Journal of Data-centric Machine Learning Research (2024) Submitted 2/08; Revised 4/08; Published ?

# Long Is More for Alignment: A Simple but Tough-to-Beat Baseline for Instruction Fine-Tuning

**Hao Zhao**
**Maksym Andriushchenko**
**Francesco Croce**
**Nicolas Flammarion**
*École Polytechnique Fédérale de Lausanne*

**Reviewed on OpenReview:** *https://openreview.net/forum?id=X5l8bqk5K8*

**Editor:** My editor

## Abstract

There is a consensus that instruction fine-tuning of LLMs requires *high-quality* data, but what are they? LIMA (NeurIPS 2023) and AlpaGasus (ICLR 2024) are state-of-the-art methods for selecting such high-quality examples, either via manual curation or using GPT-3.5-Turbo as a quality scorer. We show that the extremely simple baseline of selecting the *1,000 instructions with longest responses* from standard datasets can consistently outperform these sophisticated methods according to GPT-4 and PaLM-2 as judges, while remaining competitive on the Open LLM benchmarks that test factual knowledge. We demonstrate this for several state-of-the-art LLMs (Llama-2-7B, Llama-2-13B, and Mistral-7B) and datasets (Alpaca-52k and Evol-Instruct-70k). In addition, a lightweight refinement of such long instructions can further improve the abilities of the fine-tuned LLMs, and allows us to obtain the 2nd highest-ranked Llama-2-7B-based model on AlpacaEval 2.0 while training on only 1,000 examples and no extra preference data. We also conduct a thorough analysis of our models to ensure that their enhanced performance is not simply due to GPT-4's preference for longer responses, thus ruling out any artificial improvement. In conclusion, our findings suggest that fine-tuning on the longest instructions should be the default baseline for any research on instruction fine-tuning.

**Keywords:** large language model, supervised instruction fine-tuning, data selection

## 1 Introduction

Pre-trained large language models (LLMs) need to undergo an alignment phase [2, 4, 25, 34, 31] to make them suitable for downstream tasks like user interaction or question answering. While the details may vary, alignment often relies on supervised fine-tuning (SFT) on a dataset of instruction-response pairs to improve conversational ability, followed by reinforcement learning from either human (RLHF) [25] or automated (RLAIF) [3, 19] feedback to promote the preferred style and content of replies. It is an active research direction to study whether it is possible to achieve satisfactory results while relying only on SFT, which would avoid the (potentially expensive) process of collecting preference data. Taori et al. created Alpaca, an open source dataset of 52k instruction-response pairs, and fine-tuned

©2024 Hao Zhao, Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion.

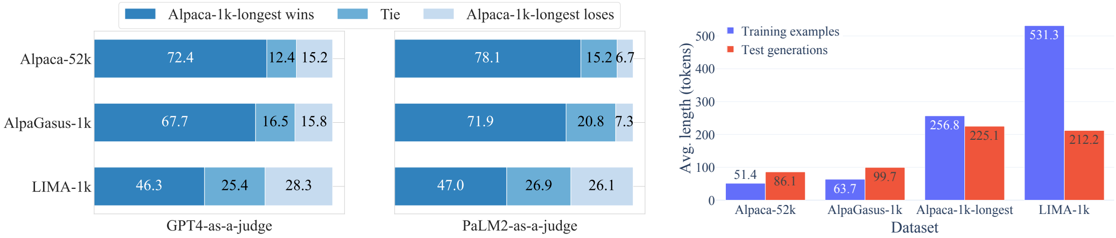


Figure 1: **Selecting the longest responses leads to a strong IFT dataset.** We fine-tune LLaMA-2-7B models on Alpaca-52k [31], AlpaGasus-1k [7], LIMA-1k [42] and our Alpaca-1k-longest datasets. **(a)** Alpaca-1k-longest beats three baselines in instruction-following performance according to both GPT-4 and PaLM-2 as judges. **(b)** Alpaca-1k-longest leads to an average response length at test time higher than Alpaca-52k and AlpaGasus-1k, but similar to LIMA-1k: then its higher win rate cannot be solely attributed to the model having learnt to generate long responses.

on it a Llama-2-7B model to match the performance of the closed-source text-davinci-003 model. Then, Chen et al. introduced AlpaGasus, consisting of the 9k examples of Alpaca which are judged of highest quality by GPT-3.5-Turbo, to further improve the instruction-following abilities of the fine-tuned models. The intuition that instruction fine-tuning (IFT) might benefit from fewer demonstrations but of higher quality has been further pursued by Zhou et al. which manually curated LIMA, a dataset of 1k examples, which outperforms AlpaGasus. While the quality of the instructions seems to play a major role for IFT, it remains unclear which are the distinguishing features of high quality demonstrations.

In this work, we revisit the significant efforts in constructing instruction-tuning datasets from prior work. Inspired by the fact LIMA contains much longer examples than Alpaca and the observation of recent works [30, 38] that RLHF and direct preference optimization (DPO) [28] seem to mostly make the outputs longer, we test selecting longest responses as a simple and inexpensive heuristic to curate a small (only 1k examples) and high-quality IFT dataset from a larger one. Surprisingly, fine-tuning a Llama-2-7B [32] base model on the 1k longest elements of Alpaca outperforms both AlpaGasus and LIMA in one-to-one comparison with different LLMs as judges (see Fig. 1). Moreover, simply improving the quality and the style of the response in Alpaca-1k-longest with GPT-3.5-Turbo, in combination with NEFTune noise augmentation [16], allows us to obtain the *the 2nd highest-ranked Llama-2-7B-based model* on AlpacaEval 2.0. In this case, our simple method yields models which surpass LLMs with the same base model but fine-tuned with orders of magnitude more instructions as well as millions of preference data points.

Next we analyze several aspects of our models to understand the unexpected effectiveness of our approach. First, via several ablation studies, we show that our models do not just exploit the bias to favor longer responses of GPT-4 [24] or PaLM-2 [1], but provide higher quality replies. Then, since Jha et al. and Gudibande et al. suggest that optimizing performance of instruction-following tasks might be disconnected from factual knowledge, we additionally test our models on then Open LLM benchmarks. On these datasets assessing reasoning and factuality, our models perform similarly or better than the baselines fine-tuned on AlpaGasus and LIMA from the same base model, i.e. with the same factual knowledge coming from pre-training. Finally, we confirm our findings with extensive experiments using multiple IFT datasets (Alpaca, Evol-Instruct) and architectures (Llama-2-7B, Llama-2-13B, Mistral-7B [18]), and including head-to-head evaluation and on established benchmarks (AlpacaEval 2.0, Open LLM), to show the generality of our approach.

In summary, we uncover the surprising effectiveness of fine-tuning only on the longest 1,000 instructions of large datasets to obtain aligned models. Moreover, we show that such small datasets, potentially refined via an inexpensive automatic process, constitute a strong and tough-to-beat baseline for any method for instruction fine-tuning.

## 2 Fine-tuning on long instructions is a very strong baseline

### 2.1 Subsampling high-quality IFT datasets

**Existing methods.** Recent works have shown that IFT on a small curated dataset of instructions is sufficient to enhance the ability of LLMs to follow instructions and complete tasks. In particular, Chen et al. adopt GPT-3.5-Turbo as the oracle to judge the quality of (instruction, input, output) tuples with grades on a 1-5 scale. Only the highest scoring examples (grade $\geq 4.5$) from Alpaca-52k (but the same approach can be generalized to other datasets) are used to form the AlpaGasus dataset on 9k instructions. Later, Zhou et al. collect 750 top instruction-response pairs from community forums with some heuristic rules, such as comments and upvotes, and manually write 250 examples to enhance task diversity and quality. These 1,000 examples are optimized for a uniform response style to turn the LLM into a useful conversational agent, and constitute the LIMA-1k dataset.

**Our simple baseline: 1k-longest instructions.** Though both AlpaGasus and LIMA present promising performance improvements, they require either access to proprietary LLMs or very expensive human labor. Then, since previous works suggest that longer responses naturally arise during alignment [30, 38], we explore response length as the selection criterion to prune IFT datasets. *We select the 1,000 longest responses from the popular Alpaca-52k and Evol-Instruct-70k datasets* to form our IFT datasets that we refer to as Alpaca-1k-longest and Evol-Instruct-1k-longest. We restrict ourselves to using 1,000 examples for consistency with LIMA and since we are interested in testing how far the instruction following ability of LLMs can be pushed with a minimal SFT dataset. Using longer examples can be seen as a natural choice since these are usually more informative and thus contain more features relevant to human intentions. Longer responses are also intuitively harder for LLMs to fit, which forces the model to actually learn the response style rather than just memorize the answer. In addition, fitting longer responses encourages the model to capture long-distance semantic connections, and stay on-topic when answering complicated instructions. Interestingly, we observe that the instructions with longest responses minimally overlap with those receiving high score by LLMs: for example, most of the 1k longest examples from Alpaca receive a score of 3.5 from GPT-3.5-Turbo, i.e. signficantly lower than those in AlpaGasus (see details in Fig. 10 in App. C.1).

### 2.2 Effectiveness of our approach for open-ended generation

**Setting.** To test the effectiveness of our approach, we compare our 1k-longest datasets to the full original Alpaca and Evol-Instruct datasets (52k and 70k examples), the 1k examples with highest scores according to GPT-3.5-Turbo as done by Chen et al. (hence we refer to these as AlpaGasus-1k and Evol-Instruct-AlpaGasus-1k), and LIMA-1k. For each instruction dataset, we fine-tune Llama-2-7B base models (complete training configurations in App. B.2). Then, we test their abilities on five evaluation datasets (LIMA, Vicuna, Koala, WizardLM, Self-Instruct, see the description of the datasets in App. B.1). We provide head-to-head comparisons in terms of win rate, where GPT-4 judges the preferable response (ties are allowed, details in App. B.3).

**Results.** Fig. 5 shows that the responses of our models fine-tuned on the 1k-longest examples of either Alpaca or Evol-Instruct consistently outperform the existing methods across evaluation datasets. In particular, Alpaca-1k-longest is largely preferred over all competitors, and has an average win rate of 46.3% vs. LIMA-1k, with only 28.3% of losses (see Fig. 1). This performance is significant when considering that LIMA has been carefully curated *manually* while our instructions come from a simpler dataset and selected only according to their length. Similarly, Evol-Instruct-1k-longest clearly outperforms LIMA-1k and the full Evol-Instruct-72k, while it has a smaller but consistent advantage over Evol-Instruct-AlpaGasus-1k. We hypothesize that the advantage is smaller on Evol-Instruct because Evol-Instruct contains higher-quality data than Alpaca, thus even selecting examples using GPT-3.5-Turbo scores can find relatively effective training examples. Finally, to exclude the possibility of overfitting to GPT-4 preferences, we repeat this evaluation with PaLM-2 as judge and even in this case our models are largely preferred (see Fig. 11 in App. C.2).

**Role of response length.** As frontier LLMs like GPT-4 might be biased to favor longer responses [41], Fig. 1 additionally illustrates the average length (as number of tokens) of the responses in several datasets described above, as well as the average length of the responses generated by the LLMs fine-tuned on them during evaluation (on 1030 new instructions from the 5 evaluation datasets). As expected, both training and generated answers of Alpaca-1k-longest are longer than those of Alpaca and AlpaGasus. Interestingly, the training examples of LIMA-1k are more than two times longer than those of Alpaca-1k-longest, while the generated responses of the two models are similar. We conclude that the length of the responses is not the main factor for our model being consistently preferred to LIMA-1k.

## 3 How far can we go with 1,000 instructions?

### 3.1 Refining the instructions via introspection

As suggested by Zhou et al., the goal of IFT is to teach LLMs the format to employ when interacting with the users rather than instilling new knowledge. We argue that fine-tuning on rich and detailed instructions may improve the ability of the models to capture deeper semantic structure and logic. Then, we want to refine our 1k-longest instructions to improve the quality of responses of training examples in terms of style, structure and the level of detail. In fact, there is no guarantee that the instructions selected by length also have high quality in terms of structure, glossary and logic.

Given that LLMs are surprisingly good at self-improving [14, 26] and judging [41, 21], we propose using an Oracle LLM for this task, via encouraging it to introspect. In particular, inspired by Chain-of-Thought prompting [35], we prompt the GPT-3.5-Turbo model to produce a brief review of the original response given the instruction, followed by a new

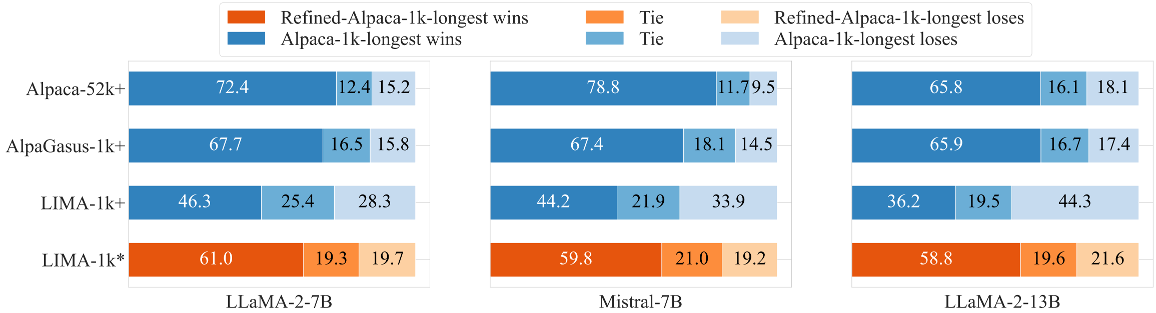


Figure 2: **Refinement via introspection improves instruction-following performance across architectures.** We report the average preference performance (%) across five evaluation sets using GPT-4 as a judge. We show win rate of models with different architectures fine-tuned on Alpaca-1k-longest against Alpaca-52k, AlpaGasus-1k, and LIMA-1k in blue (+ symbol). Additionally we illustrate the improvement brought by our Refined-Alpaca-1k-longest over LIMA-1k, the strongest baseline, in red (* symbol).

response generation process that has access to the original instruction-response pair and the introspection output. The details of the prompt are presented in Fig. 7. Applying this procedure to the 1k-longest instructions of Alpaca we obtain our new IFT dataset named Refined-Alpaca-1k-longest.

### 3.2 Instruction-following evaluation

**Setup.** First, we provide a pairwise comparison between fine-tuning different LLMs on our Refined-1k-longest and baseline datasets, in particular LIMA-1k. Next, to facilitate a unified comparison of all models and position them among existing baselines, we compute their performance on the AlpacaEval 2.0 benchmark [21], which reports the win rate of each model compared to GPT-4-Turbo (according to the preferred responses by GPT-4-Turbo as a judge, see details in App. B.3). This allows us to compare many LLMs, including those reported on the AlpacaEval leaderboard by previous works, more efficiently than with pairwise analyses.

**Head-to-head comparisons.** We compare fine-tuning on our Refined-Alpaca-1k-longest to the Alpaca-1k-longest against Alpaca-52k , AlpaGasus-1k and LIMA-1k in a head-to-head fashion: Fig. 2 reports the average (over the 5 evaluation datasets introduced in Sec. 2.2) preference of GPT-4, repeated for three base models, i.e. Llama-2-7B, Mistral-7B, Llama-2-13B (the corresponding results with PaLM-2 as judge are shown in Fig. 12 in App. C.2). In all cases the models fine-tuned on the plain Alpaca-1k-longest already outperform the baselines with the exception of LIMA-1k for Llama-2-13B. In particular, LIMA-1k makes the strongest existing method: however, when we compare it with our Refined-Alpaca-1k-longest, this last one has a significant advantage over LIMA-1k, with an average win rate of 59.9% across architectures vs the 20.2% of LIMA. This shows the effectiveness of the refinement via introspection on the longest instructions from Alpaca, even when used by different base models.

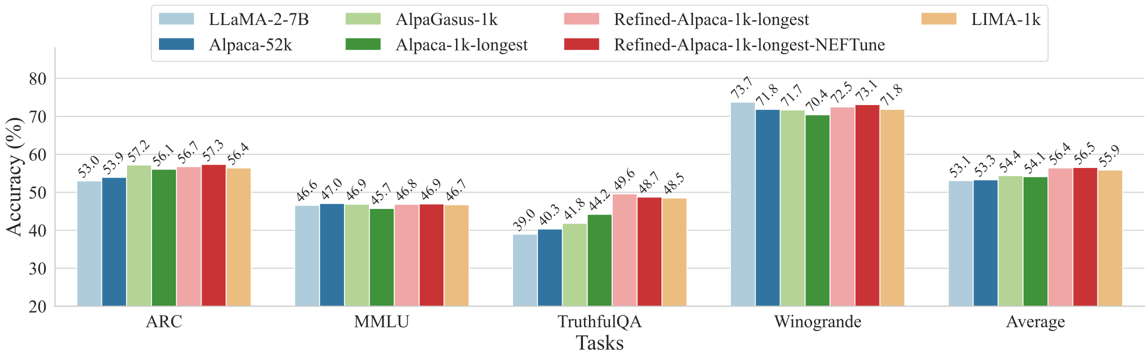


Figure 3: **Open LLM Leaderboard tasks with Llama-2-7B fine-tuned on Alpaca-based datasets and LIMA.**

**AlpacaEval 2.0 evaluation.** In Table 1 we report the results on the AlpacaEval 2.0 benchmark of our models and some baselines copied from the public leaderboard.[1] Moreover, we show the architecture, size of IFT and preference datasets, and average response length for each entry. Among Llama-2-7B models, both LIMA-1k and Alpaca-52k fine-tuned models achieve win rate below 3%, which is outperformed by Alpaca-1k-longest (3.11%). Switching to the instructions refined by introspection (Refined-Alpaca-1k-longest) almost doubles the win rate, achieving 6.00%, which even surpasses the original Llama-2-Chat-7B, fine-tuned with 27k instructions and 3M preference pairs. Since Jain et al. showed that NEFTune, which injects noise on the embedded inputs as augmentation, can improve the performance of IFT, we test it in combination with our dataset: this yields 7.88% win rate, i.e. the second best Llama-2-7B model appearing on the leaderboard, ahead of Llama-2-7B-Evol-Instruct-NEFTune [16] and not far from the 8.20% win rate of Tulu-2-DPO-7B [15]. Overall, these results illustrate how using a simple dataset of 1,000 instructions which did not necessitate any manual curation can compete with more expensive and sophisticated alignment schemes relying on SFT with hundreds of thousands of examples and involving RLHF on up to 3M preference pairs. Moreover, we observe similar behavior with other architectures: for Mistral-7B Alpaca-1k-longest already outperforms the baseline methods, but the refined instructions give the most notable increase (7.13% to 11.74%) in win rate. Similarly, Refined-Alpaca-1k-longest attains the best results for Llama-2-13B. Interestingly, unlike for Llama-2-7B, in these cases the improvements given by NEFTune are marginal ($\leq$ 0.32%), which highlights the importance of the fine-tuning dataset.

**Changing response length does not affect quality.** As shown in Table 1, the LLMs fine-tuned on (Refined-)1k-longest lead to longer generation than most competitors. To test if longer replies are sufficient for higher scores on AlpacaEval 2.0, we increase the maximum number of generated tokens from the default 2048 (used for all baselines as well) to 4096. This makes the average response length of our best Llama-2-7B model (refined dataset with NEFTune) to increase from 1801 to 2478. However, this slightly degrades win rate (-0.05%). Similar small variations can be also observed for other models and architectures (see Table 1). Then, length alone does not significantly influence the results on the benchmark.

---

[1] `https://tatsu-lab.github.io/alpaca_eval/`

### 3.3 Evaluation on factual knowledge benchmarks

In the following, we study how the models trained on small instruction datasets behave in tasks other than instruction following with an LLM as a judge, and the shortcomings it entails. For this, we evaluate them on a subset of the Open LLM benchmark: it includes six datasets, from which we exclude HellaSwag because it contains examples also present in the training set of LIMA-1k (see discussion in App. F) and GSM80K since all models fail to achieve non trivial performance, which assess several abilities of an LLM including commonsense reasoning, multitask knowledge and truthfulness, at various difficulty levels.

Fig. 3 reports the results of the models fine-tuned from Llama-2-7B on the dataset derived from Alpaca and LIMA-1k (the corresponding evaluations for other architectures and Evol-Instruct-based datasets can be found in App. C.4). We observe that, on average over the datasets, IFT on Alpaca-52k yields marginal improvement over the base model, while both AlpaGasus-1k and 1k-longest give around a 1% increase. Significantly better results are achieved by LIMA-1k, with 55.9% vs 53.1% of the base model. However, the two models relying on Alpaca-Refined-1k-longest, without and with NEFTune, are the best performing ones with 56.4% and 56.5% (without and with NEFTune respectively). This suggests that the IFT dataset might have an effect beyond quality of user interactions. In fact, all LLMs are fine-tuned from the same base model, thus we can assume that they have the same factual knowledge, and the different performance is due to how well the alignment phase teaches the model how to follow the right steps to accomplish a given task. We hypothesize that using longer and more detailed instructions, which force the LLM to better capture the semantics of the task at hand, might positively influence the performance on quantitative (e.g. multiple choice questions answering) tasks as those in Open LLM.

## 4 Additional analyses of our models

**Comparison on generations of similar length.** To further support the idea that the length of responses does not explain our models' performance, we artificially increase the response length of the replies from Llama-2-7B models fine-tuned on Alpaca-52k and AlpaGasus-1k.

This extension is achieved by ensuring that the end-of-sentence token does not appear until after the 150th token has been generated. Fig. 6 shows that this adjustment makes both baselines output responses of similar length as our Alpaca-1k-longest.

However, even in this case, both GPT-4 and PaLM-2 judges still significantly prefer our Alpaca-longest-1k model, indicating that artificially increasing the number of generated tokens does not effectively enhance response quality.

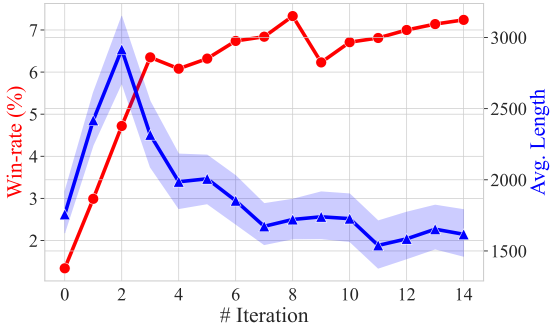


Figure 4: **Performance of the Llama-2-7B model fine-tuned on refined-Alpaca-1k-longest across different epochs.**

**Length and win rate are anticorrelated during fine-tuning.** We track the average length of replies over epochs when fine-tuning for one of our models. As shown in Fig. 4, except for the early stage of fine-tuning, the response length progressively decreases while the win rate keeps improving. This indicates that the model does not simply learn to output long generations from long training examples but also to produce more refined structures.

**Example generations.** In Fig. 18 we provide two examples of completions generated by our Llama-2-7B model fine-tuned on the Alpaca-1k-longest dataset. We see that the LLM provides organic and detailed responses. We provide an extended qualitative comparison to other models in App. E, where one can see that, for example, LIMA can sometimes lead to repetitive outputs while 1k-longest models tend to have a more engaging tone.

## 5 Discussion

**Quality of the instructions in IFT.** Chen et al. and Zhou et al. argue that IFT requires high-quality training examples and use different proxies for quality to create the AlpaGasus and LIMA datasets. However, our experiments demonstrate that a simple heuristic for selecting training instructions, such as the length of the response, leads to better-performing models. It is important to note that length alone is not sufficient. For example, the LIMA training examples are on average twice as long as those in Alpaca-1k-longest. Additionally, we emphasize that length does not necessarily reflect quality, as illustrated by the lower scores given by GPT-3.5-Turbo to the examples in our Alpaca-1k-longest (Fig. 10). This suggests that other factors come into play when determining the effectiveness of IFT datasets. As a result, it remains uncertain which specific components in the fine-tuning dataset are crucial for achieving the best model performance.

**IFT can improve factuality.** Gudibande et al. show the possibility of fine-tuning LLMs to imitate the style of ChatGPT. They achieve this by using ChatGPT's responses as an IFT dataset, which can consist of up to 150 million tokens. Remarkably, both human evaluators and LLM-as-a-judge evaluators rate the responses generated by these fine-tuned models nearly as high as those generated by ChatGPT. However, this fine-tuning approach does not enhance, and in some cases even diminishes, the performance of these models on NLP benchmarks compared to the base model. A similar observation is made by Jha et al., who suggest that LIMA-1k (when used to fine-tune the MPT models from MosaicML) does not yield the same level of performance as Alpaca-52k on tasks that do not rely on automated evaluation by an LLM. In contrast, we demonstrate that IFT can lead to both a stronger preference from various LLMs serving as judges and improved performance on Open LLM tasks. However, it is key to carefully select the instruction dataset for this purpose. The question of systematically constructing optimal IFT datasets remains an open challenge.

**Conclusions.** In this work we have shown that using reply length as a heuristic can effectively pre-select instructions for LLMs alignment in SFT. Moreover, a straightforward refinement step is enough to create a dataset of only 1k instruction-response pairs which yields competitive results compared to complex alignment methods like RLHF and DPO. Thus, this approach constitutes an inexpensive yet strong baseline for future works on alignment. Our analysis also challenges the current understanding of high-quality IFT datasets and their impact on fine-tuned model performance in standard NLP benchmarks.

## Reproducibility Statement

We are going to make our code, datasets, and fine-tuned models public soon. The reproducibility of our work is further boosted by several open-source projects, such as FastChat, lm-evaluation-harness (Open LLM benchmark), and AlpacaEval, which provide the standardized training and evaluation code we relied on.

## Broader Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## References

[1] Rohan Anil et al. "Palm 2 technical report". In: *arXiv preprint arXiv:2305.10403* (2023).

[2] Amanda Askell et al. "A general language assistant as a laboratory for alignment". In: *arXiv preprint arXiv:2112.00861* (2021).

[3] Yuntao Bai et al. "Constitutional ai: Harmlessness from ai feedback". In: *arXiv preprint arXiv:2212.08073* (2022).

[4] Yuntao Bai et al. "Training a helpful and harmless assistant with reinforcement learning from human feedback". In: *arXiv preprint arXiv:2204.05862* (2022).

[5] Jason Baumgartner et al. "The pushshift reddit dataset". In: *Proceedings of the international AAAI conference on web and social media*. Vol. 14. 2020, pp. 830–839.

[6] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[7] Lichang Chen et al. "Alpagasus: Training a better alpaca with fewer data". In: *arXiv preprint arXiv:2307.08701* (2023).

[8] Wei-Lin Chiang et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. Mar. 2023. URL: `https://lmsys.org/blog/2023-03-30-vicuna/`.

[9] Hyung Won Chung et al. "Scaling instruction-finetuned language models". In: *arXiv preprint arXiv:2210.11416* (2022).

[10] Peter Clark et al. "Think you have solved question answering? try arc, the ai2 reasoning challenge". In: *arXiv preprint arXiv:1803.05457* (2018).

[11] Xinyang Geng et al. *Koala: A Dialogue Model for Academic Research*. Blog post. Apr. 2023. URL: `https://bair.berkeley.edu/blog/2023/04/03/koala/` (visited on 04/03/2023).

[12] Arnav Gudibande et al. "The false promise of imitating proprietary llms". In: *arXiv preprint arXiv:2305.15717* (2023).

[13] Dan Hendrycks et al. "Measuring Massive Multitask Language Understanding". In: *International Conference on Learning Representations*. 2020.

[14] Jiaxin Huang et al. "Large language models can self-improve". In: *arXiv preprint arXiv:2210.11610* (2022).

[15] Hamish Ivison et al. "Camels in a changing climate: Enhancing lm adaptation with tulu 2". In: *arXiv preprint arXiv:2311.10702* (2023).

[16] Neel Jain et al. "NEFTune: Noisy Embeddings Improve Instruction Finetuning". In: *arXiv preprint arXiv:2310.05914* (2023).

[17] Aditi Jha et al. "LIMIT: Less Is More for Instruction Tuning Across Evaluation Paradigms". In: *arXiv preprint arXiv:2311.13133* (2023).

[18] Albert Q Jiang et al. "Mistral 7B". In: *arXiv preprint arXiv:2310.06825* (2023).

[19] Harrison Lee et al. "Rlaif: Scaling reinforcement learning from human feedback with ai feedback". In: *arXiv preprint arXiv:2309.00267* (2023).

[20] Ming Li et al. "Reflection-Tuning: Data Recycling Improves LLM Instruction-Tuning". In: *arXiv preprint arXiv:2310.11716* (2023).

[21] Xuechen Li et al. *AlpacaEval: An Automatic Evaluator of Instruction-following Models*. `https://github.com/tatsu-lab/alpaca_eval`. 2023.

[22] Stephanie Lin, Jacob Hilton, and Owain Evans. "TruthfulQA: Measuring How Models Mimic Human Falsehoods". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022, pp. 3214–3252.

[23] MosaicML. *Introducing MPT-7B: A New Standard for Open-Source, Commercially Usable LLMs*. `www.mosaicml.com/blog/mpt-7b`, accessed: 2023-08-02. 2023. URL: `www.mosaicml.com/blog/mpt-7b` (visited on 05/05/2023).

[24] OpenAI. "GPT-4 Technical Report". In: *arXiv preprint arXiv:2303.08774* (2023).

[25] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744.

[26] Liangming Pan et al. "Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies". In: *arXiv preprint arXiv:2308.03188* (2023).

[27] Baolin Peng et al. "Instruction tuning with gpt-4". In: *arXiv preprint arXiv:2304.03277* (2023).

[28] Rafael Rafailov et al. "Direct preference optimization: Your language model is secretly a reward model". In: *arXiv preprint arXiv:2305.18290* (2023).

[29] Keisuke Sakaguchi et al. "Winogrande: An adversarial winograd schema challenge at scale". In: *Communications of the ACM* 64.9 (2021), pp. 99–106.

[30] Prasann Singhal et al. "A long way to go: Investigating length correlations in rlhf". In: *arXiv preprint arXiv:2310.03716* (2023).

[31] Rohan Taori et al. *Stanford Alpaca: An Instruction-following LLaMA model.* `https://github.com/tatsu-lab/stanford_alpaca`. 2023.

[32] Hugo Touvron et al. “Llama 2: Open foundation and fine-tuned chat models”. In: *arXiv preprint arXiv:2307.09288* (2023).

[33] Yizhong Wang et al. “How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources”. In: *arXiv preprint arXiv:2306.04751* (2023).

[34] Yizhong Wang et al. “Self-instruct: Aligning language model with self generated instructions”. In: *arXiv preprint arXiv:2212.10560* (2022).

[35] Jason Wei et al. “Chain-of-thought prompting elicits reasoning in large language models”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 24824–24837.

[36] Can Xu et al. “Wizardlm: Empowering large language models to follow complex instructions”. In: *arXiv preprint arXiv:2304.12244* (2023).

[37] Hanwei Xu et al. “ZeroPrompt: scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization”. In: *arXiv preprint arXiv:2201.06910* (2022).

[38] Weizhe Yuan et al. “Self-Rewarding Language Models”. In: *arXiv preprint arXiv:2401.10020* (2024).

[39] Rowan Zellers et al. “HellaSwag: Can a Machine Really Finish Your Sentence?” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4791–4800.

[40] Zhiyuan Zeng et al. “Evaluating large language models at evaluating instruction following”. In: *arXiv preprint arXiv:2310.07641* (2023).

[41] Lianmin Zheng et al. “Judging LLM-as-a-judge with MT-Bench and Chatbot Arena”. In: *arXiv preprint arXiv:2306.05685* (2023).

[42] Chunting Zhou et al. “LIMA: Less Is More for Alignment”. In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

## Appendix A. Related work

**Instruction fine-tuning of LLMs.** Since pre-trained LLMs usually do not accurately understand user intents and provide coherent and beneficial responses, an instruction fine-tuning stage is necessary [25, 4]. Diversity of demonstrations and tasks [9, 37] plays a pivotal role in enhancing the instruction-following performance of LMs. InstructGPT [25] first demonstrated how to achieve impressive performance in handling open-ended queries by fine-tuning GPT-3 models [6] with RLHF, which led to the release of ChatGPT. Subsequently, the community attempted to replicate the exceptional performance of proprietary models [33, 36, 8], but Gudibande et al. show that it might be easy to mimic the style but not the factuality of closed-source LLMs. Singhal et al. identify a strong correlation between response length and reward when doing RLHF, implying that optimizing response length might be an implicit goal of RLHF. Also, Yuan et al. show that their self-improved reward model based on DPO encourages more verbose responses.

**Data selection for IFT.** The community has focused on creating IFT datasets of high quality. As one of the pioneering works, Alpaca [31] collects 52k interactions with the text-davinci-003 model using techniques from Self-Instruct [34]. With more powerful models as the teacher, GPT-4-LLM [27] is shown to have superior reasoning and math skills than Alpaca. Xu et al. propose using In-depth Evolving and In-breadth Evolving prompting strategies to generate interaction instances of varying complexity. However, direct distillation from language models without careful screening inevitably introduces demonstrations with incorrect or ill-favored answers, such as hallucinated content as opposed to the facts. To filter these cases out, AlpaGasus [7] measures the quality of each demonstration using a powerful LLM (GPT-3.5-Turbo) as a scorer. To better reflect human intentions, LIMA [42] relies on community forums and human labor to curate 1,000 demonstrations with an emphasis on quality and diversity, achieving strong instruction-following ability, surpassing some proprietary LLMs. They also formulate the *Superficial Alignment Hypothesis*: the general-purpose capabilities of an LLM mostly come from pre-training, and instruction tuning only guides the LLM to mimic the style, persona, and instruction adherence of desired outputs. Similarly to LIMA, Touvron et al. note that fewer (in the order of tens of thousands) but higher-quality examples annotated by their own vendors significantly improve their Llama-2-Chat models.

## Appendix B. Experimental details

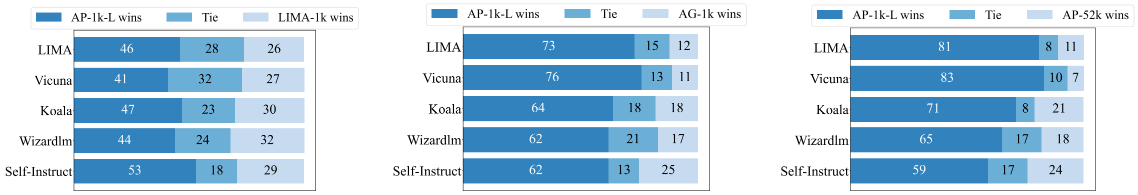
### B.1 IFT datasets

This section contains a list of instruction fine-tuning datasets that appear in our experiments, along with relevant information.

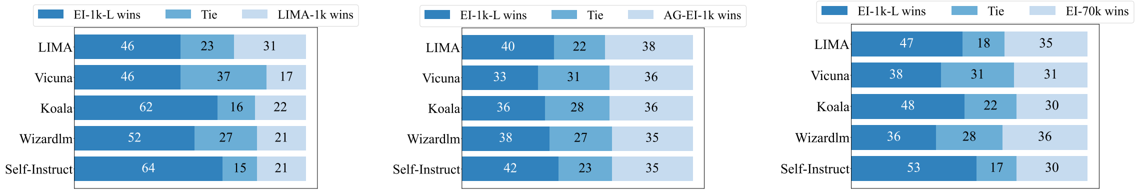
- **Alpaca** [31] contains 52k synthetic examples generated by explicitly giving the requirement instruction generation to the text-davinci-003 model. Although the created dataset is intended to be varied, a thorough examination reveals that it is heavily US-centric. It is also discovered that the original version has numerous issues that affect its quality and suitability for training a trustworthy language model. These issues

Table 1: **Preference evaluation results on AlpacaEval 2.0.** The evaluator used to measure instruction-following performance comprehensively considers quality, price, time, variance, and length bias. For our models, if not specified otherwise, we use a limit of 2048 tokens for generation. * denotes results which are directly copied from the AlpacaEval 2.0 leaderboard.

| Models | # IFT Data | # Pref. Data | Win Rate | Avg. Length |
|---|---|---|---|---|
| **Notable baselines** | | | | |
| GPT-4-Turbo* | ? | ? | 50.0 | 2049 |
| Alpaca-7B* | 52k | 0 | 2.59 | 396 |
| Vicuna-7B* | 70k | 0 | 4.16 | 1044 |
| Base model: **Llama-2-7B** | | | | |
| Llama-2-Chat-7B* | 27k | 3M | 4.96 | 1479 |
| + Evol70k-NEFTune* | 97k | 3M | 7.60 | 1612 |
| Tulu-2-DPO-7B* | 326k | 64k | 8.20 | 1663 |
| AlpaGasus-1k | 1k | 0 | 2.69 | 745 |
| LIMA-1k | 1k | 0 | 2.74 | 1360 |
| Alpaca-52k | 52k | 0 | 2.74 | 586 |
| Alpaca-1k-longest | 1k | 0 | 3.16 | 1810 |
| + max gen. 2048 → 4096 | 1k | 0 | 3.11 | 2290 |
| Evol-Instruct-70k | 70k | 0 | 3.44 | 850 |
| Evol-Instruct-1k-longest | 1k | 0 | 4.09 | 1866 |
| + max gen. 2048 → 4096 | 1k | 0 | 4.16 | 2486 |
| Evol-Instruct-AlpaGasus-1k | 1k | 0 | 4.32 | 1156 |
| Refined-Alpaca-1k-longest | 1k | 0 | 6.00 | 1732 |
| + max gen. 2048 → 4096 | 1k | 0 | 6.03 | 2326 |
| + NEFTune | 1k | 0 | 7.88 | 1801 |
| + NEFTune + 2048 → 4096 | 1k | 0 | 7.83 | 2478 |
| Base model: **Mistral-7B** | | | | |
| Alpaca-52k | 52k | 0 | 3.42 | 450 |
| AlpaGasus-1k | 1k | 0 | 4.91 | 502 |
| LIMA-1k | 1k | 0 | 6.76 | 1197 |
| Alpaca-1k-longest | 1k | 0 | 7.13 | 937 |
| Refined-Alpaca-1k-longest | 1k | 0 | 11.74 | 1170 |
| + max gen. 2048 → 4096 | 1k | 0 | 11.76 | 1330 |
| + NEFTune | 1k | 0 | 11.94 | 1199 |
| Base model: **Llama-2-13B** | | | | |
| Alpaca-52k | 52k | 0 | 3.90 | 556 |
| Alpaca-1k-longest | 1k | 0 | 4.80 | 1104 |
| AlpaGasus-1k | 1k | 0 | 4.87 | 540 |
| LIMA-1k | 1k | 0 | 5.64 | 1097 |
| Refined-Alpaca-1k-longest | 1k | 0 | 8.44 | 1646 |
| + max gen. 2048 → 4096 | 1k | 0 | 8.30 | 2244 |
| + NEFTune | 1k | 0 | 8.76 | 1582 |

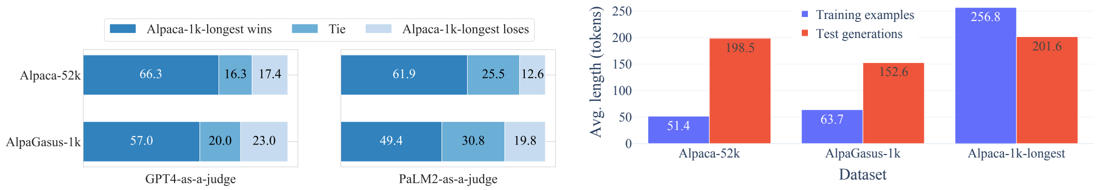


(a) Alpaca-1k-longest vs. LIMA-1k (b) Alpaca-1k-longest vs. AlpaGasus-1k (c) Alpaca-1k-longest vs. Alpaca-52k



(d) Evol-Instruct-1k-longest vs. LIMA-1k (e) Evol-Instr.-1k-longest vs. EvolInstr.-AlpaGasus-1k (f) Evol-Instruct-1k-longest vs. Evol-Instruct-70k

Figure 5: **Detailed preference evaluation (in %).** For each pair of LLMs we report the win rate on 5 datasets (LIMA, Vicuna, Koala, WizardLM, Self-Instruct) according to GPT-4-as-a-judge. **Top:** we compare fine-tuning on Alpaca-1k-longest (AP-1k-L) to Alpaca-52k, AlpaGasus-1k, and LIMA-1k. **Bottom:** we compare fine-tuning on Evol-Instruct-1k-longest (EI-1k-L) to Evol-Instruct-70k, Evol-Instruct-AlpaGasus-1k (i.e. using the method of Chen et al. to subsample Evol-Instruct-70k), and LIMA-1k. Our datasets of long responses consistently lead to higher preferences (higher win rate) than the existing methods.



(a) Head-to-head comparisons with two different LLM judges (b) Average number of tokens in responses

Figure 6: **Preference evaluation (in %) on generations of similar length.** Forcing the models fine-tuned on Alpaca-52k and AlpaGasus-1k to generate longer response (b) does not improve their win rate vs Alpaca-1k-longest (a).

includes hallucinations, merged instructions, empty outputs, impractical instructions like generating images, wrong answers, and non-sensical instructions, etc.

- **AlpaGasus-1k/9k** [7] contains 1k/9k high-quality examples filtered from the original Alpaca-52k dataset. It suggests implementing data selection by means of strong LLMs, such as ChatGPT, to automatically detect and filter out low-quality data. By doing this, they leave out problematic samples, which endanger the effectiveness of refined models.

- **Recycled-Alpaca** [20] comprises of 52k enhanced examples based on Alpaca-52k. Given the initial basic dataset, a high-quality version of each data point is generated using an Oracle model, such as chatGPT. However, a typical issue with using LLMs as judges is the inability to produce different results. To address this potential issue, inspired by Chain-of-Thought prompting, numerous particular criterias are proposed for the Oracle model to follow, and then strong target LMs respond to those precise requirements with critical responses. The responses to these criteria can then be used as bridges (chains of thought) to create new, satisfied instruction-response combinations.

- **LIMA** [42] collects a dataset of 1000 prompts and responses for training, with the outputs stylistically aligned but the inputs different. It also provides an open-source test set of 300 prompts and a development set of 50. Curated from multiple sources, LIMA is primarily divided among community Q&A websites like Stack Exchange, wikiHow, and the Pushshift Reddit Dataset [5], as well as manually created examples. In terms of Q&A communities, frequently upvoted answers on Reddit are typically hilarious or trolling, requiring more manual effort to align responses that adhere to the proper style. In contrast, answers from Stack Exchange and wikiHow are well-aligned with the behavior of a helpful chat assistant. Human-authored examples are used to boost the diversity of dataset.

- **Evol-Instruct (WizardLM)** [36] contains 70k training examples with varying complexity and 218 test instances. The training dataset is initially initialized using Alpaca's 52k instruction data. After iteratively completing $M = 4$ evolutions, the dataset has 250k instructions. More specifically, for each instruction in each round of evolution, one evolving prompt from total six new prompts (i.e., five from in-depth evolving and one from in-breadth evolving) is selected with equal probability. Then, ChatGPT is used to produce answers for each instruction, yielding $52 \times 4 \times 3 = 624$k instruction-response pairs. Finally, the Evol-Instruct dataset is created by picking a subset of 70k instructions. 218 test instructions are collected from diverse sources including online opensource projects, platforms, and forums. This test set is primarily a union of 29 distinct skills identified among real-world human instructions, such as Coding Generation & Debugging, Reasoning, Math, Writing, Complex Formats, Extensive Disciplines, and so on.

- **Vicuna** [8] divides 80 test instructions into 8 question categories, including Fermi problems, commonsense, roleplay scenarios, coding/math/writing tasks, counterfactual, knowledge, and generic, to evaluate various aspects of a chatbot's performance. Vicuna has been demonstrated to mostly include instructions of low difficulty and complexity [36].

- **Self-Instruct** [34] has 252 human-authored test instructions with 1 handcrafted output per instruction. Self-Instrction test set is created to better reflect the practical

Table 2: Details of training hyperparameters for all experiments.

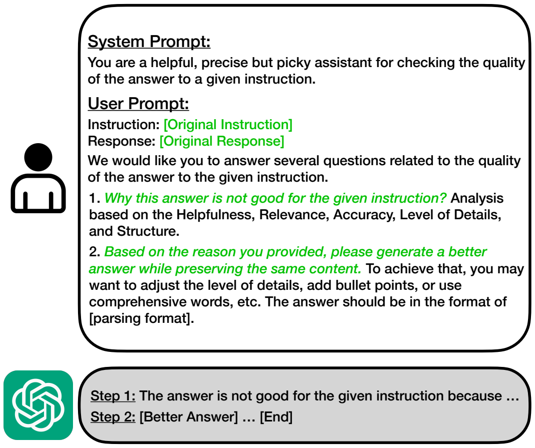
| Datasets | Data Size | # GPUs | Epochs | LR | LR Scheduler | Batch Size | Context Win. Len. | WD | Warmup Rate |
|---|---|---|---|---|---|---|---|---|---|
| ***Llama-2-7B*** | | | | | | | | | |
| Evol-Instruct-70k | 70k | 4 | 3 | 2e-5 | Cosine | 128 | 512 | 0.0 | 0.3 |
| Alpaca-52k | 52k | 4 | 3 | 2e-5 | Cosine | 128 | 512 | 0.0 | 0.3 |
| AlpaGasus-9k | 9k | 4 | 3 | 2e-5 | Cosine | 128 | 512 | 0.0 | 0.3 |
| Alpaca-9k-longest | 9k | 4 | 3 | 2e-5 | Cosine | 128 | 512 | 0.0 | 0.3 |
| AlpaGasus-1k | 1k | 4 | 15 | 1e-5 | Linear | 128 | 2048 | 0.1 | 0.0 |
| LIMA-1k | 1k | 4 | 15 | 1e-5 | Linear | 128 | 2048 | 0.1 | 0.0 |
| Alpaca-1k-longest | 1k | 4 | 15 | 1e-5 | Linear | 128 | 2048 | 0.1 | 0.0 |
| Evol-Instruct-AlpaGasus-1k | 1k | 4 | 15 | 1e-5 | Linear | 128 | 2048 | 0.1 | 0.0 |
| Evol-Instruct-1k-longest | 1k | 4 | 15 | 1e-5 | Linear | 128 | 2048 | 0.1 | 0.0 |
| ***Mistral-7B-v0.1*** | | | | | | | | | |
| Alpaca-52k | 52k | 4 | 3 | 4e-6 | Cosine | 128 | 512 | 0.0 | 0.3 |
| AlpaGasus-1k | 1k | 4 | 15 | 2e-6 | Linear | 128 | 2048 | 0.1 | 0.0 |
| LIMA-1k | 1k | 4 | 15 | 2e-6 | Linear | 128 | 2048 | 0.1 | 0.0 |
| Alpaca-1k-longest | 1k | 4 | 15 | 2e-6 | Linear | 128 | 2048 | 0.1 | 0.0 |
| ***Llama-2-13B*** | | | | | | | | | |
| Alpaca-52k | 52k | 4 | 5 | 1e-5 | Cosine | 128 | 512 | 0.0 | 0.3 |
| AlpaGasus-1k | 1k | 4 | 15 | 1e-5 | Linear | 128 | 2048 | 0.1 | 0.0 |
| LIMA-1k | 1k | 4 | 15 | 1e-5 | Linear | 128 | 2048 | 0.1 | 0.0 |
| Alpaca-1k-longest | 1k | 4 | 15 | 1e-5 | Linear | 128 | 2048 | 0.1 | 0.0 |

value of instruction-following models. The authors were motivated to curate instructions of different domains ranging from email writing and social media to productivity tools and programming. Authors also deliberately diversify the styles and formats of tasks, such as including instructions of different lengths and considering input/output that takes the form of bullet points, tables, codes, equations, etc.

- **Koala** [11] consists of 180 real user queries that were posted on the Internet. These user-initiated queries cover a wide range of subjects, typically have a conversational tone, and are probably more indicative of the practical applications of chat-based systems. Queries with a BLEU score of more than 20% with any example from our training set are filtered away in order to reduce the possibility of test-set leaking. Prompts pertaining to code and languages other than English are also excluded because the crowd workers, who make up the pool of raters, are unable to accurately examine the answers to these questions.

### B.2 Training hyperparameters

This section lists the hyperparameters necessary for reproducing our work. Our experiments are built upon FastChat framework [41]. In particular, we follow the training configuration as reported in Taori et al. to fine-tune the base model on full IFT datasets like Alpaca-52k and Evol-Instruct-70k, while we refer to LIMA [42] and AlpaGasus [7] when fine-tuning the base model on IFT datasets with 1k and 9k training examples, respectively. In addition to existing experimental setups in prior work, we adopt the recently proposed NEFTune augmentation for our (Refined-)Alpaca-1k-longest experiments. We have `neftune_noise_level` set to 5 for Llama-2-7B, while for Mistral-7B-v0.1 and Llama-2-13B it is set to 3. It should be noted that we use 4 × 40G A100 to finetune Llama-2-7B and 4 × 80G A100 to finetune Mistral-7B-v0.1 and Llama-2-13B. We present the detailed training hyperparameters in Table 2.



System Prompt:

You are a helpful, precise but picky assistant for checking the quality of the answer to a given instruction.

User Prompt:

Instruction: [Original Instruction]

Response: [Original Response]

We would like you to answer several questions related to the quality of the answer to the given instruction.

1. *Why this answer is not good for the given instruction?* Analysis based on the Helpfulness, Relevance, Accuracy, Level of Details, and Structure.

2. *Based on the reason you provided, please generate a better answer while preserving the same content.* To achieve that, you may want to adjust the level of details, add bullet points, or use comprehensive words, etc. The answer should be in the format of [parsing format].

Step 1: The answer is not good for the given instruction because ...

Step 2: [Better Answer] ... [End]

Figure 7: **The template of introspection prompting** used to refine the responses in terms of style, structure, and the level of details.

### B.3 Evaluation details

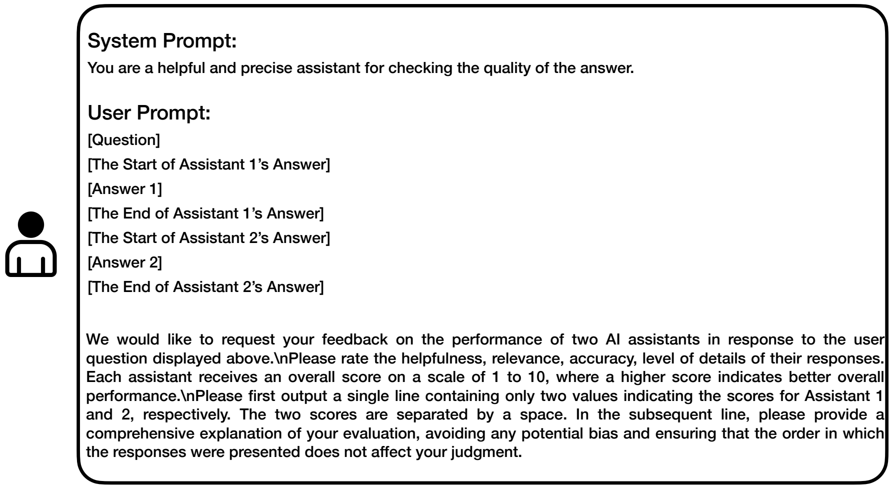
**Evaluation metrics for head-to-head comparisons.** Since automated evaluation based on powerful LLMs offers superior scalability, explainability and reproducibility than human evaluation, we apply an LLM that has high human preference agreement as the judge to evaluate the target model (e.g., Llama-2-7B fine-tuned on Alpaca-1k-longest) and compare it with a baseline model (e.g., GPT-4-Turbo). We append both models' outputs in the input instruction to the LLM judge, followed by a request to the judge which prompts the model to rate the responses with a score between 1 and 10. Since there exists position bias within LLM-based automated evaluation [41], we run evaluation on both orders (i.e., placing the response of the target model before/after the baseline model's response) and calculate the win rate (tie is allowed).

**LLM-as-a-judge.** Given their good agreement with human evaluators shown in LLMBar [40], we decide to adopt GPT-4 (i.e., GPT-4-0613) and PaLM2 (i.e., text-bison@002) as the LLM judges to appropriately assess the instruction-following performance of instruction-tuned models.

**Evaluation prompt for GPT4- and PaLM2-as-a-judge.** We adopt the same evaluation prompt for both GPT4- and PaLM2-as-a-judge as what AlpaGasus [7] uses, which is also the prompt for evaluation used in the original Vicuna work [8]. We provide the detailed form of the prompt in Fig. 8.

**AlpacaEval 2.0.** We apply the AlpacaEval 2.0 benchmark in our experiments since it provides transferable comparisons, which is impossible to achieve in head-to-head evaluation. AlpacaEval 2.0 provides 805 test instructions, on which we generate new responses using the target model, and then calculate the score by competing with the baseline model (i.e., GPT-4-Turbo) judged by a designated automatic evaluator.

**Open LLM Leaderboard.** Several multiclass classification datasets are used to compute the models ranking: ARC [10], MMLU [13], TruthfulQA [22], Winogrande [29], HellaSwag [39]. The combination of datasets widely measures an LLM's capacity to react to



**System Prompt:**

You are a helpful and precise assistant for checking the quality of the answer.

**User Prompt:**

[Question]
[The Start of Assistant 1's Answer]
[Answer 1]
[The End of Assistant 1's Answer]
[The Start of Assistant 2's Answer]
[Answer 2]
[The End of Assistant 2's Answer]

We would like to request your feedback on the performance of two AI assistants in response to the user question displayed above.\nPlease rate the helpfulness, relevance, accuracy, level of details of their responses. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance.\nPlease first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores are separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

Figure 8: The prompt template for evaluation using GPT-4 or PaLM2 as the judge.

factual queries and reasoning challenges, and we use this benchmark to compare the model's change in factual capabilities before and after instruction fine-tuning.

## Appendix C. Additional results

### C.1 Scores of Alpaca-1k-longest according to GPT-3.5-Turbo

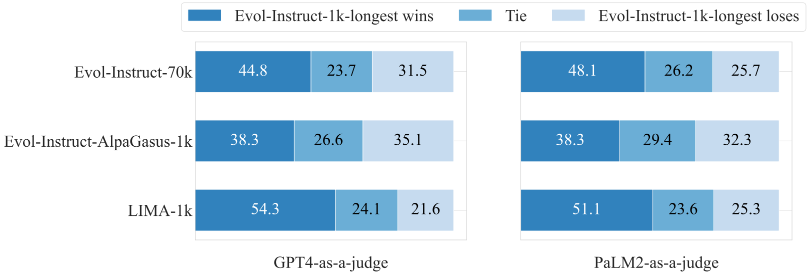
In Fig. 10 we show the score distribution from Chen et al. for the 1k longest instructions compared to those of AlpaGasus-1k (i.e. that highest scoring ones): we see that the overlap between the two datasets is minimal, and most of the longest instructions have score of 3.5. Interestingly, this suggests that GPT-3.5-Turbo prefers longer responses when used as a judge, e.g. in the AlpacaEval 2.0 benchmark, while favors different features when asked to score the quality of the instruction-response pairs in Alpaca.

### C.2 PaLM-2-as-a-judge details

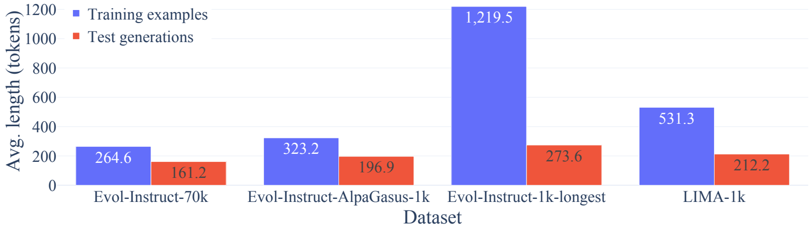
We present detailed preference evaluation results using PaLM2-as-a-judge on an array of Llama-2-7B-based models in Fig. 11. Moreover, we show the improvement given by the refined dataset in Fig. 12. In both cases the observations are consistent with what obtained with GPT-4 as judge (see Fig. 5 and Fig. 2 respectively).

### C.3 Preference evaluation on Mistral-7B and LLaMA-2-13B

This section contains the average preference evaluation results on Mistral-7B model and Llama-2-13B model over 5 evaluation sets (i.e., LIMA, Vicuna, Koala, WizardLM, and Self-Instruct) as shown in Fig. 13 and Fig. 14.



(a) Head-to-head comparisons (in %) with different LLM-judges



(b) Avg. number of tokens in responses

Figure 9: **Effect of using long instructions from Evol-Instruct-70.** We finetune LLaMA-2-7B models on Evol-Instruct-70k [36], Evol-Instruct-AlpaGasus-1k, LIMA-1k and our Evol-Instruct-1k-longest. **(a)** Evol-Instruct-1k-longest beats three baselines in instruction-following performance according to both GPT-4 and PaLM-2 as judges. **(b)** Evol-Instruct-1k-longest leads to the largest average response length at test time. Interestingly, the average length of training responses for Evol-Instruct-1k-longest is more than twice as long as that of LIMA-1k, but the average length of Evol-Instruct-1k-longest at test time only increases by 28.9%.

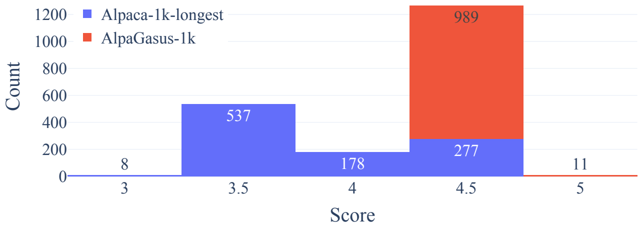
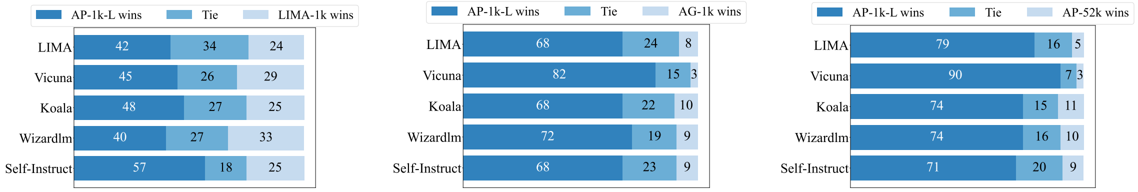
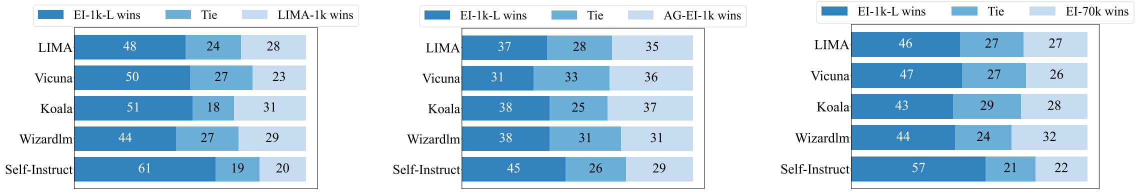


Figure 10: **Quality of training examples.** We show the distribution of the scores, as measured by GPT-3.5-Turbo, of the AlpaGasus-1k and Alpaca-1k-longest datasets (scale of scores is 1-5).



(a) Alpaca-1k-longest vs. LIMA-1k (b) Alpaca-1k-longest vs. AlpaGasus-1k (c) Alpaca-1k-longest vs. Alpaca-52k



(d) Evol-Instruct-longest vs. LIMA-1k (e) EI-1k-longest vs. EI-AlpaGasus-1k (f) EI-1k-longest vs. Evol-Instruct-70k

Figure 11: **Detailed preference evaluation (in %, with PaLM-2-as-a-judge).** For each pair of LLMs we report the win rate on 5 datasets (LIMA, Vicuna, Koala, WizardLM, Self-Instruct) according to PaLM-2-as-a-judge. **Top:** we compare fine-tuning on Alpaca-1k-longest (AP-1k-L) to Alpaca-52k, AlpaGasus-1k, and LIMA-1k. **Bottom:** we compare fine-tuning on Evol-Instruct-1k-longest (EI-1k-L) to Evol-Instruct-70k, Evol-Instruct-AlpaGasus-1k (i.e. using the method of Chen et al. to subsample Evol-Instruct-70k), and LIMA-1k. Our datasets of long responses consistently lead to higher preferences (higher win rate) on average than the existing methods.

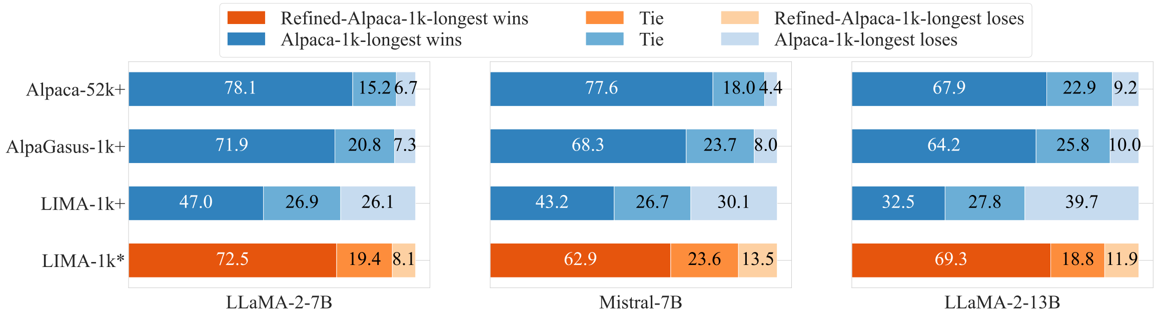
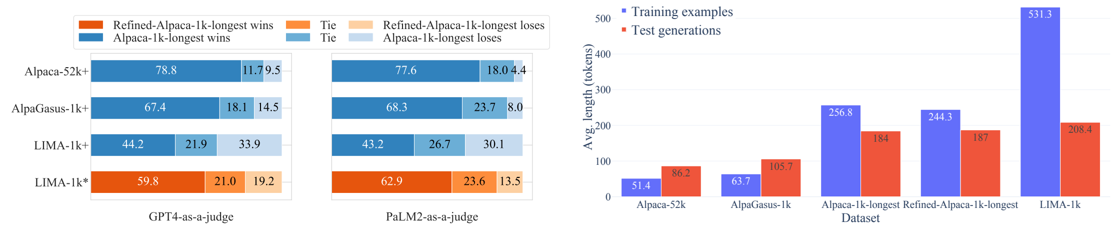


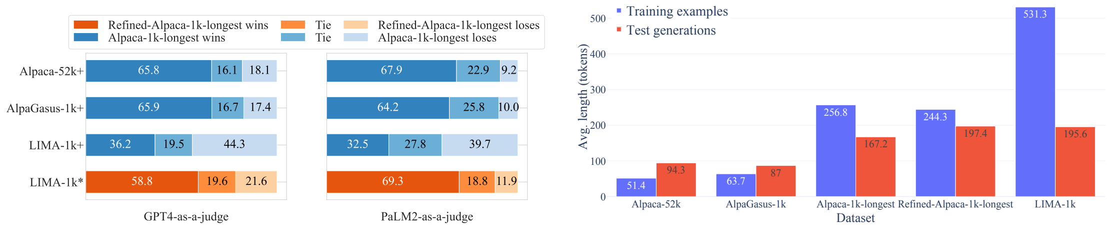
Figure 12: **Refinement via introspection improves instruction-following performance across architectures (PaLM-2-as-a-judge).** We report the average preference performance (%) across 5 evaluation sets. We show win rate of models with different architectures fine-tuned on Alpaca-1k-longest against Alpaca-52k, AlpaGasus-1k and LIMA-1k in blue (+ symbol). Additionally we illustrate the improvement brought by our Refined-Alpaca-1k-longest over LIMA-1k, the strongest baseline, in red (* symbol).



(a) Head-to-head comparisons with different LLM-judges

(b) Avg. number of tokens in responses

Figure 13: **Selecting the longest responses leads to a strong IFT dataset (Mistral-7B).** We fine-tune Mistral-7B models on Alpaca-52k [31], AlpaGasus-1k [7], LIMA-1k [42] and our Alpaca-1k-longest. We show win rate of models with different architectures fine-tuned on Alpaca-1k-longest against Alpaca-52k, AlpaGasus-1k and LIMA-1k in blue (+ symbol). Additionally we illustrate the improvement brought by our Refined-Alpaca-1k-longest over LIMA-1k, the strongest baseline, in red (* symbol). **(a)** Alpaca-1k-longest beats three baselines in instruction-following performance according to both GPT-4 and PaLM-2 as judges. And Refined-Alpaca-1k-longest further enhance the instruction fine-tuning performance as demonstrated by larger win-rates given by both LLM-judges. **(b)** Alpaca-1k-longest leads to an average response length at test time higher than Alpaca-52k and AlpaGasus-1k, but smaller than LIMA-1k, which demonstrates the LLM-judges' preference on our models is induced by better response quality instead of length bias.



(a) Head-to-head comparisons with different LLM-judges

(b) Avg. number of tokens in responses

Figure 14: **Selecting the longest responses leads to a strong IFT dataset (Llama-2-13B).** We fine-tune Llama-2-13B models on Alpaca-52k [31], AlpaGasus-1k [7], LIMA-1k [42] and our Alpaca-1k-longest. We show win rate of models with different architectures fine-tuned on Alpaca-1k-longest against Alpaca-52k, AlpaGasus-1k and LIMA-1k in blue (+ symbol). Additionally we illustrate the improvement brought by our Refined-Alpaca-1k-longest over LIMA-1k, the strongest baseline, in red (* symbol). **(a)** Alpaca-1k-longest beats Alpaca-52k and AlpaGasus-1k in instruction-following performance according to both GPT-4 and PaLM-2 as judges, but underperforms LIMA-1k. However, Refined-Alpaca-1k-longest significantly enhance the instruction fine-tuning performance of the model, surpassing LIMA-1k. **(b)** Alpaca-1k-longest leads to an average response length at test time higher than Alpaca-52k and AlpaGasus-1k, but smaller than LIMA-1k. And the average response length of Refined-Alpaca-1klongest at test time is comparable to that of LIMA-1k, which demonstrates the LLM-judges' preference on Refined-Alpaca-1k-longest is induced by better response quality instead of length bias.

### C.4 Open LLM results on Mistral-7B, LLaMA-2-13B, Evol-Instruct-70k

This section contains the evaluation results of Mistral-7B model and Llama-2-13B model on (Fig. 15) and of Llama-2-7B fine-tuned on Evol-Instruct-based datasets (Fig. 16) over the Open LLM benchmark.

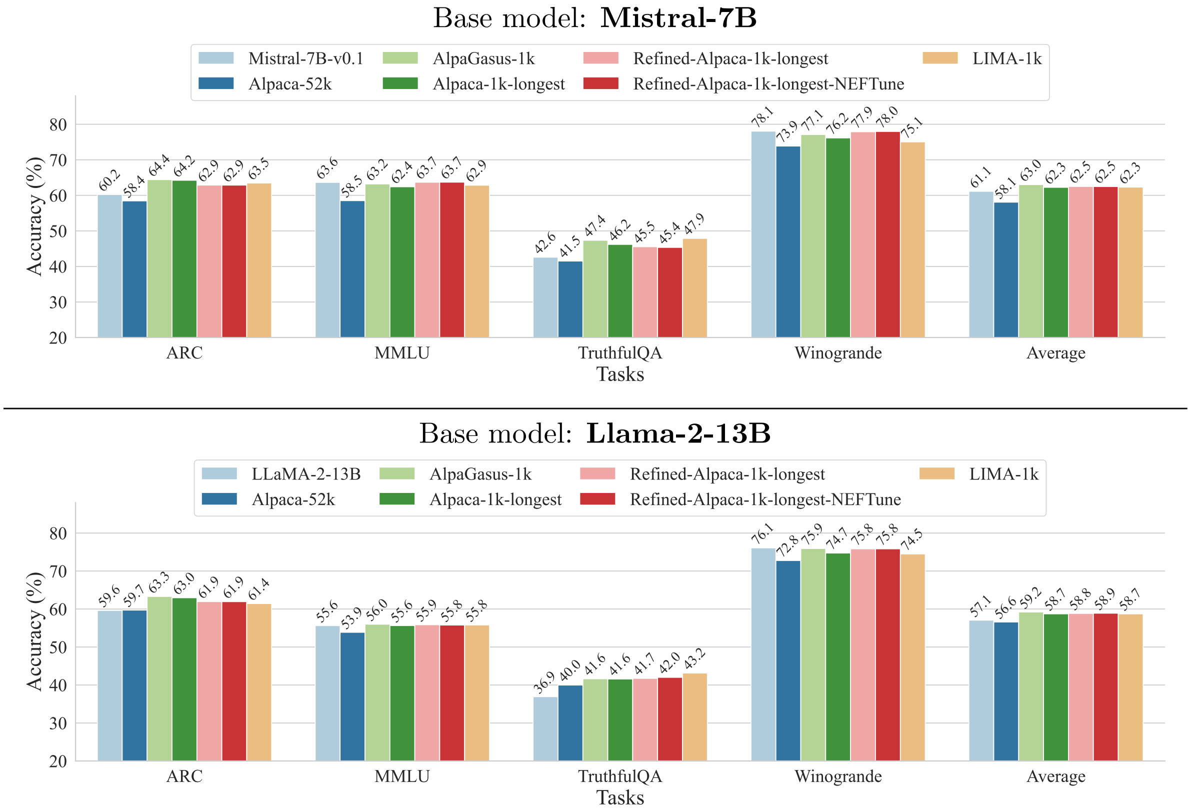


Figure 15: **Open LLM Leaderboard tasks with Mistral-7B and Llama-2-13B fine-tuned on Alpaca-based datasets and LIMA.** The model fine-tuned on Alpaca-1k-longest achieves comparable performance to that of LIMA-1k and significantly outperforms both base models and Alpaca-52k on average, showing that the performance gain on instruction-following capability does not compromise factuality.

## Appendix D. Comparison to additional baselines

### D.1 AlpaGasus-9k

In this section, we validate the advantage of length heuristics by comparing Alpaca-9k-longest with AlpaGasus-9k, which is the best filtered subset from Alpaca-52k in the AlpaGasus paper [7]. The detailed experimental results are shown in Fig. 17(a), where Alpaca-9k-longest consistently outperforms AlpaGasus-9k in 5 evaluation sets. We further show comparisons between Alpaca-1k-longest and AlpaGasus-9k in Fig. 17(b), which also sup-

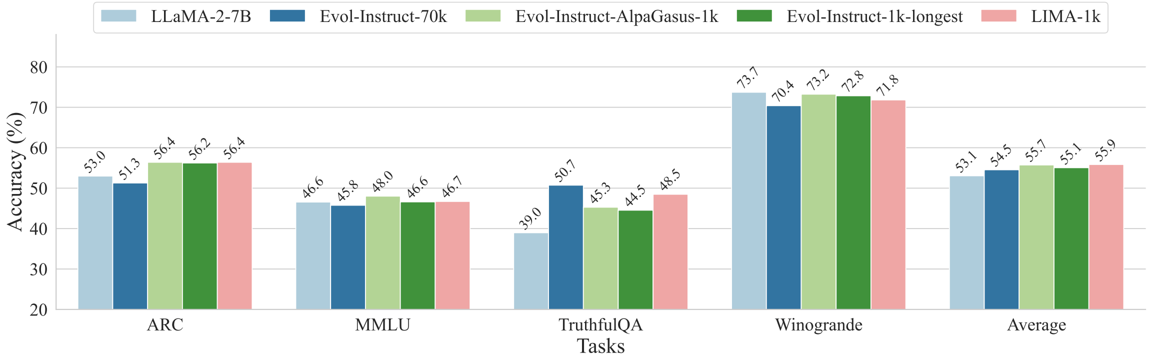
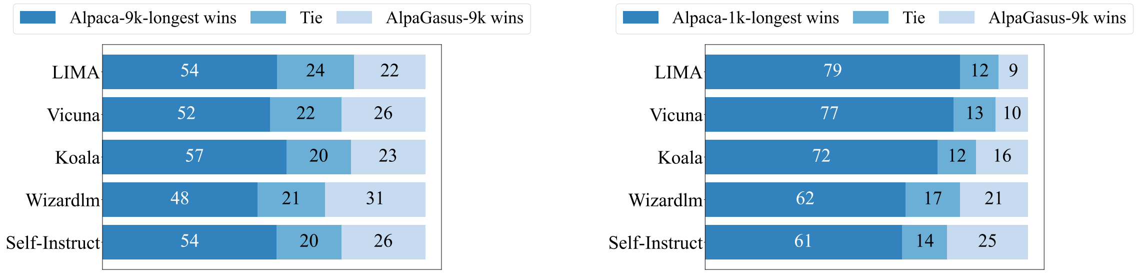


Figure 16: **Open LLM Leaderboard tasks with Llama-2-7B fine-tuned on Evol-Instruct-based datasets and LIMA.** The model fine-tuned on Evol-Instruct-1k-longest surpasses Llama-2-7B and Evol-Instruct-70k on average, showing that the performance gain on instruction-following capability does not compromise factuality.

ports our main claim: length is a strong criterion for constructing instruction fine-tuning dataset. Details of experimental setup can be seen in Table 2.



(a) Alpaca-9k-longest vs. AlpaGasus-9k

(b) Alpaca-1k-longest vs. AlpaGasus-9k

Figure 17: **Preference evaluation (%) using GPT4-as-a-judge** on LLaMA-2-7B models fine-tuned on AlpaGasus-9k, Alpaca-9k-longest, and Alpaca-1k-longest.

### D.2 Reflection-tuning

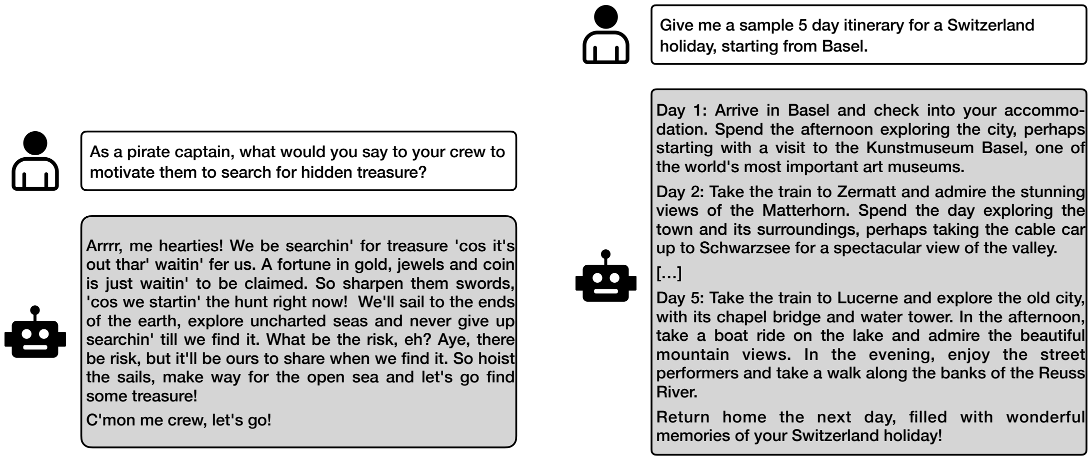
In this section, we show the advantage of proposed introspection technique by comparing it with reflection-tuning [20] on Llama-2-7B and Llama-2-13B models. We present experimental results on the Open LLM benchmark and AlpacaEval 2.0 in Table 3.

## Appendix E. Case study

This section consists of ten test instructions and corresponding responses of Llama-2-7B (Fig. 19 and Fig. 20), Mistral-7B (Fig. 21), and Llama-2-13B (Fig. 22 and Fig. 23) models fine-tuned on Alpaca-1k-longest, AlpaGasus-1k, Alpaca-52k, and LIMA-1k datasets. Details of training hyperparameters are shown in Table 2. We add detailed comments for qualitative

Table 3: Comparison between our introspection strategy and that of in the Reflection-Tuning [20]. * denotes that results are copied from the paper. Note that the performance of Llama-2-7B-Recycled-Alpaca-52k on the Winogrande task is evaluated using the open-source model checkpoint provided by Reflection-Tuning.

| Models | # SFT data | ARC | HellaSwag | MMLU | TruthfulQA | Winogrande | Average | AlpacaEval 2.0 | Avg. Length |
|---|---|---|---|---|---|---|---|---|---|
| Llama-2-7B | 0 | 52.99 | 78.64 | 46.56 | 38.97 | 73.72 | 58.18 | / | / |
| Llama-2-7B-Alpaca-52k | 52k | 53.92 | 78.82 | 47.05 | 40.32 | 71.82 | 58.39 | 2.74 | 586 |
| Llama-2-7B-Recycled-Alpaca-52k* | 52k | 53.92 | 77.68 | **47.55** | 45.55 | 71.82 | 59.30 | 5.93 | 1470 |
| Llama-2-7B-Refined-Alpaca-1k-L | 1k | **56.74** | **80.23** | 46.82 | **49.59** | **72.45** | **61.17** | **6.00** | 1732 |
| Llama-2-13B | 0 | 59.64 | 82.15 | 55.63 | 36.92 | 76.09 | 62.09 | / | / |
| Llama-2-13B-Alpaca-52k | 52k | 59.73 | 83.08 | 53.87 | 39.98 | 72.77 | 61.24 | 3.90 | 556 |
| Llama-2-13B-Recycled-Alpaca-52k* | 52k | 58.70 | 80.80 | 53.11 | **43.12** | ? | ? | ? | ? |
| Llama-2-13B-Refined-Alpaca-1k-L | 1k | **61.95** | **83.88** | **55.86** | 41.74 | **75.85** | **63.86** | **8.44** | 1646 |



As a pirate captain, what would you say to your crew to motivate them to search for hidden treasure?

Arrrr, me hearties! We be searchin' for treasure 'cos it's out thar' waitin' fer us. A fortune in gold, jewels and coin is just waitin' to be claimed. So sharpen them swords, 'cos we startin' the hunt right now! We'll sail to the ends of the earth, explore uncharted seas and never give up searchin' till we find it. What be the risk, eh? Aye, there be risk, but it'll be ours to share when we find it. So hoist the sails, make way for the open sea and let's go find some treasure!

C'mon me crew, let's go!

Give me a sample 5 day itinerary for a Switzerland holiday, starting from Basel.

Day 1: Arrive in Basel and check into your accommodation. Spend the afternoon exploring the city, perhaps starting with a visit to the Kunstmuseum Basel, one of the world's most important art museums.

Day 2: Take the train to Zermatt and admire the stunning views of the Matterhorn. Spend the day exploring the town and its surroundings, perhaps taking the cable car up to Schwarzsee for a spectacular view of the valley.

[...]

Day 5: Take the train to Lucerne and explore the old city, with its chapel bridge and water tower. In the afternoon, take a boat ride on the lake and admire the beautiful mountain views. In the evening, enjoy the street performers and take a walk along the banks of the Reuss River.

Return home the next day, filled with wonderful memories of your Switzerland holiday!

(a) give an inspiring speech as a pirate captain (b) generate a 5-day itinerary for a Switzerland holiday

Figure 18: **Example generations.** Case study to illustrate the instruction-following performance of Llama-2-7B model fine-tuned on Alpaca-1k-longest.

analysis on responses generated by Llama-2-7B in Section E.1. We omit detailed analysis for Mistral-7B and Llama-2-13B since we make similar observations as for Llama-2-7B.

## E.1 Detailed comments on Llama-2-7B examples

### Example #1: generate an itinerary in Switzerland.

- Alpaca-1k-longest provides a well-structured and detailed itinerary for a 5-day trip to Switzerland, starting from Basel. It includes a variety of activities, such as visiting museums, hiking, exploring towns, and enjoying local cuisine. It also suggests different modes of transportation, such as trains and cable cars, which are common in Switzerland. Its answer is relevant, accurate, and helpful. However it mentions a "famous Meierihne cheese", which does not exist at all. We believe this hallucination happens because of the knowledge capabilities of the base model.

- AlpaGasus-1k also provides a well-structured response and includes a variety of activities, it is slightly less detailed than Alpaca-1k-longest's response. For example, in Interlaken, AlpaGasus-1k suggests visiting popular hiking destinations but did not provide any information about what one might see or do there. However, AlpaGasus-1k does a good job of suggesting a variety of activities and destinations, making the itinerary interesting and diverse.

- Alpaca-52k's answer is less detailed and less helpful. The assistant suggested visiting the same cities on multiple days, which is not efficient or practical for a 5-day trip. The assistant also did not provide specific activities or places to visit in each city, which makes the answer less useful for someone planning a trip.

- LIMA-1k's answer is cut off and does not cover the full 5 days. It also repeats the same dining and nightlife options for each day, which is not very helpful or realistic.

**Example #2: give an inspiring speech as a pirate captain.**

- Alpaca-1k-longest provides excellent responses to this question. It uses appropriate pirate language and provides motivating speeches that would encourage a pirate crew to search for hidden treasure. The response is relevant, accurate, and detailed, providing a vivid picture of the adventure and potential rewards.

- AlpaGasus-1k's response is shorter and less detailed, but still motivational and in line with the question.

- Alpaca-52k's response is also motivational and uses appropriate language, but is less detailed and less vivid in its description of the journey and the treasure.

- LIMA-1k also provides excellent responses to this question. It uses appropriate pirate language and provides motivating speeches that would encourage a pirate crew to search for hidden treasure. The response is relevant, accurate, and detailed, providing a vivid picture of the adventure and potential rewards.

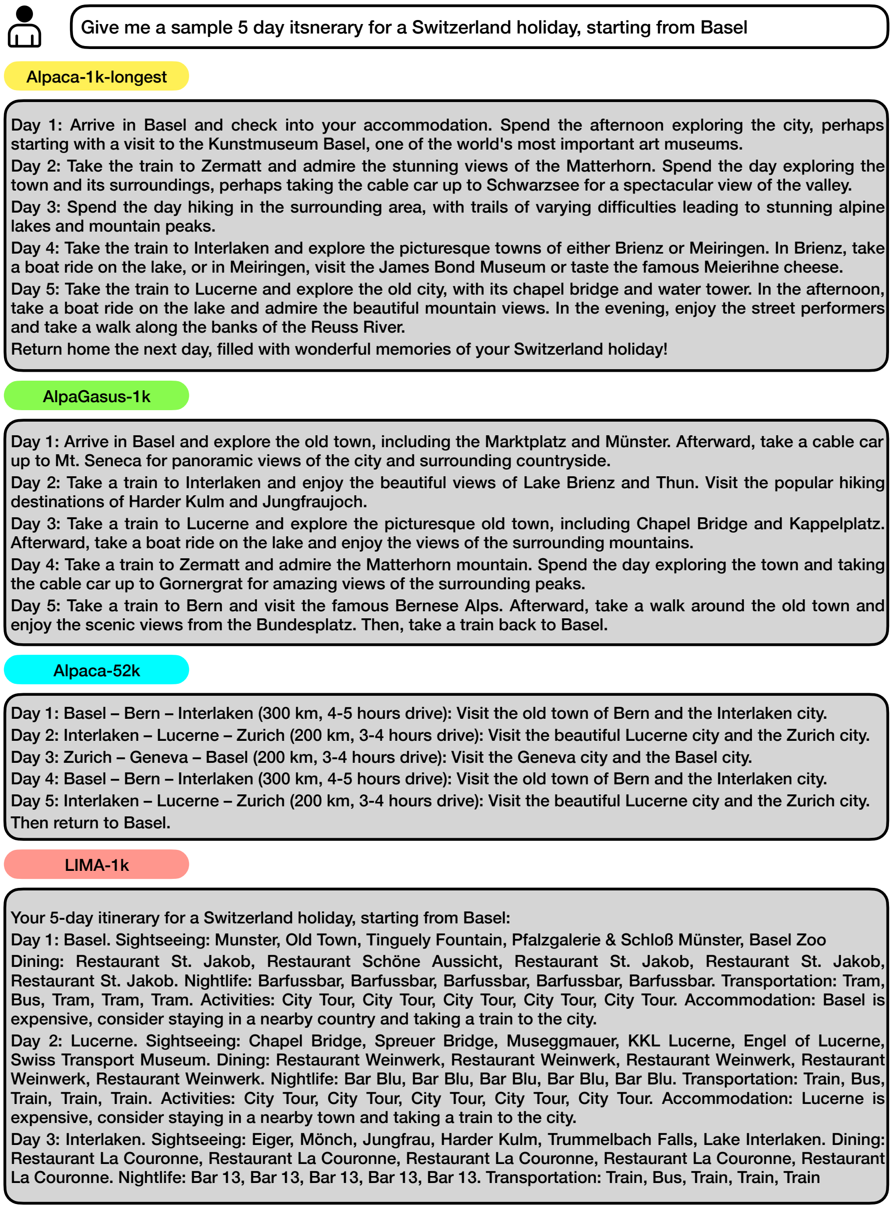
**Example #3: write a code snippet to validate an email address.**

- Alpaca-1k-longest provides a correct regular expression for validating an email address in Python and also explained what each part of the expression does. The explanation was clear and concise, making it easy to understand how the regular expression works.

- AlpaGasus-1k also provides a correct regular expression for validating an email address. However, there is no explanation or context provided, which might make it difficult for someone unfamiliar with regular expressions to understand.

- Alpaca-52k's answer is also correct and accurate, but lacks the detailed explanation

- LIMA-1k's regular expression is incorrect and does not match the standard email format. The explanation provided by LIMA-1k is also incorrect and confusing, as it does not correctly explain what each part of the regular expression does.



Give me a sample 5 day itsnerary for a Switzerland holiday, starting from Basel

**Alpaca-1k-longest**

Day 1: Arrive in Basel and check into your accommodation. Spend the afternoon exploring the city, perhaps starting with a visit to the Kunstmuseum Basel, one of the world's most important art museums.
Day 2: Take the train to Zermatt and admire the stunning views of the Matterhorn. Spend the day exploring the town and its surroundings, perhaps taking the cable car up to Schwarzsee for a spectacular view of the valley.
Day 3: Spend the day hiking in the surrounding area, with trails of varying difficulties leading to stunning alpine lakes and mountain peaks.
Day 4: Take the train to Interlaken and explore the picturesque towns of either Brienz or Meiringen. In Brienz, take a boat ride on the lake, or in Meiringen, visit the James Bond Museum or taste the famous Meierihne cheese.
Day 5: Take the train to Lucerne and explore the old city, with its chapel bridge and water tower. In the afternoon, take a boat ride on the lake and admire the beautiful mountain views. In the evening, enjoy the street performers and take a walk along the banks of the Reuss River.
Return home the next day, filled with wonderful memories of your Switzerland holiday!

**AlpaGasus-1k**

Day 1: Arrive in Basel and explore the old town, including the Marktplatz and Münster. Afterward, take a cable car up to Mt. Seneca for panoramic views of the city and surrounding countryside.
Day 2: Take a train to Interlaken and enjoy the beautiful views of Lake Brienz and Thun. Visit the popular hiking destinations of Harder Kulm and Jungfraujoch.
Day 3: Take a train to Lucerne and explore the picturesque old town, including Chapel Bridge and Kappelplatz. Afterward, take a boat ride on the lake and enjoy the views of the surrounding mountains.
Day 4: Take a train to Zermatt and admire the Matterhorn mountain. Spend the day exploring the town and taking the cable car up to Gornergrat for amazing views of the surrounding peaks.
Day 5: Take a train to Bern and visit the famous Bernese Alps. Afterward, take a walk around the old town and enjoy the scenic views from the Bundesplatz. Then, take a train back to Basel.

**Alpaca-52k**

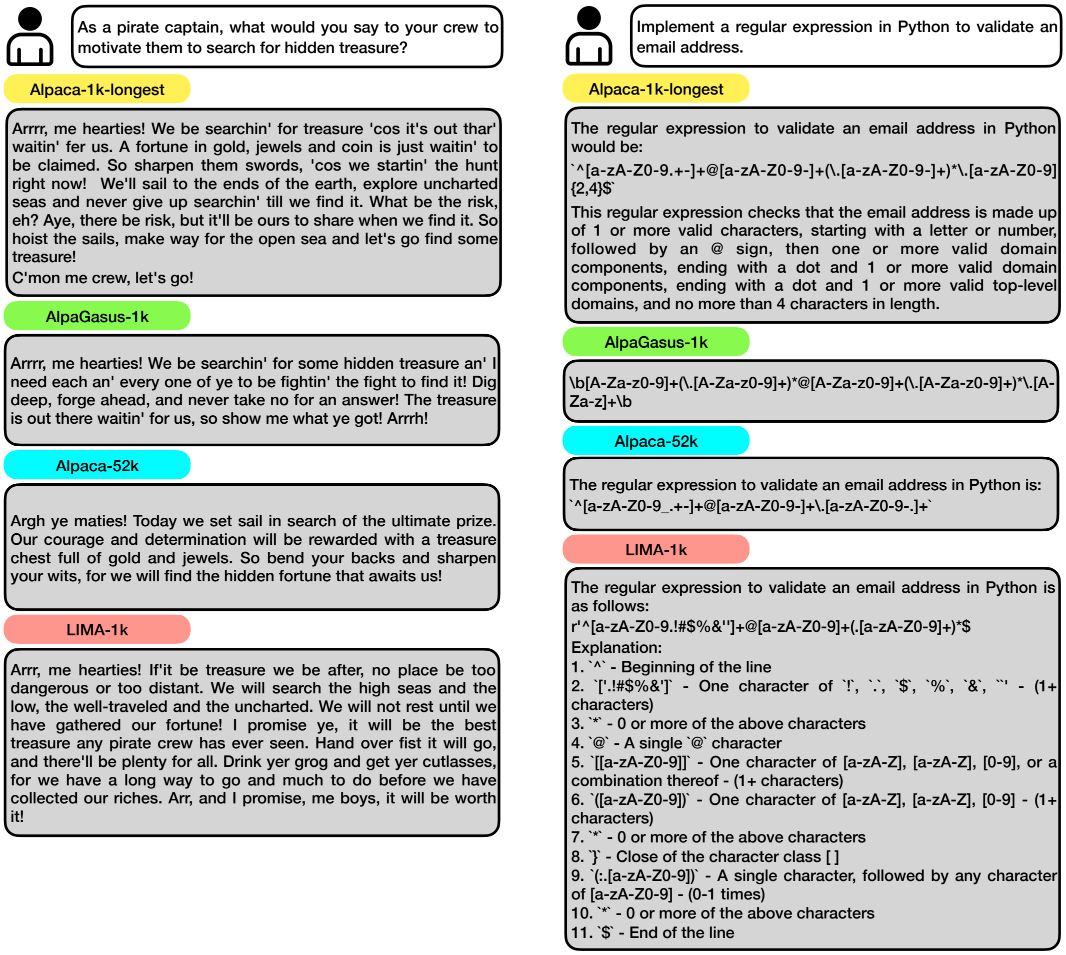
Day 1: Basel – Bern – Interlaken (300 km, 4-5 hours drive): Visit the old town of Bern and the Interlaken city.
Day 2: Interlaken – Lucerne – Zurich (200 km, 3-4 hours drive): Visit the beautiful Lucerne city and the Zurich city.
Day 3: Zurich – Geneva – Basel (200 km, 3-4 hours drive): Visit the Geneva city and the Basel city.
Day 4: Basel – Bern – Interlaken (300 km, 4-5 hours drive): Visit the old town of Bern and the Interlaken city.
Day 5: Interlaken – Lucerne – Zurich (200 km, 3-4 hours drive): Visit the beautiful Lucerne city and the Zurich city.
Then return to Basel.

**LIMA-1k**

Your 5-day itinerary for a Switzerland holiday, starting from Basel:
Day 1: Basel. Sightseeing: Munster, Old Town, Tinguely Fountain, Pfalzgalerie & Schloß Münster, Basel Zoo
Dining: Restaurant St. Jakob, Restaurant Schöne Aussicht, Restaurant St. Jakob, Restaurant St. Jakob, Restaurant St. Jakob. Nightlife: Barfussbar, Barfussbar, Barfussbar, Barfussbar, Barfussbar. Transportation: Tram, Bus, Tram, Tram, Tram. Activities: City Tour, City Tour, City Tour, City Tour, City Tour. Accommodation: Basel is expensive, consider staying in a nearby country and taking a train to the city.
Day 2: Lucerne. Sightseeing: Chapel Bridge, Spreuer Bridge, Museggmauer, KKL Lucerne, Engel of Lucerne, Swiss Transport Museum. Dining: Restaurant Weinwerk, Restaurant Weinwerk, Restaurant Weinwerk, Restaurant Weinwerk, Restaurant Weinwerk. Nightlife: Bar Blu, Bar Blu, Bar Blu, Bar Blu, Bar Blu. Transportation: Train, Bus, Train, Train, Train. Activities: City Tour, City Tour, City Tour, City Tour, City Tour. Accommodation: Lucerne is expensive, consider staying in a nearby town and taking a train to the city.
Day 3: Interlaken. Sightseeing: Eiger, Mönch, Jungfrau, Harder Kulm, Trummelbach Falls, Lake Interlaken. Dining: Restaurant La Couronne, Restaurant La Couronne, Restaurant La Couronne, Restaurant La Couronne, Restaurant La Couronne. Nightlife: Bar 13, Bar 13, Bar 13, Bar 13. Transportation: Train, Bus, Train, Train, Train

(a) Example #1: generate an itinerary in Switzerland

Figure 19: Case study on the LLaMA-2-7B model fine-tuned on different IFT datasets.



As a pirate captain, what would you say to your crew to motivate them to search for hidden treasure?

**Alpaca-1k-longest**

Arrrr, me hearties! We be searchin' for treasure 'cos it's out thar' waitin' fer us. A fortune in gold, jewels and coin is just waitin' to be claimed. So sharpen them swords, 'cos we startin' the hunt right now! We'll sail to the ends of the earth, explore uncharted seas and never give up searchin' till we find it. What be the risk, eh? Aye, there be risk, but it'll be ours to share when we find it. So hoist the sails, make way for the open sea and let's go find some treasure!

C'mon me crew, let's go!

**AlpaGasus-1k**

Arrrr, me hearties! We be searchin' for some hidden treasure an' I need each an' every one of ye to be fightin' the fight to find it! Dig deep, forge ahead, and never take no for an answer! The treasure is out there waitin' for us, so show me what ye got! Arrrh!

**Alpaca-52k**

Argh ye maties! Today we set sail in search of the ultimate prize. Our courage and determination will be rewarded with a treasure chest full of gold and jewels. So bend your backs and sharpen your wits, for we will find the hidden fortune that awaits us!

**LIMA-1k**

Arrr, me hearties! If'it be treasure we be after, no place be too dangerous or too distant. We will search the high seas and the low, the well-traveled and the uncharted. We will not rest until we have gathered our fortune! I promise ye, it will be the best treasure any pirate crew has ever seen. Hand over fist it will go, and there'll be plenty for all. Drink yer grog and get yer cutlasses, for we have a long way to go and much to do before we have collected our riches. Arr, and I promise, me boys, it will be worth it!

Implement a regular expression in Python to validate an email address.

**Alpaca-1k-longest**

The regular expression to validate an email address in Python would be:

`^[a-zA-Z0-9.+-]+@[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*\.[a-zA-Z0-9]{2,4}$`

This regular expression checks that the email address is made up of 1 or more valid characters, starting with a letter or number, followed by an @ sign, then one or more valid domain components, ending with a dot and 1 or more valid domain components, ending with a dot and 1 or more valid top-level domains, and no more than 4 characters in length.

**AlpaGasus-1k**

\b[A-Za-z0-9]+(\.[A-Za-z0-9]+)*@[A-Za-z0-9]+(\.[A-Za-z0-9]+)*\.[A-Za-z]+\b

**Alpaca-52k**

The regular expression to validate an email address in Python is:
`^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+`

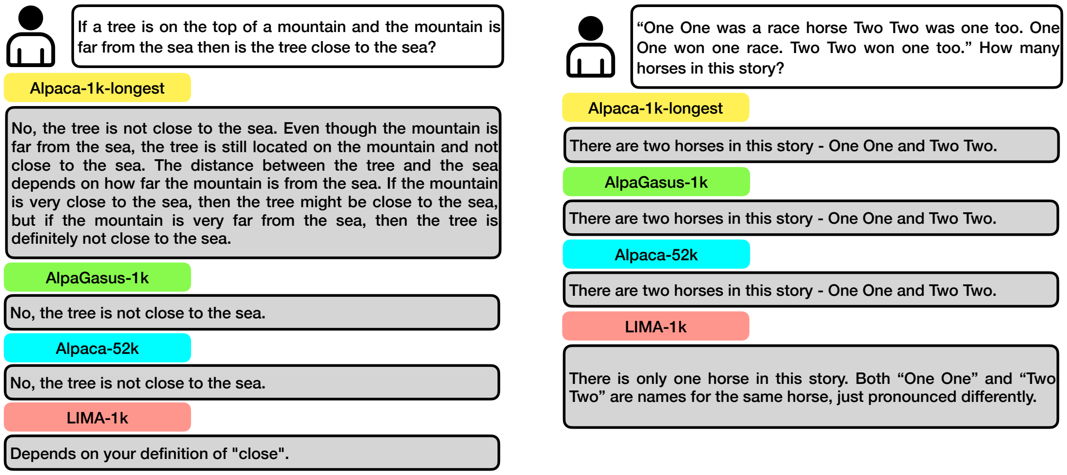
**LIMA-1k**

The regular expression to validate an email address in Python is as follows:
r'^[a-zA-Z0-9.!#$%&'']+@[a-zA-Z0-9]+(.[a-zA-Z0-9]+)*$
Explanation:
1. `^` - Beginning of the line
2. `['.!#$%&']` - One character of `!`, `.`, `$`, `%`, `&`, ``' - (1+ characters)
3. `*` - 0 or more of the above characters
4. `@` - A single `@` character
5. `[[a-zA-Z0-9]]` - One character of [a-zA-Z], [a-zA-Z], [0-9], or a combination thereof - (1+ characters)
6. `([a-zA-Z0-9])` - One character of [a-zA-Z], [a-zA-Z], [0-9] - (1+ characters)
7. `*` - 0 or more of the above characters
8. `}` - Close of the character class [ ]
9. `(:.[a-zA-Z0-9])` - A single character, followed by any character of [a-zA-Z0-9] - (0-1 times)
10. `*` - 0 or more of the above characters
11. `$` - End of the line

(a) Example #2: give an inspiring speech as a pirate captain

(b) Example #3: write a code snippet to validate an email address

Figure 20: Case study on the LLaMA-2-7B model fine-tuned on different IFT datasets.



If a tree is on the top of a mountain and the mountain is far from the sea then is the tree close to the sea?

**Alpaca-1k-longest**

No, the tree is not close to the sea. Even though the mountain is far from the sea, the tree is still located on the mountain and not close to the sea. The distance between the tree and the sea depends on how far the mountain is from the sea. If the mountain is very close to the sea, then the tree might be close to the sea, but if the mountain is very far from the sea, then the tree is definitely not close to the sea.

**AlpaGasus-1k**

No, the tree is not close to the sea.

**Alpaca-52k**

No, the tree is not close to the sea.

**LIMA-1k**

Depends on your definition of "close".

"One One was a race horse Two Two was one too. One One won one race. Two Two won one too." How many horses in this story?

**Alpaca-1k-longest**

There are two horses in this story - One One and Two Two.

**AlpaGasus-1k**

There are two horses in this story - One One and Two Two.

**Alpaca-52k**

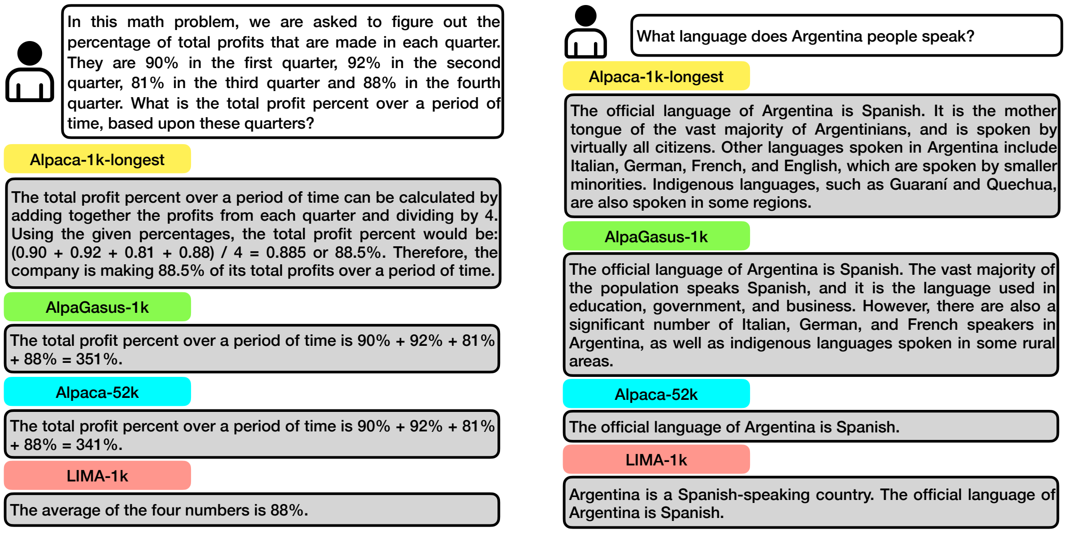
There are two horses in this story - One One and Two Two.

**LIMA-1k**

There is only one horse in this story. Both "One One" and "Two Two" are names for the same horse, just pronounced differently.

(a) Example #4: reasoning

(b) Example #5: reasoning



In this math problem, we are asked to figure out the percentage of total profits that are made in each quarter. They are 90% in the first quarter, 92% in the second quarter, 81% in the third quarter and 88% in the fourth quarter. What is the total profit percent over a period of time, based upon these quarters?

**Alpaca-1k-longest**

The total profit percent over a period of time can be calculated by adding together the profits from each quarter and dividing by 4. Using the given percentages, the total profit percent would be: (0.90 + 0.92 + 0.81 + 0.88) / 4 = 0.885 or 88.5%. Therefore, the company is making 88.5% of its total profits over a period of time.

**AlpaGasus-1k**

The total profit percent over a period of time is 90% + 92% + 81% + 88% = 351%.

**Alpaca-52k**

The total profit percent over a period of time is 90% + 92% + 81% + 88% = 341%.

**LIMA-1k**

The average of the four numbers is 88%.

What language does Argentina people speak?

**Alpaca-1k-longest**

The official language of Argentina is Spanish. It is the mother tongue of the vast majority of Argentinians, and is spoken by virtually all citizens. Other languages spoken in Argentina include Italian, German, French, and English, which are spoken by smaller minorities. Indigenous languages, such as Guaraní and Quechua, are also spoken in some regions.

**AlpaGasus-1k**

The official language of Argentina is Spanish. The vast majority of the population speaks Spanish, and it is the language used in education, government, and business. However, there are also a significant number of Italian, German, and French speakers in Argentina, as well as indigenous languages spoken in some rural areas.

**Alpaca-52k**

The official language of Argentina is Spanish.

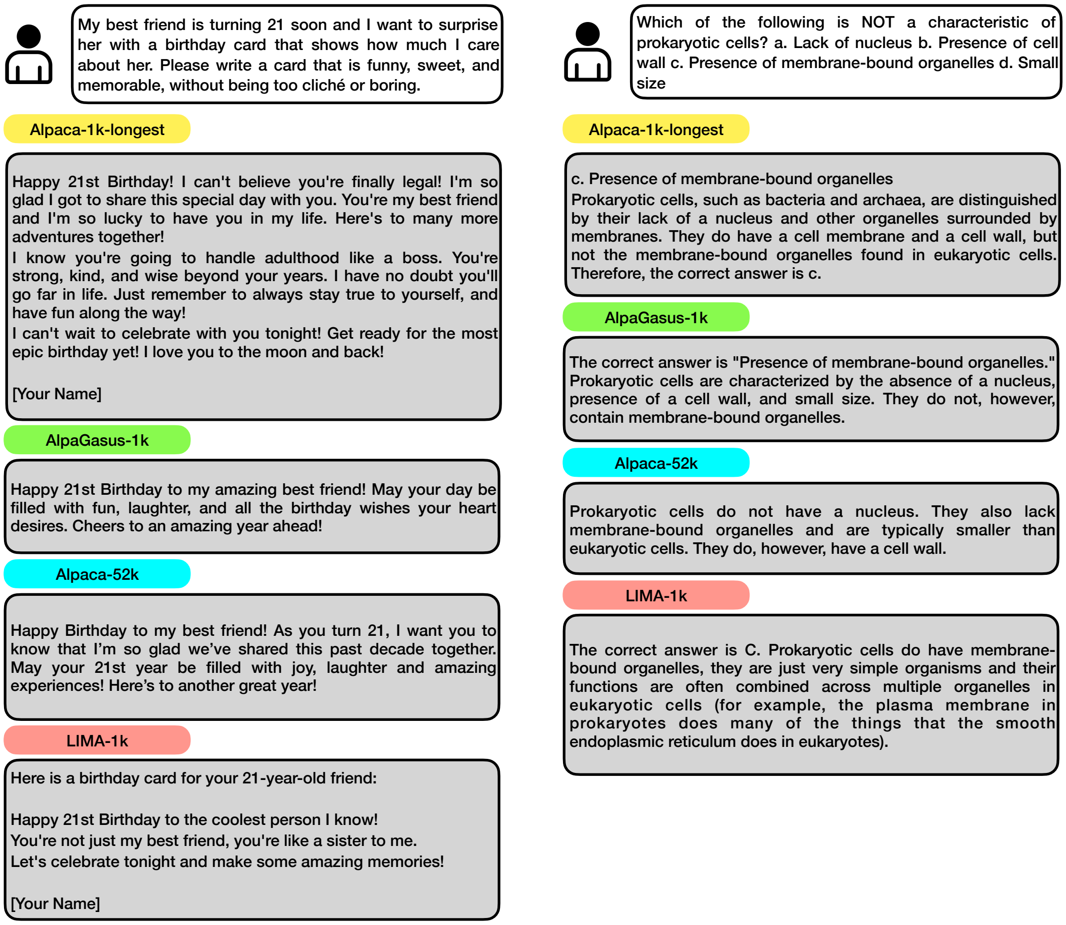
**LIMA-1k**

Argentina is a Spanish-speaking country. The official language of Argentina is Spanish.

(c) Example #6: calculate average profits

(d) Example #7: what language does Argentina people speak

Figure 21: Case study on the Mistral-7B model fine-tuned on different IFT datasets.



My best friend is turning 21 soon and I want to surprise her with a birthday card that shows how much I care about her. Please write a card that is funny, sweet, and memorable, without being too cliché or boring.

**Alpaca-1k-longest**

Happy 21st Birthday! I can't believe you're finally legal! I'm so glad I got to share this special day with you. You're my best friend and I'm so lucky to have you in my life. Here's to many more adventures together!

I know you're going to handle adulthood like a boss. You're strong, kind, and wise beyond your years. I have no doubt you'll go far in life. Just remember to always stay true to yourself, and have fun along the way!

I can't wait to celebrate with you tonight! Get ready for the most epic birthday yet! I love you to the moon and back!

[Your Name]

**AlpaGasus-1k**

Happy 21st Birthday to my amazing best friend! May your day be filled with fun, laughter, and all the birthday wishes your heart desires. Cheers to an amazing year ahead!

**Alpaca-52k**

Happy Birthday to my best friend! As you turn 21, I want you to know that I’m so glad we’ve shared this past decade together. May your 21st year be filled with joy, laughter and amazing experiences! Here’s to another great year!

**LIMA-1k**

Here is a birthday card for your 21-year-old friend:

Happy 21st Birthday to the coolest person I know!
You're not just my best friend, you're like a sister to me.
Let's celebrate tonight and make some amazing memories!

[Your Name]

Which of the following is NOT a characteristic of prokaryotic cells? a. Lack of nucleus b. Presence of cell wall c. Presence of membrane-bound organelles d. Small size

**Alpaca-1k-longest**

c. Presence of membrane-bound organelles
Prokaryotic cells, such as bacteria and archaea, are distinguished by their lack of a nucleus and other organelles surrounded by membranes. They do have a cell membrane and a cell wall, but not the membrane-bound organelles found in eukaryotic cells. Therefore, the correct answer is c.

**AlpaGasus-1k**

The correct answer is "Presence of membrane-bound organelles." Prokaryotic cells are characterized by the absence of a nucleus, presence of a cell wall, and small size. They do not, however, contain membrane-bound organelles.

**Alpaca-52k**

Prokaryotic cells do not have a nucleus. They also lack membrane-bound organelles and are typically smaller than eukaryotic cells. They do, however, have a cell wall.

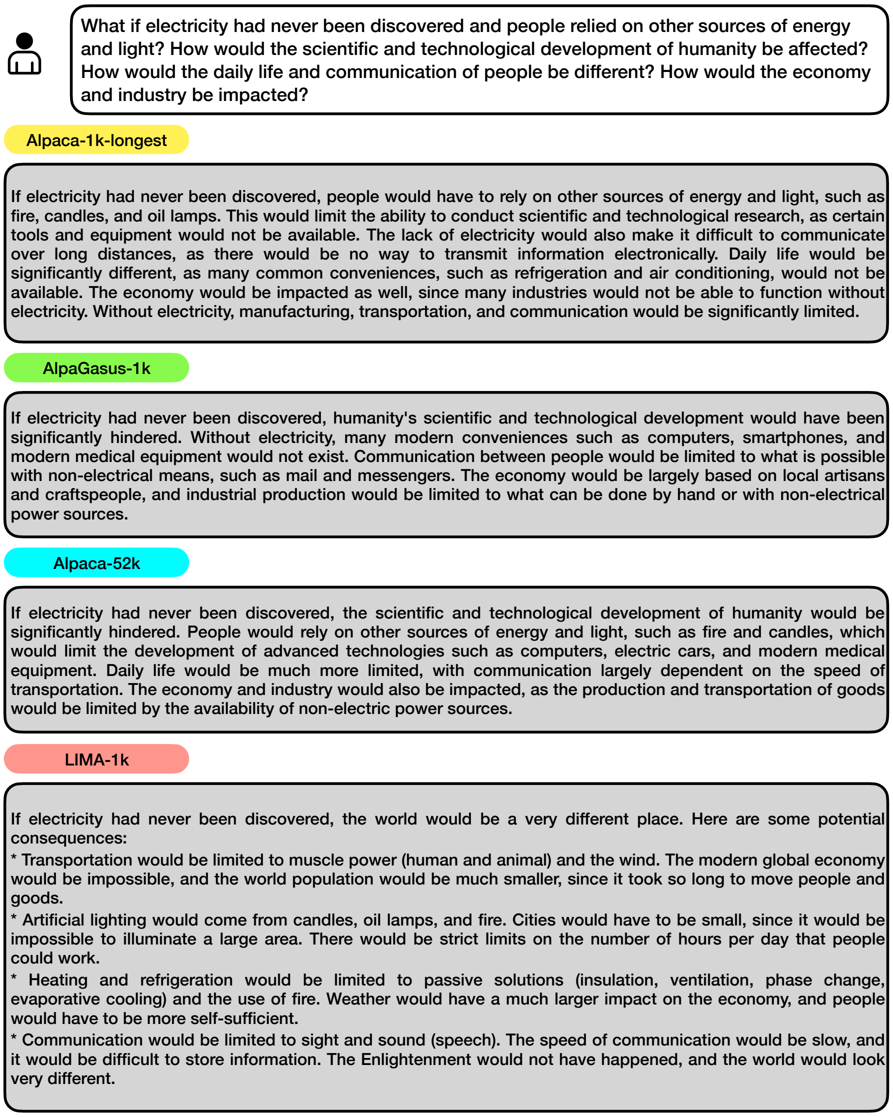
**LIMA-1k**

The correct answer is C. Prokaryotic cells do have membrane-bound organelles, they are just very simple organisms and their functions are often combined across multiple organelles in eukaryotic cells (for example, the plasma membrane in prokaryotes does many of the things that the smooth endoplasmic reticulum does in eukaryotes).

(a) Example #8: generate a birthday wish message

(b) Example #9: answer a biology question

Figure 22: Case study on the LLaMA-2-13B model fine-tuned on different IFT datasets.



What if electricity had never been discovered and people relied on other sources of energy and light? How would the scientific and technological development of humanity be affected? How would the daily life and communication of people be different? How would the economy and industry be impacted?

**Alpaca-1k-longest**

If electricity had never been discovered, people would have to rely on other sources of energy and light, such as fire, candles, and oil lamps. This would limit the ability to conduct scientific and technological research, as certain tools and equipment would not be available. The lack of electricity would also make it difficult to communicate over long distances, as there would be no way to transmit information electronically. Daily life would be significantly different, as many common conveniences, such as refrigeration and air conditioning, would not be available. The economy would be impacted as well, since many industries would not be able to function without electricity. Without electricity, manufacturing, transportation, and communication would be significantly limited.

**AlpaGasus-1k**

If electricity had never been discovered, humanity's scientific and technological development would have been significantly hindered. Without electricity, many modern conveniences such as computers, smartphones, and modern medical equipment would not exist. Communication between people would be limited to what is possible with non-electrical means, such as mail and messengers. The economy would be largely based on local artisans and craftspeople, and industrial production would be limited to what can be done by hand or with non-electrical power sources.

**Alpaca-52k**

If electricity had never been discovered, the scientific and technological development of humanity would be significantly hindered. People would rely on other sources of energy and light, such as fire and candles, which would limit the development of advanced technologies such as computers, electric cars, and modern medical equipment. Daily life would be much more limited, with communication largely dependent on the speed of transportation. The economy and industry would also be impacted, as the production and transportation of goods would be limited by the availability of non-electric power sources.

**LIMA-1k**

If electricity had never been discovered, the world would be a very different place. Here are some potential consequences:

* Transportation would be limited to muscle power (human and animal) and the wind. The modern global economy would be impossible, and the world population would be much smaller, since it took so long to move people and goods.

* Artificial lighting would come from candles, oil lamps, and fire. Cities would have to be small, since it would be impossible to illuminate a large area. There would be strict limits on the number of hours per day that people could work.

* Heating and refrigeration would be limited to passive solutions (insulation, ventilation, phase change, evaporative cooling) and the use of fire. Weather would have a much larger impact on the economy, and people would have to be more self-sufficient.

* Communication would be limited to sight and sound (speech). The speed of communication would be slow, and it would be difficult to store information. The Enlightenment would not have happened, and the world would look very different.

(a) Example #10: human society development

Figure 23: Case study on the LLaMA-2-13B model fine-tuned on different IFT datasets.

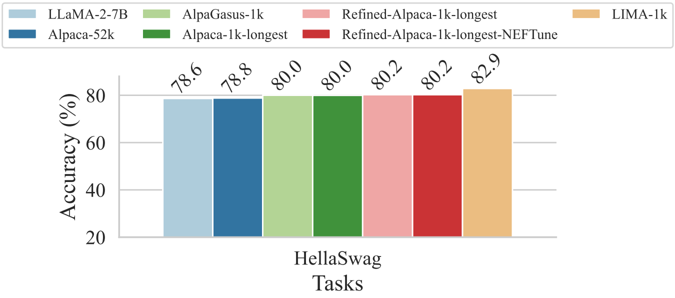


Figure 24: The performance of a diverse array of instruction fine-tuned models on the HellaSwag task. The very high accuracy of the models fine-tuned on LIMA-1k might be explained by data contamination (see discussion in App. F).

## Appendix F. Data contamintation on LIMA-1k

With over 240k how-to articles covering a wide range of topics, wikiHow is an online publication in the style of a wiki, where articles are frequently regarded as high-quality content. LIMA [42] contains 200 wikiHow examples. The article's title serves as a prompt (e.g., "How to Cook an Omelet?") and the body text as an answer. HellaSwag [39] from Open LLM leaderboard also includes wikiHow articles to enhance the content diversity. By cross validating the evaluation set of the HellaSwag task and the training set of LIMA, we find that the style and format of 200 wikiHow examples in LIMA are highly similar to that of in HellaSwag evaluation set. Also, surprisingly, we notice that multiple examples (e.g., "How to get a free room upgrade in las vegas?", "How to teach a child to use scissors?", "How to handle poking wires on braces?", etc.) appear in both datasets, which is a strong signal of data contamination. The performance of LIMA-1k model on the HellaSwag task is also suspiciously higher than the other baselines as shown in Fig. 24.