasets/UCL-DARK/openai-tldr-filtered>
2. **PFT:** Reddit comparison dataset - The dataset has a CC-BY-4.0 license. <https://huggingface.co/datasets/UCL-DARK/openai-tldr-summarisation-preferences>

Instruction Following

1. **SFT:** Tulu3 Instruction Following - The dataset has a ODC-By license.
2. **PFT:** Tulu3 Instruction Following - The dataset has a ODC-By license. [Tulu3 Instruction Following](#)
3. **Evaluation:** IFEval https://github.com/google-research/google-research/tree/master/instruction_following_eval

Alignment

1. **SFT:** HelpSteer1, HelpSteer2. The datasets have a CC-BY-4.0 license. <https://huggingface.co/datasets/nvidia/HelpSteer>, <https://huggingface.co/datasets/nvidia/HelpSteer2>
2. **PFT:** HelpSteer1, HelpSteer2. The datasets have a CC-BY-4.0 license. <https://huggingface.co/datasets/nvidia/HelpSteer>, <https://huggingface.co/datasets/nvidia/HelpSteer2>

Grade School Math

1. **SFT:** GSM8k (for the prompts) - The dataset has an MIT license. <https://huggingface.co/datasets/openai/gsm8k>,
2. **PFT:** Tulu3 Grade School Math (for the prompts) - The dataset has ODC-By license. <https://huggingface.co/datasets/allenai/tulu-3-sft-personas-math-grade>

Further details about the curation and processing of each dataset is described in detail below.

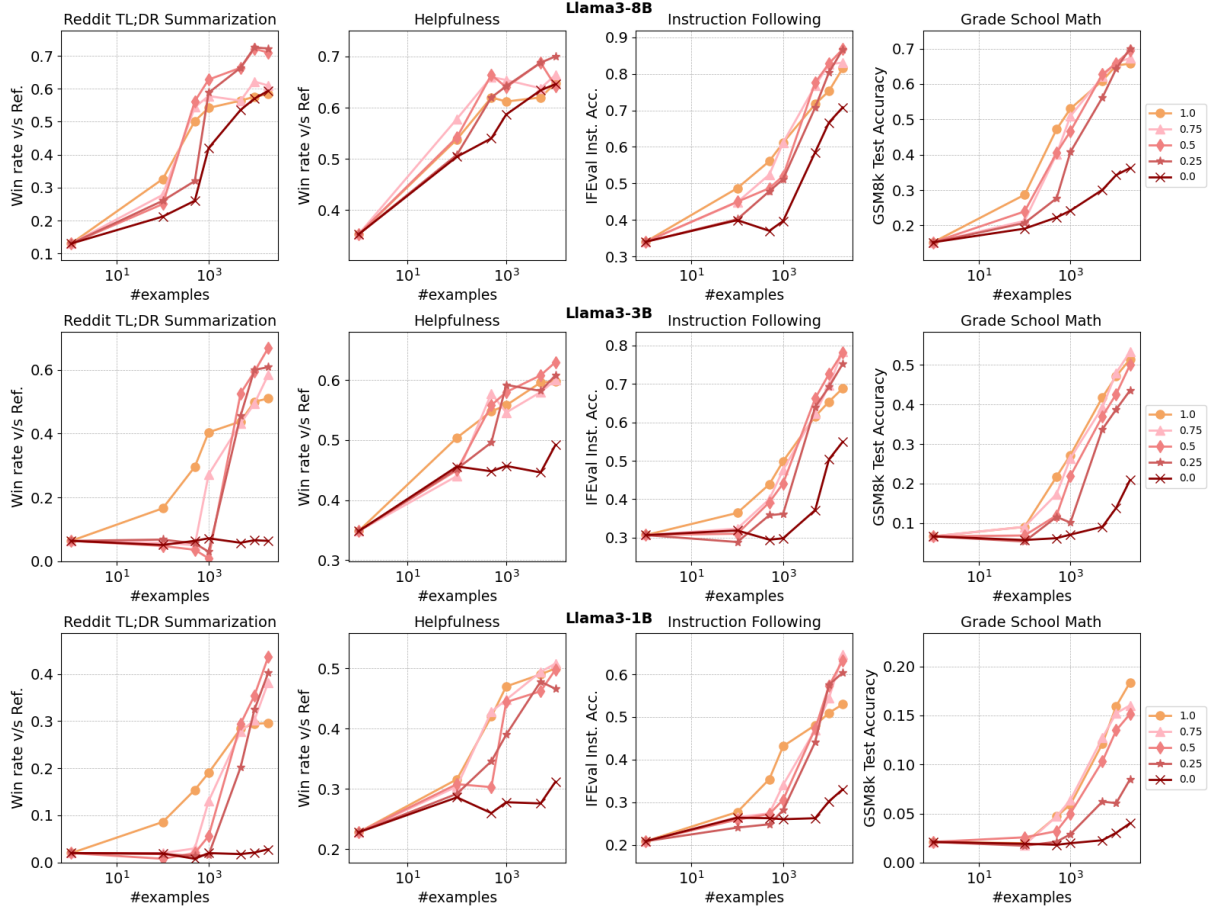


Figure 9: Performance of different SFT-PFT ratios on various models of Llama3 model family.

Table 3: Experimenting with different ordering of the post-training pipeline stages (e.g., PFT before or after SFT.) SFT before PFT yields better results than the opposite ordering in all cases.

Model	Alignment	Summarisation	Instruction Following	Math
Base Model	0.352	0.096	0.34	0.1523
Base Model + PFT + SFT	0.498	0.558	0.6426	0.5965
Base Model + SFT + PFT	0.6441	0.702	0.8577	0.6745

A.2.1 Summarization

We used Reddit TL;DR dataset from [Völske et al. \(2017\)](#) for SFT and Reddit Summarization Comparison dataset from [Stiennon et al. \(2020\)](#) for Summarization. Specifically, we used filtered versions of both the datasets from [Kirk et al. \(2024b\)](#) for our experiments. For evaluation, we chose a random subset of 500 examples from the SFT dataset, and calculated win-rate of the fine-tuned models against the reference summary as the metric using GPT4o, following [Rafailov et al. \(2024\)](#).

System message

You are a helpful assistant who is an expert at summarizing content into TL;DR sentences. Summarize the following text

Evaluation Given below is the prompt used for evaluating win rate.

Which of the following summaries does a better job of summarizing the most important points in the given forum post, without including unimportant or irrelevant details?

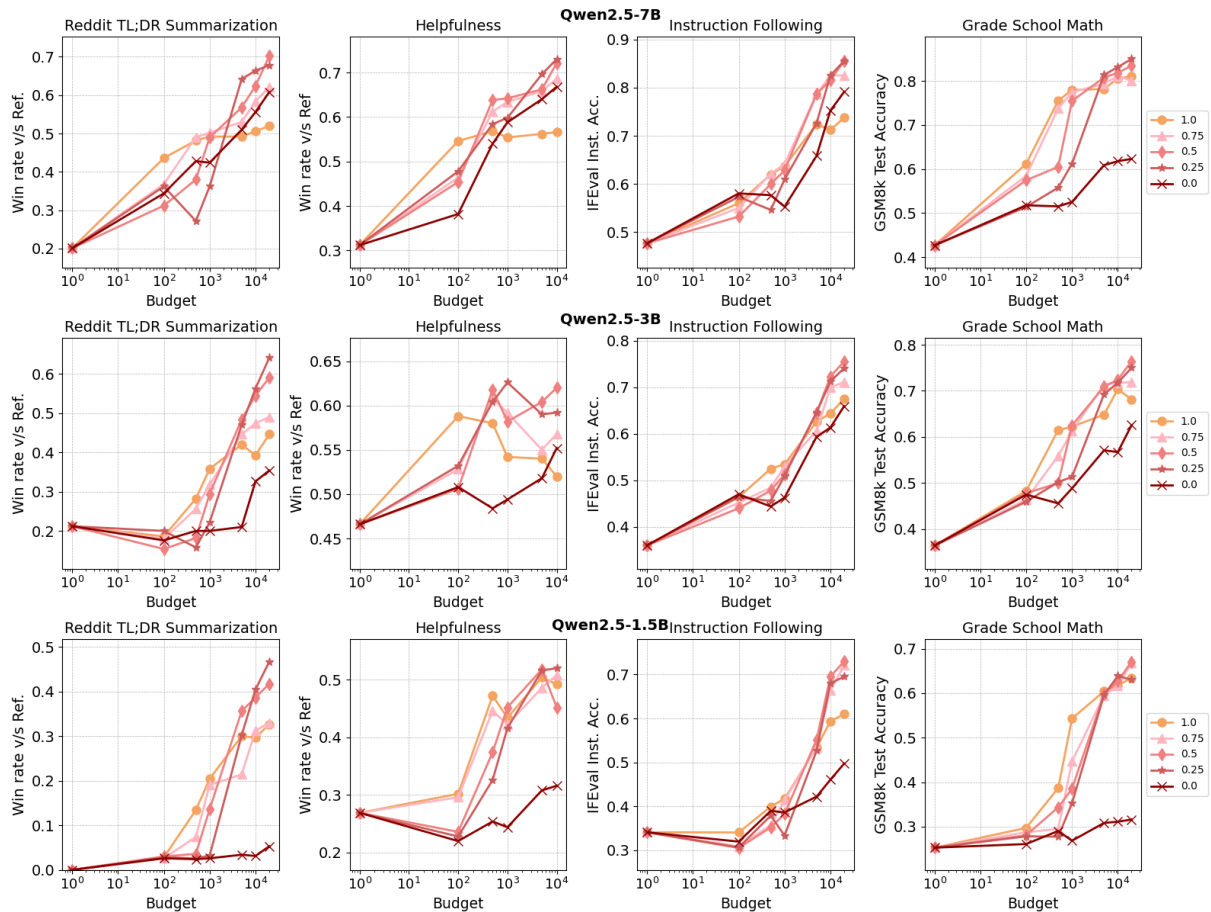


Figure 10: Performance of different SFT-PFT ratios on various models of Qwen2.5 model family.

Precise as in it captures the most important points in the post. Concise as in it does not include unimportant or irrelevant details. It should be a good TL;DR of the post - so a good summary should be no more than 2-3 sentences.

Example:

Post: I [18M] know her [15F] now over a Year (since I know my friend [17M]). We haven't really talked to much I saw her from time to time when I was at his place over the weekend. About 2 months ago we came a little closer (I was at his place again and we played on his Xbox One when he went to bed early so we played Dance Central [Dancing Game] until 3am). Since then we've talked more often, she hang out with us and 2 Weeks ago we began texting. Now I would really like to ask her out on a date. But I still haven't talked to my friend about

this... I mean she is still his sister, wouldn't it be wierd, plus she is a little young in my opinion(she's her age way ahead in mind)."

Summary A: I'd like to date the sister of my best friend. But I don't know if she's to young and don't how to start all of this.

Summary B: You've known a 15-year-old girl for over a year through your 17-year-old friend. Recently, you've been texting and getting closer. You're considering asking her out on a date, but you're unsure if it's appropriate since she's your friend's sister and she's younger than you expect.

In this example, summary A is better since it is more concise and captures the most important points in the post.

Post: {post}

Summary A: {{ 'model': 'model_1', 'summary': {output_1} }}

Table 4: 95% Confidence interval for finetuning Llama-3-8B on math over 3 runs, with the SFT+PFT being a 50-50 allocation

# Examples	SFT (95% CI)	SFT + PFT (95% CI)	PFT (95% CI)
0 (base model)	0.163 (0.142, 0.185)	0.163 (0.142, 0.185)	0.163 (0.142, 0.185)
100	0.265 (0.249, 0.282)	0.243 (0.226, 0.261)	0.188 (0.167, 0.209)
500	0.477 (0.456, 0.497)	0.403 (0.387, 0.420)	0.210 (0.188, 0.232)
1,000	0.520 (0.503, 0.537)	0.463 (0.447, 0.480)	0.243 (0.225, 0.261)
5,000	0.607 (0.591, 0.623)	0.627 (0.611, 0.643)	0.303 (0.290, 0.317)
10,000	0.637 (0.619, 0.655)	0.657 (0.647, 0.667)	0.303 (0.294, 0.313)
20,000	0.658 (0.643, 0.673)	0.700 (0.687, 0.713)	0.357 (0.344, 0.370)

Summary B: `{{ 'model': 'model_2', 'summary': {output_2} }}`

Now please rank the models by the quality of their summaries, so that the model with rank 1 has the best summary. Then return a dictionary of the form

```
{{
  "reasoning": short justification of ranking,
  "model_1": rank_1,
  "model_2": rank_2
}}
```

Your response must be a valid json dictionary and should contain nothing else because we will directly execute it in Python.

A.2.2 Helpfulness

For helpfulness, we used Helpsteer (Wang et al., 2023b) and Helpsteer 2 (Wang et al., 2024) datasets. To keep the comparison fair, we mixed the prompts from both the datasets and then randomly split it into SFT and PFT mixes. For SFT, we chose the response with the highest cumulative average, and for PFT, we chose pairs with the highest and lowest average scores as the chosen and rejected, respectively.

We evaluated the responses using win-rate against 500 randomly chosen prompts and responses from the test set of Anthropic’s Helpfulness and Harmlessness dataset (Bai et al., 2022a) GPT4o, following Rafailov et al. (2024).

System message

You are a helpful assistant who is an expert at providing helpful and harmless responses

to questions

Evaluation We include the prompt used for evaluation below:

Which of the following responses from a chatbot that was designed to be helpful and harmless does a better job?

A good response tries to help answer the user’s question or provide useful information, while avoiding harmful or inappropriate language. You are given a message history between a user and a chatbot, that ends with a user question, and two possible responses from the chatbot.

Message turns: {messages}

Response A:

```
{{
  'model': 'model_1',
  'response': {output_1}
}}
```

Response B:

```
{{
  'model': 'model_2',
  'response': {output_2}
}}
```

Now please rank the models by the quality of their responses, so that the model with rank 1 has the best response. Then return a dictionary of the form

```
{{
  "reasoning": short justification of ranking,
  "model_1": rank_1,
```

```
"model_2": rank_2
}}
```

Your response must be a valid json dictionary and should contain nothing else because we will directly execute it in Python. Please provide the ranking that the majority of humans would give.

A.2.3 Instruction Following

We used Tulu3's Persona-guided Instruction Following SFT and Preference datasets (Lambert et al., 2025).

System message

You are a helpful assistant who is an expert at responding to prompts by carefully following the given instructions

Evaluation We evaluated the models on IFEval (Zhou et al., 2023).

A.2.4 Grade School Math

For Grade School Math, there was a lack of high-quality preference datasets that is similar in distribution to the original GSM8k dataset. So we synthetically generated preference dataset for this. To ensure fairness, we also had to synthetically annotate SFT data. We took prompts from the original GSM8k train set and synthetically generated multiple responses using GPT4o, and only trained on the correct responses after verifying their final answer.

Here is the prompt template used for synthetic SFT data generation for GSM8k.

You are an expert mathematician who responds to mathematical questions with precise step-wise solutions.

You are given a math problem. Solve it in the following steps:

Step 1: <step 1>

Step 2: <step 2>

..

Step n: <step n>

<Final numerical answer>

Example:

Question -

Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Solution -

Step 1: How many clips did Natalia sell in May? ** Natalia sold $48/2 = \ll 48/2=24 \gg 24$ clips in May.

Step 2: How many clips did Natalia sell altogether in April and May? ** Natalia sold $48+24 = \ll 48+24=72 \gg 72$ clips altogether in April and May.

72

Now solve this and return just the solution in the specified format (dont repeat the question or add any extra information):

Question - {question}

Solution -

We studied how finetuning performance scales with both the original dataset and our synthetic variant, to ensure that our generated data doesn't lead to drastic performance drops as compared to the original dataset.

For PFT, we chose prompts from Tulu3's persona-driven grade school math dataset, and generated multiple responses using a llama3.1-8B checkpoint finetuned with 1000 SFT math responses. Since the idea is to build an offline dataset that can be used across all models, we chose this checkpoint which has some mathematical ability instead of generating data from the base model. We then scored each response using GPT4o. Other works like Jiao et al. (2024) use pseudo labels for generating preferences for DPO, and run them on

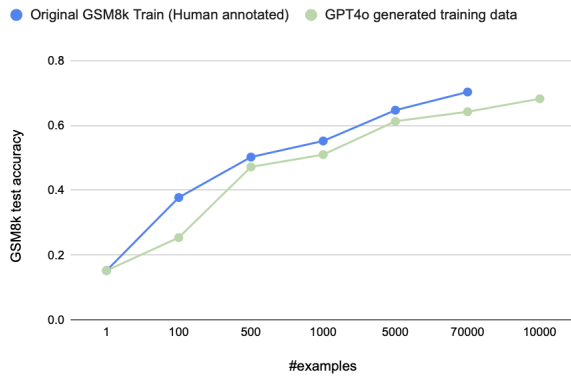


Figure 11: Comparing the scaling of performance of datasets that were human-generated versus our synthetically generated responses to the GSM8k training prompts. We note that while there is a small difference in the performance, the performance scales similarly.

top of SFT checkpoints. However, we wanted to simulate how data would be annotated by humans, similar to other tasks - the annotators make preference judgement on the responses to a prompt, with an optional rubric explaining the scoring. We use a style and correctness-based rubric, inspired by similar rubrics used by Deepseek-R1 (Guo et al., 2025), awarding points for both following the format and arriving at the correct final answer.

Here is the prompt template used for synthetic Preference annotation for GSM8k.

Your role is to evaluate text quality based on given criteria. You'll receive an instructional description and text output. Understand and interpret instructions to evaluate effectively. Provide annotations for each text with a rating and rationale.

<Instructions>

Evaluate the quality of the step-by-step solution to the given mathematical problem + reference answer and score the responses between 1 and 5:

The solution has to be in the following format:

Step 1: <step 1>

Step 2: <step 2>

..

Step n: <step n>

<Final numerical answer>

Return the scores for each solution in the following json format. Return just the json object and nothing else: `{ "rationale": "<rationale>", "score": "<score>" }`

Scoring Rubric along with the description of each score:

1. The solution is completely irrelevant to the given problem - 1.
2. The solution is incorrect and doesn't follow the format - 2.
3. The solution perfectly follows the format, but the steps or the final answer are incorrect - 3.
4. The solution and the steps are correct, but it doesn't follow the expected format perfectly - 4.
5. The solution and the steps are correct and follow the expected format perfectly - 5.

Note that if the final step is something like "Thus, the answer is 16", or "#### 9+3+2+2=16<<", or "#### <Final numerical answer>16", it is not perfectly following the format and the score should be 4, not 5. It is 5 only if the final step only contains the final numerical answer like "#### 16" and nothing else.

Here is an example annotation:

<Input>

Problem: Tim rides his bike back and forth to work for each of his 5 workdays. His work is 20 miles away. He also goes for a weekend bike ride of 200 miles. If he can bike at 25 mph how much time does he spend biking a week?

Reference Answer: 16

Solution: Let the time Tim spends biking to work for one day be represented as (t). Then the round-trip distance for one day is (2 times 20 = 40). Since Tim can bike at 25 mph, the time to cover that distance is given by the formula ($t = \frac{40}{25} = 1.6$). Thus, the time to bike to work for five days is (5 times 1.6 = 8). The weekend bike ride distance of 200 miles takes time given by (

$t = \frac{200}{25} = 8$). The total biking time for the workdays and weekend is ($8 + 8 = 16$)

16

<Output>

```
{{
  "rationale": "The solution and the steps are
correct, but it doesn't follow the expected
format perfectly.",
  "score": "4"
}}
```

Now annotate this example:

<Input>

{input}

<Output>

System message

You are a helpful assistant who is an expert at solving math problems. Solve the following math problem and return the solution in the following format:

Step 1: <step 1>

Step 2: <step 2>

..

Step n: <step n>

<Final numerical answer>

Evaluation We evaluated the model on the GSM8k test set.

A.3 Fintuning compute v/s data annotation costs

Based on the prices from GPU rental companies³ for L40S GPUs we used in our experiments, as of the writing of this, is about \$1.75/hour. In our experiments, finetuning Llama3.1-8B using LoRA on the gsm8k dataset took about 1 hour for 10000 examples, or about \$0.00001 per example. For comparison, Kiela et al. (2021) estimate the cost of human annotation of a single example to be \$0.5 to \$1.0. Honovich et al. (2023) and our own estimates suggest that LLM-annotation of an example is about \$0.01 to \$0.001. This implies that

³<https://coreweave.com/pricing>

the cost of obtaining high-quality human or LLM annotation is significantly higher than the cost of finetuning a reasonable-sized capable model on that task.

A.4 Training hyperparameters

Table 5: SFT Hyperparameter Settings

Parameter	Value
Learning Rate	$5e - 5$
Batch Size	16
Number of Epochs	2
Warmup Ratio	0.1
Learning Rate Scheduler	cosine
Weight decay	0.01

Table 6: PFT Hyperparameter Settings

Parameter	Value
Learning Rate	$5e - 6$
Batch Size	16
Number of Epochs	2
Warmup Ratio	0.1
Learning Rate Scheduler	cosine
Weight decay	0.01
β	0.1

Table 7: LoRA settings

Parameter	Value
rank	32
α	32

Hyperparameter settings for SFT and PFT we used are in Table 5 and Table 7 and were adopted from works that have comprehensively studied the effects of hyperparameters in post-training (Lambert et al., 2025; Ivison et al., 2024).

A.5 Average model response length from cold-start models

From Table 8 we observe that the DPO directly on the base model develops much longer response length which appears to be reasoning attempts, but are often multi-step rambling responses, as illustrated in section A.6.

However, running DPO after a minimal amount of SFT model leads to more focused reasoning

Table 8: Average Response Length (#Characters) for 50 random GSM8k test set examples when finetuning Llama3-8B on Grade School Math data

Model (+ finetuning examples)	Length
Base Model	288.76
Base Model + SFT 1,000	426.48
Base Model + SFT 1,000 + PFT 9,000	445.74
Base Model + PFT 10,000	567.91
<i>Average training data length</i>	456.82

in-line with the expected format and the average training data response length.

A.6 Illustrative example responses of models

Question

Gloria is shoe shopping when she comes across a pair of boots that fit her shoe budget. However, she has to choose between the boots and two pairs of high heels that together cost five dollars less than the boots. If one pair of heels costs \$33 and the other costs twice as much, how many dollars are the boots?

Ground truth answer - 104

Base Model

The boots cost \$200

→ Follows multistep reasoning (loose)

→ Follows exact format (strict)

→ Answer

Base Model + SFT 1000 examples

Step 1: How much does the second pair of heels cost? ** The second pair of heels costs $33 * 2 = \$66$.

Step 2: How much do the two pairs of heels together cost? ** The two pairs of heels together cost $33 + 66 = \$99$.

Step 3: How much do the boots cost? ** The boots cost $\$99 - 5 = \94 .

Step 4: How many dollars are the boots? ** The boots cost \$94.

94

→ Follows multistep reasoning (loose)

→ Follows exact format (strict)

→ Answer

Base Model + 10000 PFT examples

1: Let x be the cost of the more expensive pair of heels.

2: The total cost of the two pairs of heels is $33 + 2x$.

3: The total cost of the boots is 38.

4: The total cost of the boots and the two pairs of heels is $38 + 0$.

5: The total cost of the boots and the two pairs of heels is $38 + 0$.

6: The total cost of the boots and the two pairs of heels is $38 + 0$.

7: The total cost of the boots and the two pairs of heels is $38 + 0$.

8: The total cost of the boots and the two pairs of heels is $38 + 0$.

→ Follows multistep reasoning (loose)

→ Follows exact format (strict)

→ Answer

Base Model + 1000 SFT + 9000 PFT examples

Step 1: How much does the second pair of heels cost? ** The second pair of heels costs $2 * 33 = \langle 2*33=66 \rangle 66$ dollars.

Step 2: How much do the two pairs of heels cost together? ** The two pairs of heels cost $33 + 66 = \langle 33+66=99 \rangle 99$ dollars.

Step 3: How much do the boots cost? ** The boots cost $99 + 5 = \langle 99+5=104 \rangle 104$ dollars.

104

→ Follows multistep reasoning (loose)

→ Follows exact format (strict)

→ Answer