Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023b. URL http://arxiv.org/abs/2307.09288.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63, Copenhagen, Denmark, 2017. Association for Computational Linguistics. doi:10.18653/v1/W17-4508. URL https://aclanthology.org/W17-4508.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.340.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions, 2023. URL http://arxiv.org/abs/2212.10560.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SJeYe0NtvH.

Tian Xu, Ziniu Li, Yang Yu, and Zhi-Quan Luo. On Generalization of Adversarial Imitation Learning and Beyond, 2022. URL http://arxiv.org/abs/2106.10424.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. RRHF: Rank Responses to Align Language Models with Human Feedback without tears, 2023. URL http://arxiv.org/abs/2304.05302.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open Pre-trained Transformer Language Models, 2022. URL http://arxiv.org/abs/2205.01068.

Tianjun Zhang, Fangchen Liu, Justin Wong, Pieter Abbeel, and Joseph E. Gonzalez. The Wisdom of Hindsight Makes Language Models Better Instruction Followers, 2023. URL http://arxiv.org/abs/2302.05206.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena, 2023. URL http://arxiv.org/abs/2306.05685.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (eds.), *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pp. 1097–1100. ACM, 2018. doi:10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-Tuning Language Models from Human Preferences, 2020. URL http://arxiv.org/abs/1909.08593.

A Limitations and Future Work

Here we discuss some potential limitations of our work and possible future directions for research pointed to by our results. While our work shows the effects of RLHF on generalisation and output diversity empirically, we do not provide a theoretical explanation for these results. Furthermore, while we demonstrate results on multiple base models and tasks, more combinations of base models and tasks could be experimented on, as well as other methods. Future work could investigate whether these effects are more general and why they arise.

Our work also only investigates SFT, RLHF and BoN as methods for fine-tuning language models with human preferences, but there are many other methods as described in Appendix C. Understanding the effects of these methods on generalisation and output diversity would be beneficial, especially if some of those methods are able to provide the generalisation benefits of RLHF without harming output diversity to the same extent.

Finally, we only evaluate our models on automatic metrics and do not perform any human evaluation (although we validate that our metrics align well with human preferences in Appendix D.1). While automatic metrics are useful for comparing models, they are not a perfect proxy for human judgement. Future work could investigate the effects of RLHF on human judgement of model outputs.

B Broader Impact

Large Language Models are increasingly used in production systems, and so it is important to understand the effects of different fine-tuning methods on the properties of the resulting models. Our work shows that RLHF produces better-generalising models than SFT, but those models are less diverse. This could be beneficial for some use cases, but harmful for others. For example, if a model is being used to generate text for a chatbot, it is important that the model is able to generalise to new inputs, but also that it is able to produce diverse outputs. On the other hand, if the model is being used to generate text for a summarisation system, it is important that the model is able to generalise to new inputs, but less important that it is able to produce diverse outputs.

C Related Work

More RLHF and SFT details Often, SFT and RLHF are combined by performing SFT followed by RLHF (Glaese et al., 2022; Menick et al., 2022; Nakano et al., 2022; Ouyang et al., 2022). We call the process of doing SFT followed by RLHF "The RLHF Pipeline", as it's the standard approach used in the literature and in deployed products (that use RLHF) (OpenAI, 2022; Anthropic, 2023). Other work has directly used RLHF on top of a prompt-distilled language model (Bai et al., 2022). Prompt distillation gathers demonstrations from a prompted version of a base model, and then performs SFT on the base model with these outputs, effectively fine-tuning the model to behave as if it was always prompted.

Ramamurthy et al. (2022) introduced an adaptation of PPO specifically for RLHF called *Natural Language Policy Optimisation* (NLPO), which calculates an action mask with top-p sampling and applies this to the language model, resulting in slightly improved performance on a range of tasks when using automated reward functions (not trained with human preferences for the task). Ramamurthy et al. (2022) demonstrate that RL generally outperforms SFT, but that their combination performs the best. However, their work only investigates relatively small models (220 million parameters), does not evaluate OOD performance or diversity, and does not use reward functions trained with human feedback, instead using mostly hard-coded reward functions from the literature. While hard-coded reward functions can sometimes be useful, RLHF is most widely applied in settings where we do not have a hard-coded reward function, and hence need to learn one from human data.

Large Language Models (LLMs) have recently been shown to solve a wide variety of language-based tasks, often without additional gradient-based training. Examples of such models include GPT-3 (Brown et al., 2020), Gopher(Rae et al., 2022), Chinchilla (Hoffmann et al., 2022), OPT(Zhang et al., 2022) PaLM (Chowdhery et al., 2022), Claude (Anthropic, 2023) and LLaMa(Touvron et al., 2023a). These models are trained on large amounts of text data, with a simple language modelling objective, and can often be prompted to perform tasks zero-shot or few-shot, without additional

training (Brown et al., 2020). Such tasks include translation, question-answering, and other standard NLP tasks, as well as newer tasks such as using LLMs to simulate human annotators (Dubois et al., 2023; Mao et al., 2023; Liu et al., 2023b) or as content generators for improving other models (Peng et al., 2023; Wang et al., 2023).

Other methods for fine-tuning LLMs Recently, multiple approaches for fine-tuning LLMs have been proposed: *Chain of Hindsight* (Liu et al., 2023a) fine-tunes models using SFT on sequences of increasingly better outputs for a given input; *CoOp CARP* (Castricato et al., 2022) uses a dataset of story-critique pairs combined with contrastive learning and pseudo-labelling to learn a preference model that is then used in the RLHF pipeline; *RRHF* (Yuan et al., 2023), uses a RM to rank multiple outputs from the model, and then optimises the model with weighted SFT, with a negative weight (similar to unlikelihood training (Welleck et al., 2020)) on lower-ranked samples; *HIR* (Zhang et al., 2023) relabels outputs using a goal-conditioned reward function or feedback function and then trains a goal-conditioned policy on these outputs (similar to (Andrychowicz et al., 2017)); and *ILF* (Scheurer et al., 2023), which uses natural language human feedback to prompt the model to produce better outputs than its original inputs, and then optimises the model with SFT on this dataset of improved outputs. While these works sometimes show improvements, they are not used by most large-scale systems being deployed currently, and hence we focus our analysis on the more popular and widely used RLHF pipeline, as that is where understanding will be most relevant and useful.

Possible Explanations for Results. Xu et al. (2022) present results for adversarial imitation learning (AIL) as compared to behavioural cloning (BC) in a more classical RL setting, showing that often AIL methods can generalise better than BC because they optimise the policy on out-of-distribution states. Mapped to the LLM fine-tuning regime, AIL is somewhat analogous to RLHF, and BC is identical to SFT, so this result may somewhat explain why RLHF generalises better than SFT.

Goldberg (2023) hypothesises that RLHF may generalise better than SFT because RLHF does not force models to produce outputs that are not implied their internal world model (to the extent that exists), whereas SFT trains models to produce outputs even if the model "believes" those outputs to be false.

D GPT-4 Evaluation Details

For GPT-4 evaluations for both summarisation and instruction following, we use the AlpacaEval (Li et al., 2023) software package to query GPT-4. For the instruction-following prompts, we use the standard annotator configuration recommended for that dataset, `alpaca_eval_gpt4`. For summarisation, we use the same configuration, but change the prompts to utilise a variant of those provided in (Rafailov et al., 2023). For the exact prompts see Figs. 7 and 8.

D.1 Validating GPT-4 evaluation

Summarisation. We validate the use of our GPT-4 evaluator for summarisation in two ways. First, we use the evaluator to label the preference validation datasets for both TL;DR and CNN/DailyMail released by Stiennon et al. (2022) and measure their accuracy. On TL;DR our evaluators gets 71.7% accuracy and on CNN/DailyMail it gets 65.5% accuracy. Comparing this to the inter-annotator agreement reported by Stiennon et al. (2022) or $\tilde{7}0\%$ for TL;DR and $\tilde{6}6\%$ for CNN/DailyMail demonstrates that our annotator has strong agreement with the human raters that generated the preference data we use for training our reward models.

For the second validation of GPT-4 as an evaluator, we measure the agreement between human labellers and GPT-4 on a subset of 25 inputs for every test set we use, comparing both SFT and RLHF model outputs with the reference output. This results in a total of 100 datapoints, each labelled by two human labellers giving 200 total annotations. This tests whether GPT-4 is in agreement with human preferences on the models we evaluate in this work.Table 1 shows the preference rating for GPT-4 and human labellers for each dataset and model, and the agreement between labellers and GPT-4, and we see that both at an aggregate level and at a per-example level our GPT-4 evaluator has good agreement with expert labellers.