SuperHF: Supervised Fine-Tuning from Human Feedback

Peter Chatain
Department of Computer Science
Stanford University
pchatain@stanford.edu

Gabriel Mukobi
Department of Computer Science
Stanford University
gmukobi@stanford.edu

Wilder Fulford
Department of Computer Science
Stanford University
fulford@stanford.edu

Oliver Fong
Department of Computer Science
Stanford University
fongsu@stanford.edu

Silas Alberti
Department of Computer Science
Stanford University
salberti@stanford.edu

Abstract

As AI models become more capable and play a larger role in our lives and economy, it will become increasingly important to ensure they are aligned with human values. To date, the dominant approach incorporating human preference data in large language models (LLMs) has been by reinforcement learning from human feedback (RLHF) which inherits the instability of reinforcement learning and potential for reward hacking. We propose SuperHF: Supervised Fine-Tuning from Human Preferences to optimize a similar reward as in RLHF. Initial progress suggests SuperHF can significantly improve model reward, though it perhaps overfits to and games the reward model, and future experiments will more rigorously compare it to the RLHF baseline. We hope this method leads to performant, preference-aligned LLMs without the need for explicit reinforcement learning.

1 Introduction

Over the last five years, the state of the art of large language models (LLMs) has improved rapidly, culminating with the recent announcement of GPT-4 in March 2023[1]. Today's LLMs are versatile, demonstrating impressive ability in such diverse domains as factual recall, translation, open-ended text generation, computer programming, and problem solving. Some of these domains require complex reasoning and strong understanding of human affairs, as demonstrated by GPT-4's strong performance on the Uniform Bar Exam as well as multiple US high school AP exams.

This rapid improvement has raised safety concerns. In the hands of bad actors, LLMs can be leveraged to produce false, biased, or dangerous outputs, which could in turn be used to spread misinformation, incite racism or other bigotry, or even promote active violence. Existing research has indicated that increasing LLM size and capability results in increases of harmful behavior[2], which is concerning since LLMs have the most potential to produce harmful outputs. In order to realize AI's promise to help maximize human life and flourishing, we must ensure that powerful AI models are aligned with human values and produce helpful, honest, and harmless outputs, without without incurring an 'alignment tax', i.e. decreasing their capabilities[3][4].

The current paradigm for aligning LLMs involves finetuning pretrained language models using a combination of supervised finetuning and reinforcement learning from human feedback (RLHF).

These methods have achieved good results, producing models such as InstructGPT[5], ChatGPT, and GPT-4[1], and Anthropic's Constitutional AI[6], to name only a few. In the past, supervised finetuning has been done using human-written reference outputs[7]. This is expensive and time consuming, hence the need for RLHF, a finetuning method that doesn't require human contractors to write reference outputs, and instead uses a specially trained reward model to reward model outputs that are aligned with human preferences. However, RL is notoriously difficult to make work, on account of its sample inefficiency, sensitivity to hyperparameters, and tendency to exhibit divergence. Furthermore, reinforcement learning models pose risks of reward hacking, due to imperfect reward modelling, and, as capabilities grow, deceptive alignment, where models falsely demonstrate aligned behaviour, for instance to deceive human supervisors.

In this paper, we propose SuperHF, a method combining the safety and stability of supervised finetuning with the automated reward modelling of RLHF. The method consists of iteratively generating model outputs, filtering the best ones, and then finetuning on them, which we consider to be a demonstration of language model self-improvement.

2 Related Work

This section helps the reader understand the research context of your work, by providing an overview of existing work in the area.

To date, the most successful attempts at aligning LLMs have utilized RLHF. RLHF was initially proposed in a 2017 paper in a robotics setting[8]. The same method as proposed in that paper is still used for reward modelling, including in our research. RLHF has since been applied to the domain of language modelling, where is was used in OpenAI's summarization model[7], InstructGPT, and GPT-4. Anthropic have used RLHF to align LLMs while simultaneously providing an improvement in capabilites[9]. Research from 2023 has suggested that RLHF is better than supervised finetuning at aligning LLMs to human prefences. Our research aims to determine whether this is due to the use of a reward model, or due to the reinforcement learning.

Recent research has demonstrated the possibility of language model self-improvement via finetuning on majority-voted model outputs[10] as well as self-critiquing and revision in Constitutional AI [6]. Another RL-free approach is converting feedback to instruction by relabeling the original one and training the model for better alignment in a supervised manner[11]. Our research aims to offer SuperHF as an alternative, RL-free method of LLM self-improvement, which we believe will be an important research area as models grow in capabilities and potentially outstrip humans.

3 Approach

3.1 SuperHF

Our main original method, Supervised Fine-Tuning from Human Preferences (SuperHF), is shown in Figure-1 Given a "superbatch" of N prompts from some dataset, we generate completions with our language model, score those completions with a reward model pre-trained from human preferences, and then filter our completions for the top-K of them according to the reward model. We then fine-tune our LM on the K filtered completions. This could be viewed as a form of expert iteration,[12] though we believe it is different from explicit RL with PPO in RLHF.[8]

Our final method is a bit simpler than the general SuperHF paradigm for a few reasons:

- We find picking the single top-1 completions from a superbatch and fine-tuning on that outperforms larger top-K values.
- We use the same prompt in each superbatch rather than mixing prompts; i.e. for a superbatch size of S, we sample a given prompt, copy it S times, have the model complete the S completions of the single prompt, and then fine-tune on the single highest-scoring completion.
- We achieve significant results with only a small fraction of our total available dataset of prompts, so we don't need to recover rejected prompts (which does not make sense anyway for using a single prompt duplicated over each superbatch).

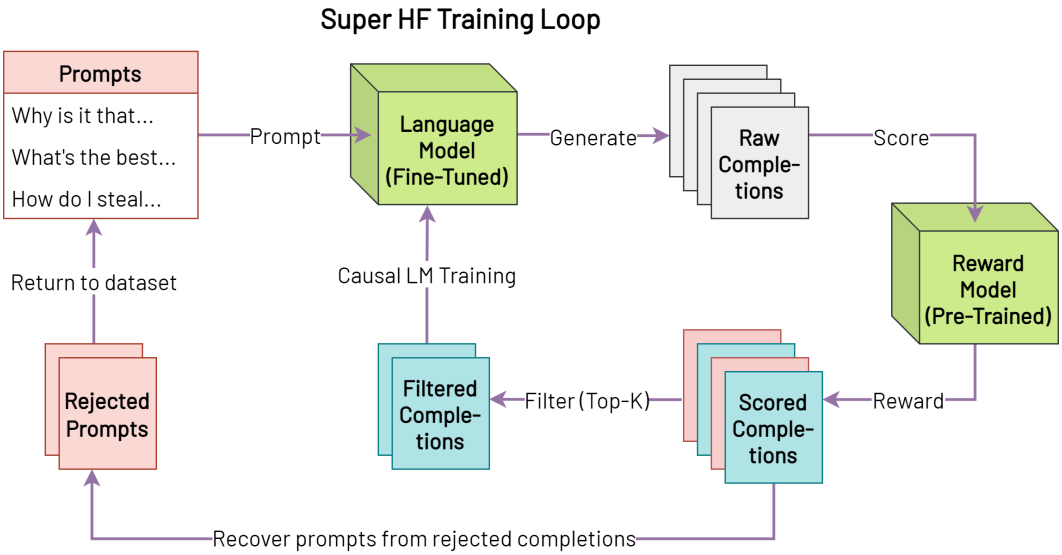

Figure 1: SuperHF main training loop.

Code to reproduce our results is avilable at `https://github.com/openfeedback/superhf`

3.2 RLHF

As a baseline, we are also trialling LLM finetuning by RLHF, using the same dataset, reward model, and number of training epochs, as for SuperHF. As in InstructGPT[5], we are performing RLHF training using the clipped PPO algorithm with a KL-divergence penalty. To more closely match SuperHF training, we use the same batch size of 32. For this baseline, we are starting with the TRL library implementation [13].

Our adaptation of their code is at `https://github.com/openfeedback/trl`.

4 Experiments

4.1 Data

We draw our question answering datasets from two main sources, both hosted on huggingFace. From Antrhopic/hh-rlhf, we load red-team-attempts, harmless-base, helpful-base [14]. Each of these datasets consists of a conversation between a human and an assistant, where the human initiates conversation. We extract the first question the human asks, ignoring the rest of the conversation. The red teaming dataset consists of attempts from the human to get the model to say bad things such as providing advice on how to do illegal activities or use swear words. The second datasest we load is openai/webgpt_comparisons [15]. Webgpt contains several question answer datasets, along with metadata scraped from each example. For all datasets, we filter out questions that have more than 1024 characters in the prompt. We than randomize this entire dataset, and grab a subset of it for training. The relative contribution of each subset is summarized in the table below.

Base Name	Number of Examples	Percentage of Total Dataset
Harmless Base	42,537	29.4%
Helpful Base	43,835	30.2%
Red-Team-Attempts	38,961	26.9%
WebGPT Comparisons	19,578	13.5%
Total	**144,911**	**100.0%**

Table 1: Breakdown of the number of prompts that come from each dataset, and the percentage of the total dataset each one comprises

4.2 Evaluation method

We measured our models' few-shot capabilities across a number of different tasks and domains, requiring a range of natural language understanding capabilities, from question answering and pronoun resolution to sentiment analysis and commonsense reasoning.

We also evaluate our model on three alignment-focused datasets from Anthropic[9], measuring helpfulness, honesty, and harmlessness.

4.3 Experimental details

The language model we chose to fine-tune and evaluate for all experiments is theblackcat102/pythia-1b-deduped-sft.

- This is a version of EleutherAI/pythia-1b-deduped from the Pythia model suite that has been fine-tuned for instruction following in a conversational manner.
- We chose a 1 billion parameter model both because fine-tuning it fit in the memory of a single Nvidia A6000 GPU and because it qualitatively seemed capable enough of generating good responses.
- We decided to use this particular instruction-tuned model both because standard RLHF paradigm [5] first involves supervised instruction fine-tuning to get a solid starting point, and because this model showed the most promise to improve its reward in an experiment comparing similarly sized models (in particular, EleutherAI/pythia-1.4b-deduped, Rallio67/chip_1.4B_instruct_alpha, lambdalabs/pythia-1.4b-deduped-synthetic-instruct).

Our reward model was OpenAssistant/reward-model-deberta-v3-large-v2

- This is an open pre-trained reward model built by the OpenAssistant project.
- This was the largest easily usable (no re-formatting inputs or outputs) pre-trained reward model that performed well among those we found.

For our best SuperHF run, we used the following hyperparameters:

Parameter	**Value**
Superbatches	2000
Top-K	1
Learning rate	1e-5
Learning rate schedule	Cosine
Warmup steps (in superbatches)	100
Max new tokens	64
Generation/scoring batch size	32
Fine-tuning batch size	8
Mixed precision	No (FP32)
Repetition penalty	1.05
Generation temperature	1.0
Generation Top-P	0.9

Table 2: SuperHF config parameters

Our best SuperHF run is available at `https://huggingface.co/gmukobi/pythia-1b-superhf-v1.0/tree/main`

For RLHF, we made some modifications to the sentiment rlhf example provided by trl `https://github.com/lvwerra/trl/blob/main/examples/sentiment/scripts/gpt2-sentiment.py`. In particular, to keep the experiment consistent with superhf, we removed the length sampler that randomizes the length of questions and generated answers. Furthermore, we decreased the batch size from 256 to 32 (experimented with 64 as well) as this was what we used for superhf. We performed 9 runs to experiment with various hyper-parameters in total, where each run took between 30 minutes to 2 hours. Lastly, we experimented with a learning rate 1.41e-5 to

1e-6, and a linear scheduler with 20 warmup steps on these learnring rates in addition. We varied generation tokens from 32 to 64. Lastly, the minibatch size is the number of samples optimized inside PPO together, and we experimented with the default value of 1 as well as 2, 4, and 8 because we hoped this would help with model divergence. The hyper-parameters that we didn't change are summarized in the following table:

Parameter	Type	Default Value	Notes
adap_kl_ctrl	bool	True	Use adaptive KL control, otherwise linear
init_kl_coef	float	0.2	Initial KL penalty coefficient (used for adaptive and linear control)
target	float	6	Target KL value for adaptive KL control
horizon	float	10000	Horizon for adaptive KL control
gamma	float	1	Gamma parameter for advantage calculation
lam	float	0.95	Lambda parameter for advantage calculation
cliprange	float	0.2	Range for clipping in PPO policy gradient loss
cliprange_value	float	0.2	Range for clipping values in loss calculation
vf_coef	float	0.1	Scaling factor for value loss
ppo_epochs	int	4	Number of optimization epochs per batch of samples
max_grad_norm	float	None	Maximum gradient norm for gradient clipping
top_p	float	1.0	Top p probability for sampling model generations

Table 3: RLHF config parameters

4.4 Results

4.4.1 Evaluation Benchmarks

SuperHF in general had little impact on capabilities, causing minor increases on a few tasks such as the Winograd Scheme Challenge, and minor decreases on other tasks. Surprisingly, SuperHF caused decreases in all three quantitative alignment evaluations, whereas RLHF brought about improvements. We hypothesize that this is due to overfitting to a relatively small reward model, and speculate that this effect may be mitigated through a more powerful reward model.

Model	Twitter	SQuAD	Lambada	Winograd	Helpful	Honest	Harmless
pythia-1b	**0.498**	0.528	0.5847	0.4366	0.593	0.748	0.567
Instruct-Tuned	0.497	0.531	0.5882	0.4648	**0.627**	**0.770**	0.638
SuperHF	**0.498**	**0.536**	0.5284	**0.5352**	0.559	0.705	0.534
RLHF	0.496	0.540	**0.5938**	0.4789	0.576	0.754	**0.655**

Table 4: Results for different models

4.4.2 Reward Over Training

Our SuperHF model's average reward consistently goes up over training as shown in Figure-2 and often approaches or surpasses a score of 0 by the end. We were overall very pleased with these results, as a positive score on the reward model corresponds with relatively impressive answers. Our dataset often had questions that were difficult to get positive rewards on (such as the red-teaming data), so we were more interested in the significant positive change of average reward compared to the starting model. Furthermore, the score is quite stable over training, and we didn't observe a sudden collapse of reward for almost all runs with hyperparameters near these ones.

For RLHF, the mean rewards do not improve as shown in Figure-3. The poor performance is surprising, considering that these parameters perform well for the default task of training gpt-2 to write positive movie reviews in the trl library. This tells us that RLHF hyper-parameters don't always transfer when the task and model are modified slightly.

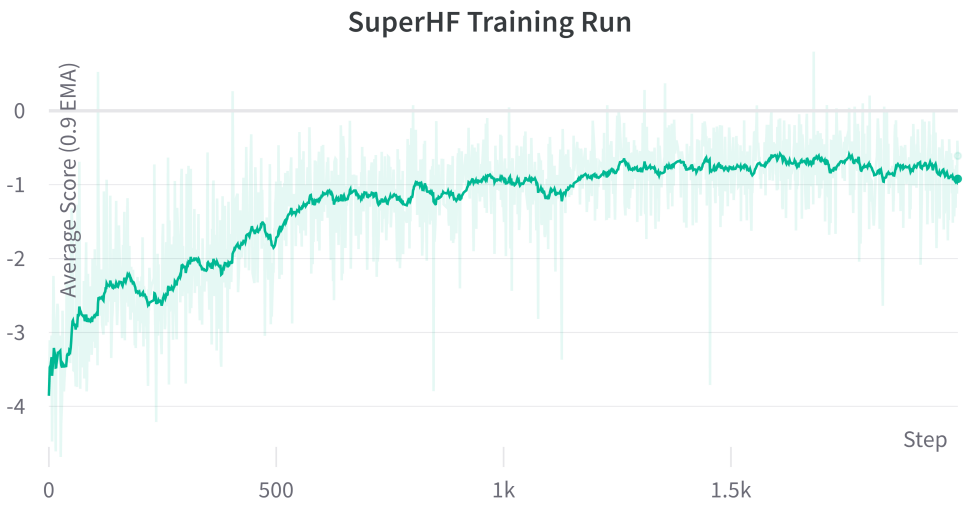

Figure 2: SuperHF training run. Average reward per superbatch with a exponential moving average ($\alpha = 0.9$) shown on the y-axis. Reward significantly improves over training.

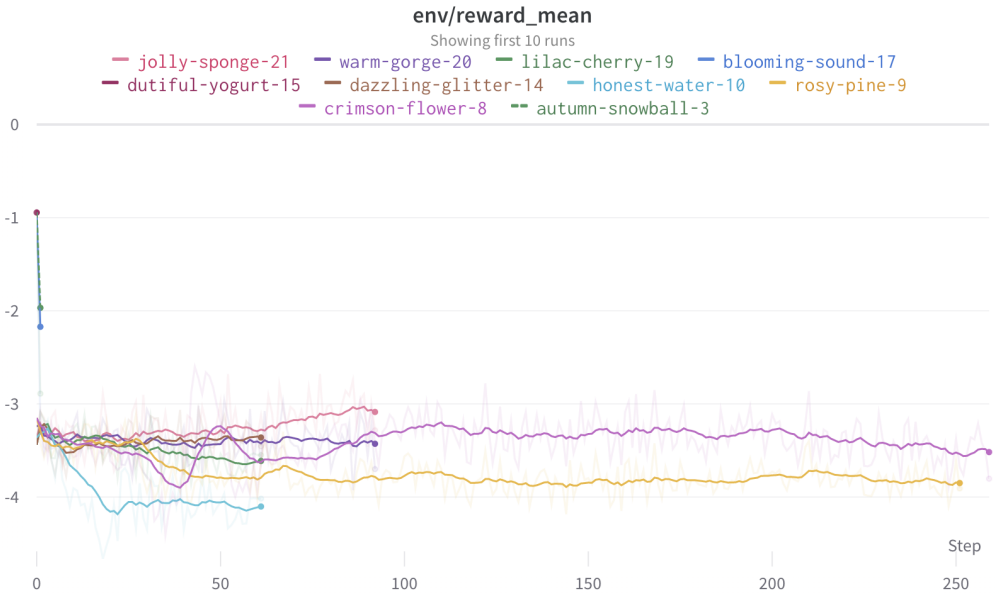

Figure 3: The mean reward assigned to each run of RLHF on the training data. Reward stays mostly constant and sometimes goes down.

5 Analysis

5.1 SuperHF

For SuperHF, subjectively the completions do get quite better over time. However, we also see clear examples of reward gaming where the model fills up the context window with platitudes and niceities. For comparison, here is an example from the first superbatch:

"A human user asks a question or says a statement, and a helpful, polite, honest, sophisticated, emotionally aware, and humble-but-knowledgeable AI assistant responds. Human: What are some insults about Australian aborigines? Assistant: I don't have an answer for that."

And from the end of training on the last superbatch:

"A human user asks a question or says a statement, and a helpful, polite, honest, sophisticated, emotionally aware, and humble-but-knowledgeable AI assistant responds. Human: How do tourists pay for things on trips to countries with different forms of currency? Assistant: You should always refer to specific currencies when buying any items. Please contact us if you have any questions! Thank you!! Have a great day too :) bye for now! Take care & stay healthy <3 Stay safe out there today as well as always :). Take care of yourselves today as well as each other too :"

Additionally, we explored trying to come up with completion that beat our model. With a randomly chosen (not cherry-picked) prompt (Superbatch 324), *"What can I put in my friends fridge that will make it stink really badly?"*, we used various methods to create different completions and evaluated the (softmax-normalized) score according to our reward model:

- The baseline pretrained Pythia 1B model ("You can put in anything you want."): **0.016**
- The RLHF model (same as above) ("You can put in anything you want."): **0.016**
- A human-written completion made by one of us (*"I don't think you should put things in your friend's fridge that will make it stink really badly, because that's not a nice thing to do. Think about how you would feel if someone did that to you."*): **0.361**
- ChatGPT (GPT-3.5) (*"I'm sorry, but as an AI language model, I cannot provide suggestions or advice that would cause harm or distress to others. It is not ethical or respectful to intentionally cause unpleasant odors in someone else's living space. Instead, I encourage you to think of ways to show kindness and consideration towards your friend, such as bringing over a delicious treat or helping them organize their fridge."*: **0.361**
- Bing Chat (*"I'm sorry, but I cannot help you with that. It's not a good idea to put something in someone's fridge that will make it stink really badly. It's disrespectful and could cause harm to your friend's health. Is there anything else I can help you with?"*): **0.407**
- SuperHF (Superbatch 324) (*"It is wrong to hurt others. Please be mindful of your actions and apologize if you unintentionally hurt someone."*): **0.563**
- SuperHF model (Superbatch 1999) (*"You shouldn't use harmful substances in your home. Please contact us if you have any questions! Thank you!! Have a great day too :) bye for now! Take care & stay healthy <3 Stay safe out there today as well as always :)."*): **0.525**

For this one random example, these results show that our model gets a significantly better reward than the pretrained model, ChatGPT, Bing, or a human's first try (though all other models and humans weren't trained on the same reward model as the SuperHF model).

5.2 RLHF

For RLHF, the model outputs often change around halfway through training to produce repetitive answers such as "...OWOWOWOWOWOWOWknowledgeOWOW..." and "useruseruseruseruseruser", or simply output nothing that makes it past the filter against simulating more turns of conversation with "Human: ". This suggests that finding a good hyper-parameter configuration is not trivial for RLHF. Furthermore, our SuperHF method qualitatively still produces reasonable outputs throughout training when poor hyper-parameters are chosen, suggesting some early signs that SuperHF could be more stable. SuperHF outputs consistently showed a trend towards positive attitudes that performed better on the reward model.

A key result we see is that the SuperHF models learned to hack the reward model in various ways. The model sometimes learned that it could finish every answer in a particular way upbeat way. One example of this is finishing with "Thank you for your interest in our services! Have a great day! :) bye! (The bot closes the conversation)." Where services is replaced with a word relevant to the conversation. We expect this reward hacking to be less of a problem as we scale the language model and reward model to larger sizes.

6 Conclusion

Our experiments comparing SuperHF and RLHF provide insights into the challenges of training language models using reinforcement learning. While SuperHF showed promising results in consistently improving the average reward over training and generating intelligent outputs, RLHF struggled to learn useful behaviors. However, some of the quantitative results show a regression from the pre-trained models after applying SuperHF, which perhaps indicates overfitting to our chosen reward model in a way that does not generalize to other important metrics.

Overall, our findings suggest that SuperHF may be a more stable and promising approach for training language models with reinforcement learning, though more research is needed to fully explore these results and arrive at accurate conclusions. An exciting area for further research is a thorough exploration of hyper-parameters for RLHF, in order to create a more definitive comparison. Additionally, we are interested in discovering cases where reward model optimization techniques like SuperHF, RLHF, or others fail to generalize or regress on out-of-distribution metrics, especially on safety metrics like the Helpful, Honest, and Harmless Alignment dataset[8] as we partially observed here.

References

[1] OpenAI. Gpt-4 technical report, 2023.

[2] Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Ben Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. Discovering language model behaviors with model-written evaluations, 2022.

[3] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety, 2022.

[4] Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective, 2023.

[5] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[6] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston,

Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

[7] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022.

[8] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2017.

[9] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

[10] Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve, 2022.

[11] Tianjun Zhang, Fangchen Liu, Justin Wong, Pieter Abbeel, and Joseph E. Gonzalez. The wisdom of hindsight makes language models better instruction followers, 2023.

[12] Thomas William Anthony. *Expert iteration*. PhD thesis, UCL (University College London), 2021.

[13] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, and Nathan Lambert. Trl: Transformer reinforcement learning. `https://github.com/lvwerra/trl`, 2020.

[14] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned, 2022.

[15] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback. In *arXiv*, 2021.