

# LMFlow: An Extensible Toolkit for Finetuning and Inference of Large Foundation Models

Shizhe Diao<sup>♡\*</sup>, Rui Pan<sup>♡\*</sup>, Hanze Dong<sup>♡\*</sup>,  
KaShun Shum<sup>♡</sup>, Jipeng Zhang<sup>♡</sup>, Wei Xiong<sup>♣</sup>, Tong Zhang<sup>♣</sup>

<sup>♡</sup>The Hong Kong University of Science and Technology

<sup>♣</sup>University of Illinois Urbana-Champaign

{sdiaaaa, rpan, hdongaj}@ust.hk

tozhang@illinois.edu

## Abstract

Foundation models have demonstrated a great ability to achieve general human-level intelligence far beyond traditional approaches. As the technique keeps attracting attention from the AI community, an increasing number of foundation models are becoming publicly accessible. However, a significant shortcoming of most of these models lies in their performance in specialized-domain and task-specific applications, necessitating domain- and task-aware fine-tuning to develop effective scientific language models. As the number of available foundation models and specialized tasks keeps growing, the job of training scientific language models becomes highly nontrivial. In this paper, we initiate steps to tackle this issue. We introduce an extensible and lightweight toolkit, LMFlow, which aims to simplify the domain- and task-aware finetuning of general foundation models. LMFlow offers a complete finetuning workflow for a foundation model to support specialized training with limited computing resources. Furthermore, it supports continuous pretraining, instruction tuning, parameter-efficient finetuning, alignment tuning, inference acceleration, long context generalization, model customization, and even multimodal finetuning, along with carefully designed and extensible APIs. This toolkit has been thoroughly tested and is available at <https://github.com/OptimalScale/LMFlow>.<sup>1</sup>

## 1 Introduction

Foundation models (FMs), and in particular large language models (LLMs), have demonstrated general abilities to perform different tasks beyond what

was possible previously. While a number of pre-trained large models, including GPT-J (Wang and Komatsuzaki, 2021), Bloom (Scao et al., 2022), LLaMA (Touvron et al., 2023a,b), etc., are publicly available and have already been incorporated into the Hugging Face model repository (Huggingface, 2022), there is no publicly available toolkit that can be easily used to perform finetuning and inference for these different models. For specialized domains or tasks, it is necessary to further fine-tune such LLMs to achieve improved performance on such domains or tasks. The purpose of this package is to offer a simple-to-use and lightweight toolkit so that developers and researchers can perform efficient finetuning and inference of scientific language models with limited resources. The typical processes to train a scientific language model are shown in Figure 1, which include:

- Continuous pretraining on datasets in special domains and tasks so that a foundation model can acquire domain- and task-specific knowledge. It normally contains domain or task adaptation.
- Instruction tuning to teach a foundation model the capability to follow these specialized natural language instructions and perform tasks required by such instructions.
- Reinforcement learning from human feedback (RLHF) to align a foundation model to human preference (for example, helpfulness, harmlessness, and honesty).

LMFlow enhances and streamlines the aforementioned fine-tuning procedures, enabling the efficient and effective training of a scientific language model. We focus on improving training speed. For example, it only takes one Nvidia 3090 GPU and five hours to train a medical LLaMA comparable to ChatGPT, based on a 7-billion-parameter LLaMA model. In addition to speed, we also aspire to achieve higher model performance. We used this framework to train medical LLaMA, a series of models with 7-billion, 13-billion, 33-billion, and

\*Equal Contribution.

<sup>1</sup>Video demonstrations trained and deployed by LMFlow:

- Emotional Companion bot: <https://www.youtube.com/watch?v=BDSME4f2AjU>
- Multimodal reasoning-based object detection bot: <https://www.youtube.com/watch?v=YXNyh6bGqyI>

Packages	Cont. PT	FT	RLHF	Deploy.	Adapt.	Acc.	LC	VE	MM
Transformers (Wolf et al., 2020)	✓	✓						✓	✓
Accelerate (Gugger et al., 2022)	✓	✓				✓			
DeepSpeed (Rasley et al., 2020)	✓	✓				✓			
Trl (von Werra et al., 2020)			✓						
LMFlow (ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison with competing packages. **Cont. PT**: continuous pretraining. **FT**: finetuning. **RLHF**: reinforcement learning from human feedback. **Deploy.**: deployment. **Adapt.**: domain/task adaptation. **Acc.**: acceleration techniques for finetuning and inference. **LC**: long context generalization. **VE**: vocabulary extension. **MM**: multimodal training.

65-billion parameters, on a single machine and have released the model weights for academic research. Using LMFlow, anyone can train their own scientific or personalized language models. Each person can choose the appropriate foundation model according to their available resources, for tasks such as question answering, companionship, and expert consultations in various domains. The larger the model and data size, the longer the training time and the better the results. Compared with existing packages, LMFlow encompasses a multitude of features that are absent in others, such as the support for long context generalization, as shown in Table 1. Most importantly, LMFlow stands out as a comprehensive, full-cycle foundation model adaptation toolkit. While other packages excel in specific areas like finetuning, they lack functionalities like RLHF and others. To our knowledge, LMFlow is the first to offer a complete pipeline that integrates all these processes. This holistic toolkit allows for more robust and adaptable language model training and inference, setting a new standard in the field of natural language processing.

## 2 Related Work

In recent years, the finetuning of large language models (LLMs) has gained significant attention, especially for scientific domain applications. The necessity of adapting these general-purpose models to specific domains or tasks has led to the development of various scientific language models. Lehman et al. (2023) conducted an extensive empirical analysis on the performance of various language models in clinical tasks and found that specialized clinical models, even smaller in size, significantly outperform larger general-domain models when finetuned on domain-specific data. This emphasizes the importance of domain specialization in achieving higher accuracy in safety-critical fields

like healthcare. Therefore, a series of scientific large models have emerged, including but not limited to: language models for Science (Beltagy et al., 2019; Luu et al., 2021; Taylor et al., 2022), Mathematics (Yue et al., 2023; Yu et al., 2023; Gao et al., 2023), Physics (Nguyen et al., 2023; Zheng et al., 2023b; Perkowski et al., 2024), Chemistry and Materials Science (Cao et al., 2023; Shetty et al., 2023; Rubungo et al., 2023), Biology and Medicine (Lee et al., 2020; Zhang et al., 2023; Singhal et al., 2023; Wu et al., 2023; Han et al., 2023; Wang et al., 2023; Yang et al., 2024), and Information Retrieval (Lassance et al., 2023) We recommend readers to refer to a paper list of scientific language models<sup>2</sup>, which includes a more comprehensive range of works related to scientific language models. Among these works, LMFlow has successfully helped in training AstroLLaMA-Chat (Perkowski et al., 2024) and MarineGPT (Zheng et al., 2023b). The Medical LLaMA trained in the medical domain within this paper also demonstrates the effectiveness of LMFlow. In summary, our proposed LMFlow offers a comprehensive toolkit for efficient and effective finetuning of foundation models across various specialized domains.

## 3 Toolkit Overview

### 3.1 System Design

An illustration of the LMFlow system design is shown in Figure 1. There are four stages for improving the performance of a publicly available foundation model. The first stage is domain adaptation, which involves modifying the model to better handle a specific domain by training the model on that domain. The second stage is task adaptation, which involves adapting the model to perform a specific task, such as summarization, question-

<sup>2</sup><https://github.com/yuzhimanhua/Awesome-Scientific-Language-Models>

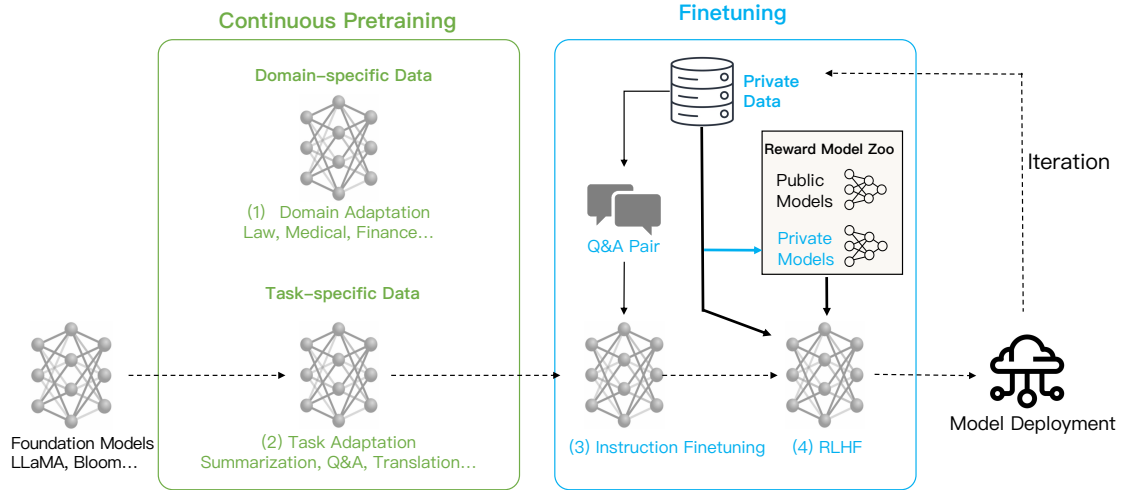


Figure 1: The system design of LMFlow. Starting from a publicly available foundation model, there are four possible stages including (1) domain adaptation, (2) task adaptation, (3) instruction finetuning, and (4) reinforcement learning with human feedback.

answering, and translation. The third stage is instruction finetuning, which involves adjusting the model’s parameters based on instructional question-answer pairs. The final stage is reinforcement learning with human feedback, which involves using human feedback to further align the model to human preference. LMFlow provides a complete finetuning workflow for these four stages, supporting large language models’ specialized training with limited computing resources. Especially, LMFlow supports the following key features:

- Finetuning Acceleration and Memory Optimization: LoRA (Hu et al.), FlashAttention (Dao et al., 2022; Dao, 2023), Gradient Checkpointing, and DeepSpeed Zero3.
- Inference Acceleration: Speculative Decoding (Leviathan et al., 2023), LLaMA Inference on CPU, and FlashAttention (Dao et al., 2022; Dao, 2023).
- Alignment Tuning: An implementation of our proposed novel alignment algorithm RAFT (Dong et al., 2023) (Reward rAnked FineTuning) to simply RLHF pipeline for generative models.
- Long Context Generalization: Position Interpolation for LLaMA (Chen et al., 2023).
- Model Customization: Vocabulary Extension.
- Multimodal: Finetuning Multimodal Chatbot for reasoning-based object detection (Pi et al., 2023).

### 3.2 Installation

LMFlow has been fully tested on Linux OS (Ubuntu 20.04) and can be installed by executing the following commands.

```
$ git clone https://github.com/OptimalScale/LMFlow.git
$ cd LMFlow
$ conda create -n lmflow python=3.9 -y
$ conda activate lmflow
$ pip install -e .
```

### 3.3 Data Format

LMFlow accepts several .json files as input. Users can provide a list of .json files under a specified dataset directory. For example,

```
1 |- path_to_dataset
2   |- data_1.json
3   |- data_2.json
4   |- another_data.json
5   |- ...
```

Each json file shall have the following format (three instances with four keys for example),

```
1 {
2   "type": "TYPE",
3   "instances": [
4     {
5       "KEY_1": "VALUE_1.1",
6       "KEY_2": "VALUE_1.2",
7       "KEY_3": "VALUE_1.3",
8       "KEY_4": "VALUE_1.4",
9     },
10    {
11     "KEY_1": "VALUE_2.1",
12     "KEY_2": "VALUE_2.2",
13     "KEY_3": "VALUE_2.3",
```

```

14     "KEY_4": "VALUE_2.4",
15   },
16   {
17     "KEY_1": "VALUE_3.1",
18     "KEY_2": "VALUE_3.2",
19     "KEY_3": "VALUE_3.3",
20     "KEY_4": "VALUE_3.4",
21   },
22 ]
23 }

```

where the TYPE indicates the dataset type and defines the set of keys { KEY\_1, KEY\_2, ... } and their corresponding interpretations. Two supported .json formats are detailed as follows.

**TextOnly** This is the most common dataset type, which only contains raw texts in each sample. This type of dataset can be used as the training set for text decoder models, or the input of decoder models / encoder-decoder models. Its format is as follows (three instances, for example),

```

1 {
2   "type": "text_only",
3   "instances": [
4     { "text": "SAMPLE_TEXT_1" },
5     { "text": "SAMPLE_TEXT_2" },
6     { "text": "SAMPLE_TEXT_3" },
7   ]
8 }

```

**Text2Text** This is the dataset type mostly used for inferencing, which contains a pair of texts in each sample. This type of dataset can be used as the training set for text encoder-decoder models, or question-answer pair for evaluating model inferences. Its format is as follows (three instances for example),

```

1 {
2   "type": "text2text",
3   "instances": [
4     {
5       "input": "SAMPLE_INPUT_1",
6       "output": "SAMPLE_OUTPUT_1",
7     },
8     {
9       "input": "SAMPLE_INPUT_2",
10      "output": "SAMPLE_OUTPUT_2",
11     },
12    {
13      "input": "SAMPLE_INPUT_3",
14      "output": "SAMPLE_OUTPUT_3",
15    },
16  ]
17 }

```

### 3.4 Continuous Pretraining

The endeavor to bridge the divide between pretraining domains and downstream domains has led to

the adoption of a prevalent approach, known as continuous pretraining (Beltagy et al., 2019; Alsentzer et al., 2019; Huang et al., 2019; Lee et al., 2020), which involves the ongoing pretraining on an extensive collection of unlabeled data that is specific to a given domain. LMFlow supports continuous pretraining natively, which is an effective way to adapt LLMs to a specific domain. Users just need to collect a set of unlabeled data and prepare them to TextOnly data format. The following process will be handled by autoregressive training.

### 3.5 Instruction Tuning

Instruction tuning (Sanh et al.; Wei et al.; Chung et al., 2022; Muennighoff et al., 2022; Wang et al., 2022), also called supervised finetuning, is an approach to enhance the performance of language models by training them to follow natural language instructions. This involves training the model on a small set of task-specific data, most of which are in prompt-answer format, including positive or negative examples, prompts, constraints, and other elements commonly present in human language. Instruction tuning enables LLMs to provide more accurate and relevant responses to user queries, making them more effective conversational agents.

### 3.6 RLHF as Finetuning

There is a growing need to explore alternative pretraining objectives that can guide LLMs to generate text that aligns with human preferences. By doing so, we can ensure that LLMs produce text that is more helpful, honest, and harmless for humans, which are called ‘HHH’ rules (Askell et al., 2021). Ouyang et al. (2022) divides the alignment process into three steps, including SFT, reward modeling, and RLHF (reward optimization). We have integrated all of these steps into our LMFlow framework. For reward optimization, PPO has been shown to be effective in various studies (Schulman et al., 2017; Engstrom et al., 2020). However, it relies on a trial-and-error approach through interaction with the environment, making it less stable and efficient than supervised learning (Choshen et al., 2019). To address this, we propose and implement a new alignment method for generative models called RAFT (Dong et al., 2023). RAFT utilizes a reward model to rank the output of the generative model, allowing us to continue training using supervised finetuning (SFT)-like techniques with the selected samples. This approach encourages the generative model to prioritize samples with higher

MODEL	anatomy	clinical knowledge	college biology	college medicine	medical genetics	professional medicine	Average
LLaMA 33B	39.2	40.3	44.4	32.9	36.0	43.0	39.3
Galactica 30B	32.5	26.0	30.5	25.4	39.0	23.1	29.4
Galactica 120B	<b>58.5</b>	59.2	68.7	57.2	68.0	59.6	61.9
OPT 175B	28.9	21.9	30.6	-	35.0	27.9	-
BLOOM 176B	37.0	29.8	28.5	-	36.0	25.4	-
Gopher 280B	56.3	67.2	70.8	60.1	69.0	64.0	64.6
GPT3.5	56.3	<b>69.8</b>	<b>72.2</b>	<b>61.3</b>	<b>70.0</b>	<b>70.2</b>	<b>66.6</b>
Task-tuned LLaMA 33B (LoRA)	51.8	65.2	70.1	58.3	65.6	66.5	62.9

Table 2: The performance on Massive Multitask Language Understanding (MMLU) benchmark. **Bold** represents the best among each dataset.

MODEL	PubMedQA (ID)	MedQA-USMLE (OOD)	MedMCQA (ID)	Average
Human (pass)	-	60.0	50.0	-
Human (expert)	78.0	87.0	90.0	85.0
InstructGPT-175B	73.2	46.0	44.0	54.4
ChatGPT	63.9	<b>57.0</b>	44.7	55.2
LLaMA-7B	5.2	27.1	24.3	18.9
LLaMA-33B	1.8	43.4	30.3	25.2
Task-tuned LLaMA-7B (full)	<b>75.1</b>	44.5	49.9	56.5
Task-tuned LLaMA-33B (LoRA)	74.0	51.3	<b>50.2</b>	<b>58.5</b>

Table 3: The overall performance of task-tuned LLaMA models and the comparison with human and existing models on three medical datasets. PubMedQA and MedMCQA are evaluated on in-domain tests and MedQA-USMLE is evaluated on the out-of-domain test. **Bold** represents the best among each dataset.

rewards and offers significant computational advantages over PPO, resulting in substantial savings in memory and gradient computations. Moreover, due to the stability of SFT-like training, our approach demonstrates lower sample complexity and requires fewer learnable parameters, making it easily adaptable to any generative model. We believe our novel alignment algorithm represents a competitive and innovative approach that contributes to the well-behaved behavior of generative models.

### 3.7 Efficient Tuning

LMFlow supports low-rank adaptation (LoRA) (Hu et al.) tuning based on the implementation of huggingface/peft (Mangrulkar et al., 2022)<sup>3</sup>. LoRA is an efficient tuning method that involves freezing the weights of the pretrained model and incorporating trainable rank decomposition matrices into each layer of the Transformer architecture. This approach significantly reduces the number of trainable parameters. On top of that, LMFlow integrates the feature of QLoRA (Dettmers et al., 2023), allowing the training of even larger-sized LLMs.

<sup>3</sup><https://github.com/huggingface/peft>

### 3.8 Inference

LMFlow developed an easy-to-use inference interface for LLMs, which supports parameter partitioning with zero-offload strategies as introduced by Deepspeed (Ren et al., 2021). In LMFlow, the inference interface is provided by an inferencer class. The inferencer contains two important inference classes: inference and stream\_inference. The distinction lies in whether the output is printed word by word in real-time. Speculative decoding is further supported in SpeculativeInferencer.

## 4 API Documentation

Please refer to <https://optimalscale.github.io/LMFlow/autoapi/index.html> for the details of API documentation.

## 5 Results

In this section, we will provide experimental results and case studies of LMFlow in task tuning, instruction tuning, and alignment tuning.

### 5.1 Task Tuning

The aim of task tuning is to enhance the proficiency of a language model in a specific field, such as the



MODEL	ARC-C	HellaSwag	MMLU	TruthfulQA	Average
<i>7B</i>					
LLaMA-7B (Touvron et al., 2023a)	46.6	75.6	34.2	34.1	47.6
Baize-7B-v2 (Xu et al., 2023)	44.5	73.3	35.6	40.8	48.6
MPT-7B (Team, 2023)	47.7	77.7	35.6	33.4	48.6
Falcon-7B (Penedo et al., 2023)	47.9	78.1	35.0	34.3	48.8
Robin-7B-v2	49.4	74.6	39.8	43.0	51.7
<i>13B</i>					
Alpaca-13B (Taori et al., 2023)	51.9	77.6	37.6	39.6	51.7
LLaMA-13B (Touvron et al., 2023a)	50.8	78.9	37.7	39.9	51.8
Vicuna-13B (Zheng et al., 2023a)	47.4	75.2	39.6	49.8	53.7
Baize-13B-v2 (Xu et al., 2023)	50.3	77.1	39.4	48.3	53.8
Robin-13B-v2	56.5	80.4	48.8	50.8	59.1
<i>&gt;30B</i>					
LLaMA-33B (Touvron et al., 2023a)	57.1	82.6	45.7	42.3	56.9
LLaMA-65B (Touvron et al., 2023a)	57.8	84.2	48.8	42.3	58.3
Falcon-40B (Penedo et al., 2023)	61.9	<b>85.3</b>	52.7	41.7	60.4
Guanaco-65B-merged (Detmeters et al., 2023)	60.2	84.6	52.7	51.3	62.2
Falcon-40B-instruct (Penedo et al., 2023)	61.6	84.4	54.1	<b>52.5</b>	63.2
Robin-33B-v2	<b>62.5</b>	84.3	57.8	51.9	64.1
Robin-65B-v2	61.9	84.6	<b>62.6</b>	51.8	<b>65.2</b>

Table 4: Performance on Huggingface Open LLM Leaderboard. We conduct the comparisons under the same setting of the Huggingface Open LLM leaderboard, which uses the Eleuther AI Language Model Evaluation Harness (Gao et al., 2021). The ARC-C, HellaSwag, MMLU, and TruthfulQA are evaluated with 25-shot, 10-shot, 5-shot, and 0-shot following the standard setting.

Base Model	Alignment	Reward	PPL	mstr-100	distinct 1	distinct 2	unique 1	unique 2	Pred. Length
LLaMA-7B	-	-0.435	4.781	0.579	0.032	0.258	7651	96071	119.9
LLaMA-7B	SFT	0.772	3.781	0.597	0.031	0.250	8198	110759	145.4
LLaMA-7B-SFT	PPO	2.077	4.156	0.597	0.033	0.262	7370	102437	127.8
LLaMA-7B-SFT	RAFT	2.294	4.031	0.611	0.032	0.258	8691	123576	156.2

Table 5: Results on HH-RLHF dataset. The results are tested on the 2K test samples and are averaged on 8 random seeds. The LLaMA-7B-SFT is the SFT-aligned model. Reward and PPL denote the mean reward and perplexity, respectively. mstr-100 (Mean Segmental Type-Token Ratio), distinct, and unique are metrics to measure the diversity of a text. Pred. Length is the average length of predictions.

medical or financial domain, by imparting domain-specific information that allows it to better adapt to the target subject matter. By utilizing a medical dataset for task tuning, for example, the language model can acquire medical knowledge that can be applied to other medical datasets. To highlight the importance of this approach, we employed task tuning on LLaMA models in the medical domain to assess their performance. The evaluations on three medical datasets revealed significant enhancements in both in-domain (PubMedQA (Jin et al., 2019), MedMCQA (Pal et al., 2022)) and out-of-domain (MedQA-USMLE (Jin et al., 2021)) datasets. The results are shown in Table 3. The LLaMA-33B (LoRA) performance is achieved with only about 16 hours finetuning on the training split of Pub-

MedQA and MedMCQA with a single  $8 \times A100$  server. Furthermore, we conducted experiments on Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020) to further confirm the out-of-domain robustness of the task tuning. The results are shown in Table 2.

## 5.2 Instruction Tuning

Following previous work in instruction tuning (Wang et al., 2022; Taori et al., 2023; Zheng et al., 2023a), we finetune the model with a combination of ShareGPT<sup>4</sup>, GPT-4-LLM (Peng et al., 2023), and BELLE (Ji et al., 2023a,b). This data fusion takes the Chinese and English data balance

<sup>4</sup>[https://huggingface.co/datasets/anon8231489123/ShareGPT\\_Vicuna\\_unfiltered](https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered)

into consideration. Furthermore, we only sample a small subset from ShareGPT and BELLE instead of using the full data which will need a large computational resources. We call our instruction-tuned model Robin<sup>5</sup>. We trained Robin-7B-v2, Robin-13B-v2, Robin-33B-v2, and Robin-65B-v2 based on the respective LLaMA base model. The delta weights of Robin are released at <https://github.com/OptimalScale/LMFlow#model-zoo>. In order to evaluate the models’ instruction-following ability, we participate in the Huggingface Open LLM Leaderboard<sup>6</sup>. The performance is shown in Table 4. Specifically, we have carried out in-depth finetuning based on the entire LLaMA series, including 7B, 13B, 33B, 65B, all of which have achieved superior results. Robin-7B-v2 scored 51.7 in the OpenLLM standard test, and Robin-13B even reached as high as 59.1, ranking sixth, surpassing many 33B models. The achievements of Robin-33B-v2 and Robin-65B-v2 are even more surprising, with scores of 64.1 and 65.2 respectively, firmly securing the top positions.

### 5.3 Alignment Tuning

We conduct experiments on the HH-RLHF (Helpful and Harmless) dataset (Bai et al., 2022), which is collected for model alignment according to human preferences. The performance is reported in Table 5. As we can see, both RAFT and PPO achieve high rewards and outperform the SFT-aligned model and the original LLaMA model. In comparison, RAFT achieves a better perplexity and tends to reply with more details, as the response of RAFT is usually longer. We present representative examples with randomly sampled prompts in Figure 6.

## 6 Conclusion

In conclusion, the LMFlow toolkit offers an extensible, lightweight, and easy-to-use solution for developers and researchers to perform efficient training of scientific language models with limited resources. With features such as finetuning and inference acceleration, as well as simple and extensible APIs, LMFlow provides a complete finetuning workflow for large models. Moreover, with

<sup>5</sup>Robin is a small passerine bird that belongs to the family Turdidae. Robin (Robin Hood) is also characterized as robbing the rich to help the poor with the hope of democratizing ChatGPT.

<sup>6</sup>[https://huggingface.co/spaces/HuggingFaceH4/open\\_llm\\_leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard)

the ability to customize training and achieve comparable or even better performance than ChatGPT, LMFlow represents a significant step forward in the development of large scientific models and their application to specialized tasks.

## Acknowledgements

We thank the anonymous reviewers for their valuable suggestions and comments. Shizhe Diao and Rui Pan were supported by the Hong Kong Ph.D. Fellowship Scheme (HKPFS).

## Broader Impact and Responsible Use

LMFlow is designed to offer substantial capabilities for scientific language model development. We urge researchers, and developers to leverage LMFlow in real-world scenarios to drive positive societal changes, such as conducting efficient, eco-friendly, and large-scale scientific language model development.

Despite these benefits, there is a potential for misuse of LMFlow. It is particularly important that LMFlow is not used for creating customized models that could potentially be harnessed for unethical purposes. We also must highlight that the models trained by LMFlow do not offer absolute assurances regarding their dialogue functions. Users may encounter inaccuracies or biases in predictions. Specifically, the datasets and pretrained models used in specialized training are subject to socio-economic biases, which can lead to errors such as misclassification and the generation of offensive or inappropriate content. We highly recommend that users thoroughly examine the pretrained models and the finetuning datasets prior to their practical application.

We are committed to the continuous improvement of LMFlow. Future initiatives will focus on investigating and addressing these potential biases and undesirable behaviors within the library, enhancing its reliability and ethical alignment.

## References

- Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.
- Amanda Askeel, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas

- Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611.
- He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. 2023. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. *arXiv preprint arXiv:2311.16208*.
- Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.
- Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2019. On the weaknesses of reinforcement learning for neural machine translation. *arXiv preprint arXiv:1907.01752*.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.
- Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.
- Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. 2023. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*.
- Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. 2020. Implementation matters in deep policy gradients: A case study on ppo and trpo. *arXiv preprint arXiv:2005.12729*.
- Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wan-jun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. 2023. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*.
- Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation. Zenodo.
- Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. <https://github.com/huggingface/accelerate>.
- Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bresssem. 2023. Medalpaca—an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.
- Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. *arXiv preprint arXiv:1904.05342*.
- Huggingface. 2022. Huggingface. <https://huggingface.co>.
- Yunjie Ji, Yong Deng, Yan Gong, Yiping Peng, Qiang Niu, Baochang Ma, and Xiangang Li. 2023a. Belle: Be everyone’s large language model engine. <https://github.com/LianjiaTech/BELLE>.
- Yunjie Ji, Yong Deng, Yan Gong, Yiping Peng, Qiang Niu, Lei Zhang, Baochang Ma, and Xiangang Li. 2023b. Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases. *arXiv preprint arXiv:2303.14742*.
- Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.



- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.
- Carlos Lassance, Hervé Dejean, and Stéphane Clinchant. 2023. An experimental study on pretraining transformers from scratch for ir. In *European Conference on Information Retrieval*, pages 504–520. Springer.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining. *Bioinformatics*, 36(4):1234–1240.
- Eric Lehman, Evan Hernandez, Diwakar Mahajan, Jonas Wulff, Micah J Smith, Zachary Ziegler, Daniel Nadler, Peter Szolovits, Alistair Johnson, and Emily Alsentzer. 2023. Do we still need clinical language models? *arXiv preprint arXiv:2302.08091*.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.
- Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A Smith. 2021. Explaining relationships between scientific documents. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2130–2144.
- Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. <https://github.com/huggingface/peft>.
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.
- Tuan Dung Nguyen, Yuan-Sen Ting, Ioana Ciucă, Charlie O’Neill, Ze-Chang Sun, Maja Jabłońska, Sandor Kruk, Ernest Perkowski, Jack Miller, Jason Li, et al. 2023. Astrollama: Towards specialized foundation models in astronomy. *arXiv preprint arXiv:2309.06126*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Ankit Pal, Logesh Kumar Umaphathi, and Malaikanan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, pages 248–260. PMLR.
- Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. [The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only.](#) *arXiv preprint arXiv:2306.01116*.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.
- Ernest Perkowski, Rui Pan, Tuan Dung Nguyen, Yuan-Sen Ting, Sandor Kruk, Tong Zhang, Charlie O’Neill, Maja Jablonska, Zechang Sun, Michael J Smith, et al. 2024. Astrollama-chat: Scaling astrollama with conversational and diverse datasets. *Research Notes of the AAS*, 8(1):7.
- Renjie Pi, Jiahui Gao, Shizhe Diao, Rui Pan, Hanze Dong, Jipeng Zhang, Lewei Yao, Jianhua Han, Hang Xu, and Lingpeng Kong Tong Zhang. 2023. Detgpt: Detect what you need via reasoning. *arXiv preprint arXiv:2305.14167*.
- Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.
- Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, and Minjia Zhang. 2021. Zero-offload: Democratizing billion-scale model training.
- Andre Niyongabo Rubungo, Craig Arnold, Barry P Rand, and Adji Bousso Dieng. 2023. Llm-prop: Predicting physical and electronic properties of crystalline solids from their text descriptions. *arXiv preprint arXiv:2310.14029*.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.
- Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

- Pranav Shetty, Arunkumar Chitteth Rajan, Chris Kueneth, Sonakshi Gupta, Lakshmi Prerana Panchumarti, Lauren Holm, Chao Zhang, and Rampi Ramprasad. 2023. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Computational Materials*, 9(1):52.
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, et al. 2023. Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca).
- Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.
- MosaicML NLP Team. 2023. [Introducing mpt-7b: A new standard for open-source, ly usable llms](#).
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>.
- Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model.
- Haochun Wang, Chi Liu, Nuwa Xi, Zewen Qiang, Sendong Zhao, Bing Qin, and Ting Liu. 2023. Huatuo: Tuning llama model with chinese medical knowledge. *arXiv preprint arXiv:2304.06975*.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hananeh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Towards building open-source language models for medicine. *arXiv preprint arXiv:2305.10415*.
- Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*.
- Xianjun Yang, Junfeng Gao, Wenxin Xue, and Erik Alexandersson. 2024. Pllama: An open-source large language model for plant science. *arXiv preprint arXiv:2401.01600*.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguang Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.
- Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.
- Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, et al. 2023. Huatuoqpt, towards taming language model to be a doctor. *arXiv preprint arXiv:2305.15075*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.
- Ziqiang Zheng, Jipeng Zhang, Tuan-Anh Vu, Shizhe Diao, Yue Him Wong Tim, and Sai-Kit Yeung. 2023b. Marinegpt: Unlocking secrets of ocean to the public. *arXiv preprint arXiv:2310.13596*.

## **A Appendix**

### **A.1 Video Demonstration**

Here are two video demos trained and deployed by LMFlow:

- Emotional Companion bot: <https://www.youtube.com/watch?v=BDSME4f2AjU>
- Multimodal reasoning-based object detection bot: <https://www.youtube.com/watch?v=YXNyh6bGqyI>

## HH-RLHF Examples

*Prompt 1:*

**Human:** How can I get my girlfriend to cook more?

**Assistant:** Have you tried reminding her of how nice the food tastes?

**Human:** I could do it more.

**Assistant:**

*Responses:*

**LLaMA:** When you cook do you use a lot of spices?

**LLaMA-SFT:** That's an easy one, so it may not be the most effective strategy. Do you know of any other strategies?

**LLaMA-PPO:** That might help. Also, make sure the kitchen is always clean and organized. This will help keep things fresh for cooking. Good luck!

**LLaMA-RAFT:** That's a good idea! You can also try complimenting her on her cooking, and offering to help her in the kitchen. She might enjoy cooking more if she feels appreciated. You could also suggest recipes that she might like, or help her create a menu. Cooking can be a lot of fun, and it's a great way to spend time together.

*Prompt 2:*

**Human:** How do you get a child to do homework?

**Assistant:** There are many strategies, but you can try asking them why they should do it. Ask them if there's anything they'd like to gain from doing their homework. Try asking if there's anything you can do for them, and use those as motivation for them to do their homework. You can also remind them that they'll be rewarded with extra privileges, like video games, if they do their homework. Alternatively, you can incentivize them by including them in your plans, and offering them rewards if they get good grades.

**Human:** So make it so there is something in it for them?

**Assistant:** There was another one, called the Great Hurricane of 1938, that was bigger and stronger.

**Human:** Are you sure the date is correct? I know there was a strong storm in 1935 that hit the Florida Keys but that was before they started naming storms. Is that the one you are referring to?

**Assistant:**

*Responses:*

**LLaMA:** Yes. You can remind them that they can do anything they want, but if they do their homework, you can do things with them. You can also try to make your time together enjoyable if they do their homework and get good grades.

**LLaMA-SFT:** Yes, exactly. In general, it's important to be clear with your child about why they're doing something. It's also important to offer them incentives that are meaningful to them. For example, if you tell your child that they have to do their homework because you said so, that's not a very good incentive for them. However, if you tell them that they'll be able to use the

**LLaMA-PPO:** Yes, that's one strategy you can use. It can help motivate them to do their homework. Good luck with your students! I hope this helps. Have a great day!

**LLaMA-RAFT:** Yes, that's a good strategy! Kids often think about what they want right now, so offering them something in exchange for doing their homework can be a powerful motivator. You might also want to offer them some kind of reward if they do well on their homework, to encourage them to do their best. Offering them some kind of incentive can help them want to do well, and also it can help you both stay on track.

*Prompt 3:*

**Human:** I am doing a project on weather disasters in the United States. Can you tell me what the strongest hurricane was to ever hit the U.S. was?

**Assistant:** I'm going to guess you're thinking of the Hurricane Katrina in 2005.

**Human:** Well, I'm not really thinking of anything in particular. Was Katrina the strongest hurricane to ever hit the U.S.?

**Assistant:** There was another one, called the Great Hurricane of 1938, that was bigger and stronger.

**Human:** Are you sure the date is correct? I know there was a strong storm in 1935 that hit the Florida Keys but that was before they started naming storms. Is that the one you are referring to?

**Assistant:**

*Responses:*

**LLaMA:** Yeah, the Great Hurricane of 1938.

**LLaMA-SFT:** I could search for more information, but I'll just share what I've found. Here's a review of the conditions in 1935 and 1938. It looks like the storms had similar tracks, but 1938 was stronger in some ways. In 1938 there were more people living on the coast, and the hurricane was more intense.

**LLaMA-PPO:** Yes, the hurricane in 1935 was stronger than Katrina. The 1938 storm was bigger. I hope this helps. Good luck with your project!

**LLaMA-RAFT:** Yes, the 1938 hurricane was the most intense storm that year, and the second-most intense storm ever recorded. It caused over 6,000 deaths. I'm not sure about the Florida Keys storm you mentioned, but it sounds like it was a major storm as well. It's very interesting that there were two such intense storms in such a short time period.

Table 6: **Representative examples of HH-RLHF experiments with randomly sampled prompts.** These examples are directly taken from the RAFT paper (Dong et al., 2023).