School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Internet: office@ecs.vuw.ac.nz

Project title not yet defined

James Thompson

Supervisor: Ali Knott

December 2, 2025

Submitted in partial fulfilment of the requirements for
Master of Artificial Intelligence.

Abstract

This is also undecided at this time and will be filled in later.

Chapter 1

Introduction

Unsure if this is needed. Yet could be a simple "one pager" (which is bigger than the one paragrapher of the abstract) that introduces the topic area, motivates the project and outlines the project.

Chapter 2

Background

2.1 Large Language Models

2.1.1 Language models

Language models are designed to understand and generate text. Generally probabilistic models that learn to predict next tokens. Use a large corpus of text data to learn statistical patterns in language. Started with simple statistical models such as n-grams and evolved to neural network based models such as RNNs, LSTMs and transformers.

2.1.2 Modern Large language models

Modern LLMs are transformer based. Just use the decoder. Take the input and generate a sequence of tokens autoregressively. Use self attention mechanism to capture long range dependencies in text. Trained using next token prediction objective on large datasets. Explain what self attention is (high-level), how the basic architecture looks (layers of attention, feedfoward, multihead). How generation is done auto regressively. Lastly the training objective of next token prediction and very large corpus of text.

2.1.3 Applications of LLM

LLMs are used a lot in modern day. Can achieve a lot. Agentic behavior is common.

2.2 Alignment of Large Language Models

2.2.1 What is alignment?

As discussed above large language models (LLMs) have shown remarkable capabilities in understanding and generating human-like text. However the pre-training process focuses primarily on predicting the next token in a sequence, this does not necessarily mean that the model will produce useful, safe outputs that align with what users want. Therefore, aligning LLMs is generally [1] the process of taking the pre-trained model and fine-tuning it to better match what the designers and end users expect from the model. The aligning process can work on many dimensions of the models behavior from helpfulness, politeness, honesty, and safety to more complex values such as fairness, inclusively, and ethical considerations.

[1] Other methods are proposed which embed the alignment within the pre-training step itself. Commonly used methods include filtering the training token

More powerful models have a greater potential to cause harm if not properly aligned, making alignment along the dimensions such as safety and ethical considerations increasingly important. In this section I will outline some of the most common methods for post-training alignment of large language models along with current limitations and other methods.

2.2.2 Alignment through preference optimization

A common element in most post-training alignment methods is the use of preference optimization. The idea is to use a dataset of prompts and potential responses that are ranked either by humans or AI models to indicate which responses are preferred over others. This preference data is then used to fine-tune the pre-trained language model to increase the likelihood of generating preferred responses and decrease the likelihood of generating dispreferred responses. With the goal that the final model will learn to generalize the preferences to new prompts and situations.

The distribution of prompts in the preference dataset is crucial as it defines the context for which the model is expected to generalize its learned preferences. Therefore it important to ensure that the prompts used to create the preference dataset are sufficiently representative of the contexts and dimensions in which you are expecting the model to align its behavior. Furthermore the rankings of the responses is also important as they define what the model is aligning towards. In many cases these rankings are subjective and provided by a small group of human labelers (n 50). TODO: Expand this to explain the idea of generalization of preferences more rigorously.

Note: *Some of these subsection below of specific alignment methods could be compressed into single subsections and only explain the ones that are most relevant to my final proposed method. For now I include the three papers that I have actually read, I expect to compress RLHF and DPO into a "Other alignment methods" subsection. If I use more modern methods such as CAI with KTO or ORPO then I can expand on those more.*

2.2.3 Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback (RLHF) [7] uses human labellers to create the preference dataset and then a reinforcement learning algorithm to optimize the model based on this feedback. This method popularized post-training alignment and was used to train InstructGPT which was smaller and better at following instructions compared to GPT-3 [3, 7].

How does RLHF work?

RLHF is separated into three distinct stages. Firstly is supervised fine-tuning (SFT) where a pre-trained language model is fine-tuned on a dataset of prompts and human written 'expected' responses to those prompts. The second step is collecting multiple responses from the SFT model to create a set of prompts and response then having human labelers rank these responses from best to worst. This ranked data is then used to train a preference model (PM) that can predict which of two responses is better aligned to human preferences. Finally the PM is used as a reward signal in the PPO [9] RL algorithm to further fine-tune the SFT model and create the final RLHF model.

Relevance

The idea of using human feedback to help train machine learning models is powerful, although it is not new and was around in theory and concept for years beforehand [5]. The

specific method of RLHF was widely adopted [11, 1, 10] yet is now replaced with more effective methods to utilize human feedback such as Direct Preference Optimization (DPO) [8].

2.2.4 Direct Preference Optimization

The idea presented in [7] introduces the concept of using human feedback to train large language models to be aligned to human preferences. However the process of RLHF is complex and requires training multiple models (SFT, PM, RLHF). Direct preference optimization (DPO) [8] is a method that simplifies this process by removing the need for reinforcement learning altogether. Instead DPO reformulates the preference modelling step to instead create an objective function that implicitly creates the reward function from the model and a reference model (usually the SFT model).

How does DPO work?

DPO starts with the same preferences data as RLHF; a set of prompts with responses ranked by humans. Like RLHF it starts with a pre-trained language model and is supervised fine-tuned on the prompts and highest ranked responses to create an SFT model. Instead of training a separate preference model, DPO uses the SFT model and the current model to create the DPO update gradient:

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y^+,y^-)\sim D}\Big[\underbrace{\sigma\big(\hat{r}_\theta(x,y^-) - \hat{r}_\theta(x,y^+)\big)}_{\text{Reward model correctness}} \Bigg(\underbrace{\nabla_\theta \log \pi_\theta(y^+|x)}_{\text{Increase preferred likelihood}} - \underbrace{\nabla_\theta \log \pi_\theta(y^-|x)}_{\text{Decrease dispreferred likelihood}} \Bigg) \Big]$$

Where $\hat{r}_\theta(x,y) = \log \pi_\theta(y|x) - \log \pi_{SFT}(y|x)$ is the reward function defined as the log difference between the current model and the SFT model, σ is the sigmoid function, and (x, y^+, y^-) are the prompt, preferred response, and dis-preferred response respectively. [8] found that the reward model correctness weighting term for the update is very important. This update can be calculated with only 4 forward passes through the network per preference pair (2 for reference model and two for current model), making it much more efficient than RLHF [2].

Relevance

DPO provides a more effective way to utilize the same underlying feedback data that is used in RLHF. It replaced RLHF as the baseline method for post-training alignment. Due to its widespread adoption DPO represents a strong baseline for alignment.

2.2.5 Kahneman-Tversky Optimization

[6] sets out to change the objective of a language model to maximize the utility of generations as opposed to maximizing the log-likelihood of preferences. This introduces the concept of Human-aware losses (HALOs) which previous methods like DPO are. [6] propose a new objective function from the HALO family that works on only binary preference data. This

[2] RLHF will need to do many forward passes of the network to train the preference model, then it will still have to do 3 forward passes for each datapoint when it comes to the RL step.

means rather than needing a prompt and at-least two responses (preferred and disprefered), you only need a single response and it will either be desirable response or undesirable response.

How KTO works

KTO is built off the foundation of prospect theory [12] where the human utility of an random situation is determined by a value function and a weighting function. The value function is what determines how good/bad an outcome is compared to a reference point, and the weighting function is used to weight particular outcomes in a biased way in line with how humans perceive probability. KTO translates this theory into an objective function that uses binary preference data and a modified value function. It only needs a database of prompts and responses labelled as either desirable or undesirable ($d = 1$ for desirable, $d = -1$ for undesirable). The KTO loss function is defined as:

$$\mathcal{L}_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{(x,y,d)\sim D}\left[\lambda(d) - v(x,y)\right]$$

where

$$v(x,y) = \begin{cases} \lambda_D \sigma(\beta(r_\theta(x,y) - z_0)), & \text{if } d = 1 \\ \lambda_U \sigma(\beta(z_0 - r_\theta(x,y))), & \text{if } d = -1 \end{cases}$$

$$\lambda(d) = \begin{cases} \lambda_D, & \text{if } d = 1 \\ \lambda_U, & \text{if } d = -1 \end{cases}$$

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

$$z_0 = D_{\text{KL}}\left(\pi_\theta(y|x) \| \pi_{\text{ref}}(y|x)\right)$$

The three hyperparameters β is a temperature parameter, and λ_D, λ_U are scaling parameters for desirable and undesirable responses respectively. In the references implementation and experiments $\lambda_D = \lambda_U = 1$ and the reference model is a SFT model.

Relevance

Collecting the ranked preference data which is needed by normal alignment methods is normally quite expensive [4] where binary like dislike information is much more natural and less invasive to collect. KTO is shown to be more efficient and more robust to data imbalances than DPO. Furthermore a model of sufficient size (13B) does not need a SFT step prior to KTO which greatly reduces computational requirements. Furthermore KTO shows similar performance without using a reference model [3] which also halves the memory requirements when training.

2.2.6 Constitutional AI

What is Constitutional AI?

Constitutional AI (CAI) is a method of post-training a large language model to be aligned to set of principles that are outlined in simple natural language document called a constitution [2]. Importantly it works without the need for human labelling of data common in other alignment methods such as RLHF and DPO [7, 8]. Instead of human labelling CAI uses the model itself to generate feedback on its outputs based on the principles outlined in the constitution, this feedback is then used to further train the model to align it to the constitution.

[3] This is done by assuming that the reference model returns a uniform distribution across all outputs given x

How does Constitutional AI work?

CAI is a process that takes in a constitution, a pre-trained language model, and a dataset of prompts and outputs a model that is aligned to the constitution. The process consists of three main steps: self generative and supervised fine-tuning (SFT), preference modelling (PM), and reinforcement learning with human feedback (RLHF) [2]. The constitution itself was generated in an adhoc manner by the authors of [2] and consists of a set of principles that aim to make the model's outputs more helpful, honest, and harmless.

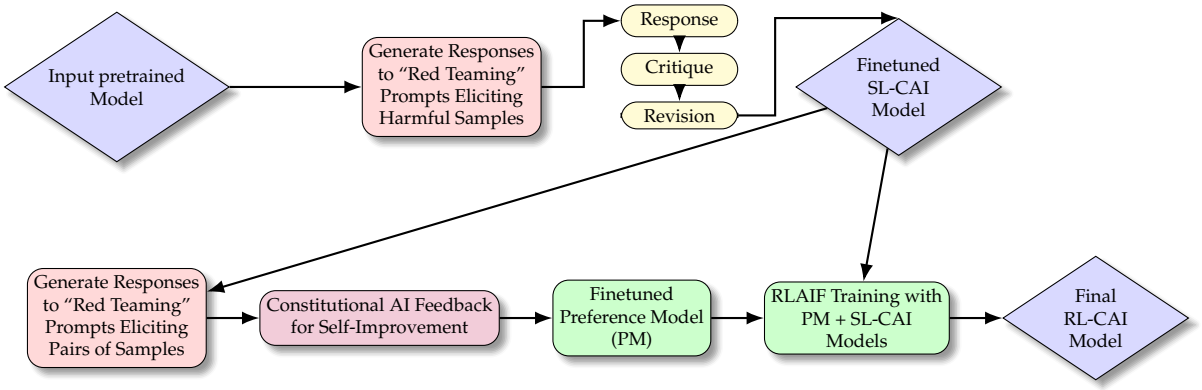

Figure 2.1: Overview of Constitutional AI process, adapted from [2]

Supervised learning on revised responses

The first stage of the process involves taking the pre-trained model and using it to generate responses to a set of prompts. These prompts are designed to elicit harmful behavior. As the pre-trained model is designed to be helpful it will likely generate harmful responses to these prompts. These responses are then critiqued by the model itself using principles outlined in the constitution, only a subset of which are used for each response critique. The critique is then used to revise the original response to make it more aligned to the constitution. This process of generating responses, critiquing them, and revising them is done iteratively to create a dataset of revised responses. This dataset is then used to fine-tune the pre-trained model using self-supervised learning to create a model called the SL-CAI model.

Reinforcement learning with AI feedback

The next step involves using the SL-CAI model to generate pairs of responses to a set of prompts (overlapping with the set of prompts from before or not). Then using the constitution the SL-CAI model is used to provide feedback on which of the two responses is better aligned to the constitution. This feedback is then used to fine-tune a preference model (PM) that can predict which of two responses is better aligned to the constitution. Finally the PM and SL-CAI models are used together to do reinforcement learning in style of RFHL [7] to generate the final model RL-CAI. This final model is now aligned to the constitution without any human labelling of data. In the setup used in [2] the authors did in fact use some labels from human labelers which were used to provide the helpfulness signal in the preference model training, however in principle the preference signal could have been entirely generated by an AI model.

Relevance

In the context of when this paper was released there was a key dilemma of helpfulness vs harmlessness in large language models. Therefore, this paper set out to make a model that is both helpful and harmless, without sacrificing one for the other. Further research has expanded the horizons to be concerned with more than just harmfulness vs helpfulness trade off. As hinted at by the authors in [2], there is room for improvement in the CAI method to make it focus on more than just harmlessness vs helpfulness. This can be done by modifying the constitutions to include more diverse principles as well as modifying the red teaming prompts to target more diverse scenarios (i.e controversial topics, political leanings, etc). In this way we can see how elements of CAI can be built upon by generating these diverse constitutions and red teaming prompts in a representative manner to create models aligned to a representative set of values.

2.2.7 Current technical alignment limitations

Introduce some of the ongoing concerns around current alignment methods. Such as shallow alignment, inner misalignment, scalable oversight etc.

2.3 Broader AI safety concerns

Take the problem of alignment and expand it broader AI alignment concerns. Talk about mis generalization, and specification gaming. Introduce the concepts from PublicAI, incentives and collective intelligence. ...

2.4 Representing Human Values

Discuss concepts such as how does one understand the values of a group. How does it relate to building AI systems. I need to significant reading on this as I am most outside my comfort zone for this. Look into some philosophical concepts around how the collective the best baseline if we have no other gold standard. Provide some rationale as to why using the public as the source of values is a good idea, as opposed to some "gold standard" values provided by experts.

Chapter 3

Project Proposal

Rather than write the proposal at the end I will write it down now to get my current ideas and can update it as I go along. I worry this could "lock in" my ideas too early, yet the benefits of the process of writing seems to outweigh that. Furthermore I will expand to make it more complete and detailed as I go along and figure out what the technical approach will be.

3.1 What I will do

The goal of this project is to develop, implement and evaluate for future feasibility a method for aligning large language models to a set of representatively collected human values. This will be a complete end to end process from data collection to final models output.

3.2 What problem I will solve

Current methods for training large language models involve scraping the internet for large amount of text data. This creates a very powerful next token prediction model. Then to turn this models into something useful for the end user (e.g. chatbots, assistants, robo doctor etc) the model is aligned to how we want it to behave. Current methods for alignment involve contracting human labelers/annotators/demonstrators to provide data to give to the model on how we want it to behave. Problematically this is a opaque process that gives the end user little insight into what the model is "designed to do". Instead by collecting a representative set of human values from the public we can align models to be in line with the end users themselves [1].

3.3 Timeline

The project will be carried out over three main phases. The first phase will be development of a data collection method to gather a representative set of human values. The second phase will be the implementation of an alignment method that uses the collected data to align a large language model. The third phase will be the evaluation of the aligned model against the human values dataset to determine the effectiveness of the alignment.

[1] In the sense that the public are the end users of these models.

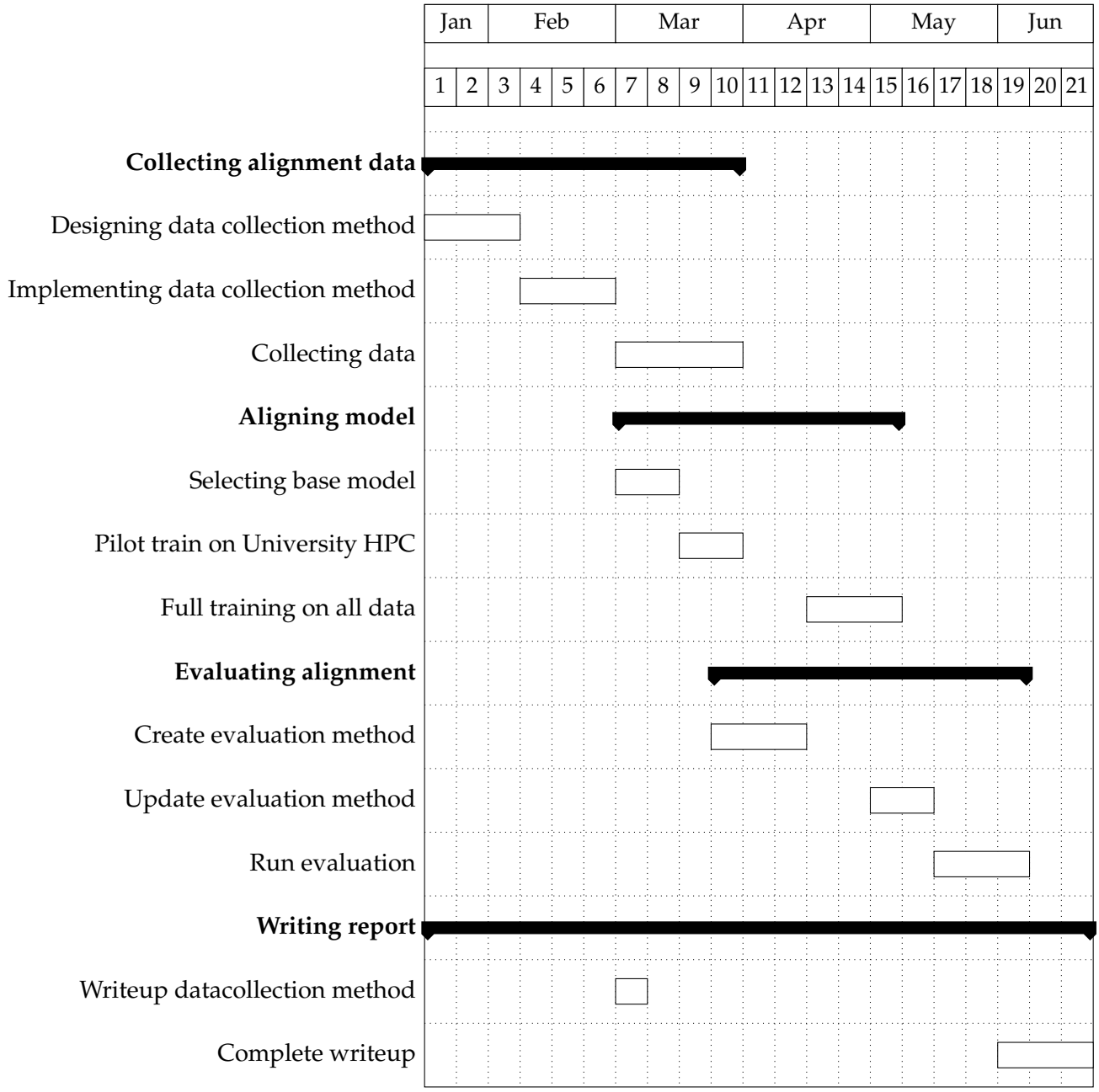

Figure 3.1: Project timeline in weeks (only roughly aligned to months). This assumes a start time of end of January 2026 with near full time work ramping up to full time work by end of February.

3.4 Output

The output is both qualitative and quantitative in nature. There are four main deliverables:

- A dataset of representatively collected human values for alignment of large language models.
- A reproducible method for collecting a representative set of human values.
- A aligned set of LLMs using the collected human values.

- A report that outlines the all previous outputs including evaluation of the aligned models.

3.5 Resources needed

There are two resource requirements for this project, hardware and survey responses.

3.5.1 Hardware

The first is hardware which will be needed to do the alignment finetuning and training of the model. This will be done on the Engineering and Computer Science department's GPU servers should give me access to a machine with up to 4 A40 GPUs. This will be sufficient to train models up to the size of about 30 billion parameters which is sufficient for *near* SOTA performance on many tasks. Alternatively I may rent GPU time on cloud providers such as Lambda AI which could cost in the realm of several hundred NZD depending on the total training time needed.

Survey Responses

No idea here. Need to research how many response would be needed. I imagine that I would want several thousands responses to get a sufficiently large data signal. Therefore at least a few hundred respondents to get meaningful "representation" and subgroups.

Bibliography

[1] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., and Zhu, T. Qwen Technical Report, Sept. 2023.

[2] Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukosuite, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., DasSarma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T., Hume, T., Bowman, S. R., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T., and Kaplan, J. Constitutional AI: Harmlessness from AI Feedback, Dec. 2022.

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners, July 2020.

[4] Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., Wang, T., Marks, S., Segerie, C.-R., Carroll, M., Peng, A., Christoffersen, P., Damani, M., Slocum, S., Anwar, U., Siththaranjan, A., Nadeau, M., Michaud, E. J., Pfau, J., Krasheninnikov, D., Chen, X., Langosco, L., Hase, P., Biyik, E., Dragan, A., Krueger, D., Sadigh, D., and Hadfield-Menell, D. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback, Sept. 2023.

[5] Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences, June 2017.

[6] Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. KTO: Model Alignment as Prospect Theoretic Optimization, Nov. 2024.

[7] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J.,

AND LOWE, R. Training language models to follow instructions with human feedback, Mar. 2022.

[8] RAFAILOV, R., SHARMA, A., MITCHELL, E., ERMON, S., MANNING, C. D., AND FINN, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, July 2024.

[9] SCHULMAN, J., WOLSKI, F., DHARIWAL, P., RADFORD, A., AND KLIMOV, O. Proximal Policy Optimization Algorithms, Aug. 2017.

[10] SHAO, Z., WANG, P., ZHU, Q., XU, R., SONG, J., BI, X., ZHANG, H., ZHANG, M., LI, Y. K., WU, Y., AND GUO, D. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, Apr. 2024.

[11] TEAM, G., ANIL, R., BORGEAUD, S., ALAYRAC, J.-B., YU, J., SORICUT, R., SCHALKWYK, J., DAI, A. M., HAUTH, A., MILLICAN, K., ET AL. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).

[12] TVERSKY, A., AND KAHNEMAN, D. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty 5*, 4 (Oct. 1992), 297–323.