
Inverse Constitutional AI: Compressing Preferences into Principles

Arduin Findeis
University of Cambridge
af691@cam.ac.uk

Timo Kaufmann
LMU Munich, MCML Munich
timo.kaufmann@ifi.lmu.de

Eyke Hüllermeier
LMU Munich, MCML Munich
eyke@ifi.lmu.de

Samuel Albanie
samuel.albanie.academic@gmail.com

Robert Mullins
University of Cambridge
robert.mullins@cl.cam.ac.uk

Abstract

Feedback data plays an important role in fine-tuning and evaluating state-of-the-art AI models. Often pairwise text preferences are used: given two texts, human (or AI) annotators select the “better” one. Such feedback data is widely used to align models to human preferences (e.g., *reinforcement learning from human feedback*), or to rank models according to human preferences (e.g., *Chatbot Arena*). Despite its wide-spread use, prior work has demonstrated that human-annotated pairwise text preference data often exhibits unintended biases. For example, human annotators have been shown to prefer *assertive* over *truthful* texts in certain contexts. Models trained or evaluated on this data may implicitly encode these biases in a manner hard to identify. In this paper, we formulate the interpretation of existing pairwise text preference data as a compression task: the *Inverse Constitutional AI* (ICAI) problem. In constitutional AI, a set of principles (or *constitution*) is used to provide feedback and fine-tune AI models. The ICAI problem inverts this process: given a dataset of feedback, we aim to extract a *constitution* that best enables a *large language model* (LLM) to reconstruct the original annotations. We propose a corresponding initial ICAI algorithm and validate its generated constitutions quantitatively based on reconstructed annotations. Generated constitutions have many potential use-cases—they may help identify undesirable biases, scale feedback to unseen data or assist with adapting LLMs to individual user preferences. We demonstrate our approach on a variety of datasets: (a) synthetic feedback datasets with known underlying principles; (b) the AlpacaEval dataset of cross-annotated human feedback; and (c) the crowdsourced Chatbot Arena data set. We release the code for our algorithm and experiments at <https://github.com/rdnfn/icai>.

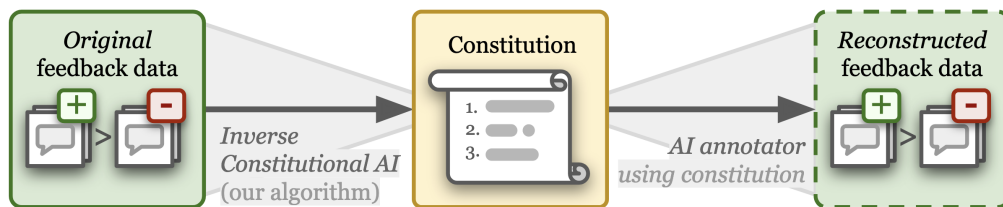


Figure 1: **The Inverse Constitutional AI problem.** Starting from a set of pairwise preference feedback, we derive a set of natural language principles (a *constitution*) that explain the preference data. For validation, we re-construct the original preferences with an LLM judging according to the generated constitution. The constitution represents a (highly compact) compression of the preferences.

1 Introduction

State-of-the-art *large language models* (LLMs) rely heavily on human feedback for training and evaluation. This feedback, often in the form of *pairwise text preferences*, is crucial to assess advanced capabilities hard to evaluate automatically. Strategies for training on such data have seen widespread adoption, with notable examples including *reinforcement learning from human feedback* (RLHF) [Ouyang et al., 2022, Stiennon et al., 2020] and *direct preference optimization* (DPO) [Rafailov et al., 2023]. Beyond training, pairwise text preferences are also used for evaluating LLMs, such as in the *Chatbot Arena* [Chiang et al., 2024], where crowdsourced preferences determine rankings. Here, users interact with two anonymous models simultaneously via a web interface and select the preferred output. The resulting ranking may offer a better estimation of real-world human preferences than conventional static benchmarks, since it captures the multi-faceted nature of human preferences [Xu et al., 2023].

Unfortunately, human feedback is not without its flaws. Systematic biases in human judgement have been documented extensively in the psychology literature [Tversky and Kahneman, 1974]. It is therefore unsurprising that the human feedback used to guide and evaluate LLMs exhibits biases as well [Hosking et al., 2024, Wu and Aji, 2023, Bansal et al., 2024, Sharma et al., 2023, Xu et al., 2023]. For example, Hosking et al. [2024] find that human annotators can sometimes favour assertiveness over truthfulness. Wu and Aji [2023] observe that annotators sometimes place more importance on grammatical errors than truthfulness. Feedback data with unintended biases can be problematic: when used for fine-tuning, biased data may lead to models that exhibit the same biases. Similarly, leaderboards based on biased data will *favour misaligned models* [Dubois et al., 2023, 2024]. As such, it is valuable to understand the implicit rules and biases guiding annotators of feedback data. To date, however, few tools exist to detect biases in pre-existing preference data at scale. Prior work usually builds on specially designed datasets to observe biases and cannot be directly applied to pre-existing data or data generated in less controlled settings.

In this paper, we propose a novel approach to understanding preference corpora: *Inverse Constitutional AI* (ICAI). We make the following contributions:

1. **Formulating the *Inverse Constitutional AI* (ICAI) problem.** In *Constitutional AI* [Bai et al., 2022], a set of principles (or *constitution*) is used to provide feedback and fine-tune language models. The ICAI problem inverts this process: given a dataset of feedback by a human or model, we seek to compress the annotations into a set of principles that enable *reconstruction* of the annotations (Fig. 1).
2. **Proposing an initial ICAI algorithm.** We introduce a first ICAI algorithm that generates a set of principles based on a feedback dataset. We validate the constitutions generated by our algorithm based on their ability to help reconstruct feedback. Given the complexity of human judgement, the constitution necessarily represents a “lossy”, non-unique compression of the feedback data. Nevertheless, the interpretable nature of the principles may enable a number of promising downstream use-cases: (a) highlighting potential issues in preference data; (b) creating interpretable reward models; (c) scaling human-annotated evaluation to new models and use-cases; and (d) generating personal constitutions for customized model behavior.
3. **Providing extensive experimental results.** We test our approach experimentally on three different datasets: (a) we first provide a proof-of-concept on *synthetic data* with known underlying principles; (b) we then demonstrate applicability to human-annotated data on the *AlpacaEval dataset* [Dubois et al., 2023]; and (c) finally we showcase applicability to individual users via crowdsourced *Chatbot Arena Conversations* data [Zheng et al., 2023]. On the latter dataset we investigate individualized preferences, an area attracting growing interest [Kirk et al., 2024]. We demonstrate highly sample-efficient generation of individual constitutions with human-readable principles that can be edited and adapted. We release the code to reproduce our results publicly.¹

2 The Inverse Constitutional AI Problem

We describe the *Inverse Constitutional AI* (ICAI) problem as follows: given a set of pairwise preference feedback, the goal is to generate a corresponding *constitution* of natural-language principles that

¹<https://github.com/rdnfn/icai>

Inverse Constitutional AI *algorithm*

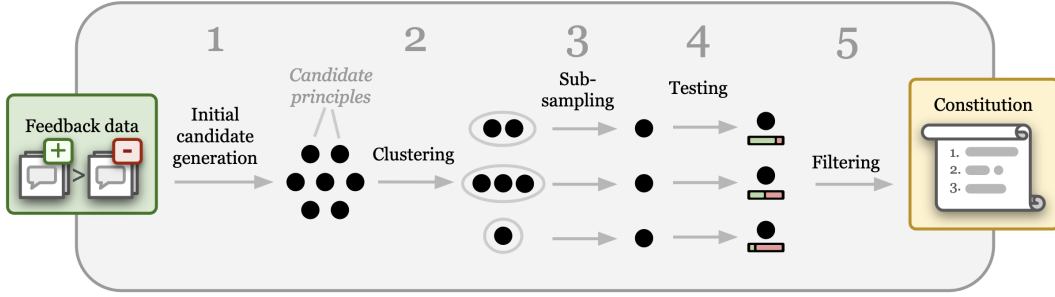


Figure 2: **Overview of our *Inverse Constitutional AI (ICAI) algorithm*.** Given a dataset of pairwise rankings, in Step 1 candidate principles are *generated* using an LLM. In Step 2, these principles are *clustered* using an embedding model. In Step 3, similar principles are “de-duplicated” by *sampling* one principle per cluster. In Step 4, each principle is tested to evaluate its ability to help an LLM reconstruct the original annotations. Finally in Step 5, the principles are *filtered* according to the testing results and a set of filtered principles are returned as the final *constitution*. Optionally, this last step is augmented with additional clustering and subsampling steps to ensure diverse principles.

enable an LLM annotator to reconstruct the original preferences as well as possible. More formally, we consider the following optimization problem:

$$\operatorname{argmax}_c \{ \text{agreement}(p_o, p(c)) \text{ s.t. } |c| \leq n \}, \quad (1)$$

where p_o are the original preferences and $p(c)$ are *constitutional* preferences over a pairwise text corpus T , generated by LLM M using the *constitution*, c . The constitution is subject to the constraint of consisting of up to n natural language principles that are human-readable. Agreement is defined as the percentage of constitutional preferences $p(c)$ that are identical to the original preferences p_o (i.e. select the same output in a pair from corpus T). A constitution with high agreement can help interpret a preference dataset to gain insight into the underlying biases and preferences of human or AI annotators. The constitution may also be used for future preference synthesis, with an interpretable and editable set of principles.

3 Method

Our proposed first *Inverse Constitutional AI (ICAI) algorithm*, outlined in Fig. 2, consists of five main steps: *principle generation*, *principle clustering*, *principle sub-sampling*, *principle testing*, and *principle filtering*. In the following, we will describe each step in detail.

Step 1: Principle generation. We extract candidate principles using an LLM with access to the feedback data. The principles are generated on a per-comparison basis: an LLM is prompted with a pair of texts and corresponding preference, and then asked to propose principles that explain the preference (see Appendix B.1 for prompts). The generated principles are in the form of natural language instructions that annotators can use to make preference decisions (e.g., “select the more polite output”). We generate a large number of candidate principles to ensure that we cover a wide range of potential rules. To ensure principle diversity, we further use multiple passes (by default 2) over the preferences with different generation prompts. The generation prompts affect the type of principles that get generated and tested (e.g., whether specific/general, positive/negative rules). Depending on the use-case, different generation prompts may be helpful.

Step 2: Principle clustering. Since the first step generates a large number of candidate principles independently, almost identical principles may be generated multiple times. We then use K-means-based clustering to identify principles that are similar and can be merged into a single principle. The parameter k determines the number principles considered downstream and thus affects overall computational cost.

Step 3: Principle sub-sampling. In the third step, we de-duplicate the principles by randomly sampling one principle per cluster, leading to a diverse set of remaining principles.

Step 4: Principle testing. The fourth step evaluates the generated principles’ ability to help an LLM reconstruct the original annotations. We prompt the LLM with the generated principles to determine their support for each pair (see Appendix B.2). We parallelize this step, prompting the LLM with a pair of texts and multiple principles to provide a separate response (first preferred, second preferred, not relevant) for each principle. This parallelization reduces the tokens requirements compared to separate testing. While LLMs can exhibit anchoring effects when predicting multiple labels in one output [Stureborg et al., 2024], we hypothesize this effect is less pronounced for relative preferences and our experimental results indicate sufficient reliability on our datasets. We compare these results to the original labels and count the correct, incorrect, and not relevant labels for each principle separately, thereby identifying principles that help the LLM to correctly annotate the dataset.

Step 5: Principle filtering. Finally, the principles are filtered based on the results of the previous testing step. We only keep principles that improve the reconstruction loss, while discarding principles that do not help or even hinder the reconstruction. We further discard principles that are marked as relevant on less than $x\%$ of the data (default 10%), to avoid overly specific principles that do not generalize. We order the principles according to their net contribution to correctly annotating the dataset ($\# \text{ correct} - \# \text{ incorrect annotations}$). We then select the top n principles² according to this order. Optionally, to increase principle diversity, we cluster the top m ($> n$) principles into n clusters as before, and sub-sample the highest ordered principle from each cluster.³ The final ordered list of principles from this filtering step are returned as the *constitution*.

Inference. Given a constitution, we can validate its ability to “explain” the original feedback dataset. We do this validation using AI annotators prompted with the constitution, an approach pioneered by Bai et al. [2022] and commonly referred to as constitutional AI. Notably this leaves room for interpretation of the constitution by the AI annotator, as the constitution may be ambiguous, contradictory or incomplete. It is also dependent on the exact phrasing of the prompt and the constitution, an effect extensively studied by Li et al. [2024b] on whose work we build. When seen as an explainability tool, the possibility of conducting inference with the generated constitution is a key benefit of our approach, as it allows for validation of the explanation on a test set. This constitutional AI annotator-based inference enables us to quantitatively test the validity of our generated constitutions and their ability to explain the data while also enabling downstream use-cases such as personalized preference models.

4 Experiments

We conduct experiments on three datasets: (1) *synthetic data* to demonstrate the basic functionality of our algorithm, (2) human-annotated *AlpacaEval data* to demonstrate the applicability of our algorithm to real-world data, and (3) *Chatbot Arena data* to illustrate the application of our algorithm to infer individual user preferences. Throughout our experiments we primarily use two models from OpenAI: gpt-3.5-turbo-0125 (referred to as *GPT-3.5-Turbo*) and gpt-4o-2024-05-13 (referred to as *GPT-4o*).⁴ We include the full numerical results in Appendix D.

Annotators. We use the AlpacaEval⁵ [Li et al., 2024b] package and their annotators as the baselines (see Appendix B.4). These annotators have been shown to strongly correlate with human preferences. To evaluate constitution effectiveness, we create further custom prompts that ask the model to annotate according to the principles in the constitution (see prompts in Appendix B.3). All our annotator outputs are parsed using functional APIs (where supported), as prior AlpacaEval experiments show this improves annotator performance. We show random baseline performance as a grey dashed line at 50% in all plots.

²Experimental results for varying n values available in Appendix A. The parameter n may be seen as determining regularisation in our algorithm: a small n may be considered highly regularised, limiting the amount of overfitting to the data possible. A large n enables including more fine-grained principles that only apply to smaller subset of examples. Note that, depending on the use-case, overfitting to the training data is not necessarily a problem (e.g., for data interpretability).

³During our experiments we found that it is important not to be too restrictive with the number of clusters in Step 2, as good principles may never be tested. By instead filtering again for duplicates after testing, we can increase the probability of the best candidate principle passing through our algorithm (at a slightly higher testing token cost).

⁴See this page for a detailed model description: <https://platform.openai.com/docs/models/>

⁵https://github.com/tatsu-lab/alpaca_eval, license: Apache-2.0

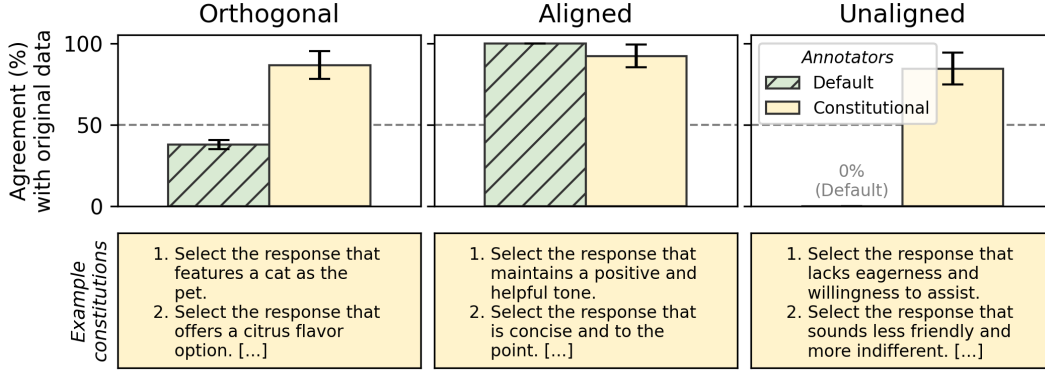


Figure 3: **Results on Synthetic Data. Using limited data and without fine-tuning, our constitutional annotators can reconstruct various kinds of preferences.** We demonstrate our algorithm’s adaptability on three synthetic datasets (30 preferences each): one *orthogonal* to the base LLM’s learned preferences, one *aligned* with those preferences and one *unaligned* with them, flipping the aligned dataset. We generate constitutions for each and report agreement with the original annotations of a *default* LLM annotator (prompted to select the “best” output) and a *constitutional* annotator (prompted with a constitution). Our constitutions notably improve agreement in the orthogonal and unaligned cases and retain high agreement in the aligned case, albeit with more variance. Our method’s ability to detect biases is illustrated by the example constitution in the unaligned case. Examples of generated constitutions were chosen for illustrative purposes. Experiments use GPT-3.5-Turbo, values and error bars report mean and standard deviation over six random seeds.

4.1 Synthetic data as a proof of concept

We first apply our algorithm to three *synthetic datasets* created according to known rules intended to be aligned, unaligned, and orthogonal to the base LLM’s preferences. We provide an overview of the dataset here, with further details in Appendix F.

Orthogonal. This dataset is based on preferences according to *orthogonal* principles, intended to be on average neither supported nor opposed by humans or language models. In particular, we create a dataset based on three principles: “prefer cats over dogs”, “prefer green over blue color”, and “select lemon over raspberry ice-cream”. We generate 10 pairs per principle, overall 30 pairs.

Aligned. This dataset uses preferences generally accepted by humans and (especially) language models. Our dataset follows three principles: “select truthful over factually incorrect answers”, “select helpful over useless answers”, “select polite over impolite answers”. We again generate 10 pairs per principle, overall 30 pairs.

Unaligned. This dataset flips the aligned dataset’s annotations, creating a dataset a default LLM annotator mostly disagrees with.

Figure 3 shows our results on synthetic data. We compare *default* annotators (prompted to select the “best” model) to *constitutional* annotators (prompted with a generated constitution). LLM annotators with generated constitutions reconstruct original annotations better in the orthogonal and unaligned datasets, and keep high agreement in the aligned case. These results indicate the constitutions capture helpful information about the preferences. Qualitatively, the generated constitutions⁶ often closely correspond to the principles described above.

4.2 Human-annotated AlpacaEval data

Next, we test our approach on human-annotated texts, using the *AlpacaEval dataset*⁷ [Dubois et al., 2023]. The dataset, used for the *AlpacaEval leaderboard*, contains about 650 data points cross-annotated by four annotators, with well-tested baseline AI annotators and evaluation tooling.

⁶The best, median and worst constitutions are available in full in Appendix C for all relevant experiments.

⁷https://huggingface.co/datasets/tatsu-lab/alpaca_eval, license: CC-BY-NC-4.0

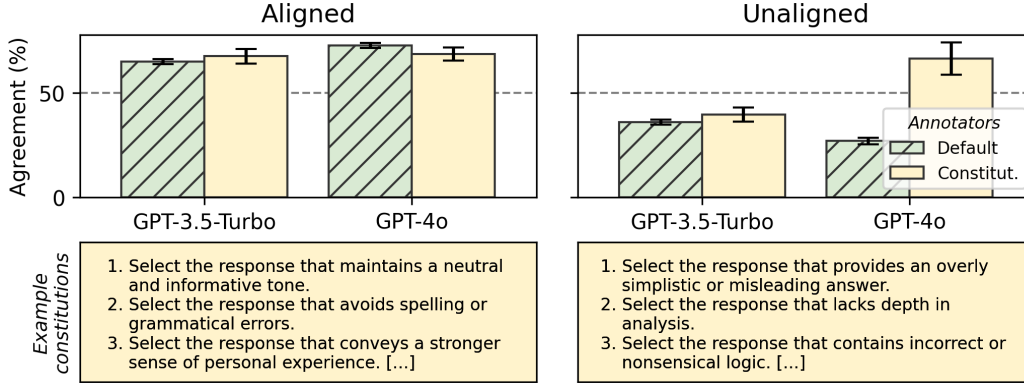


Figure 4: **Results on AlpacaEval data. GPT-4o is capable of generating and using constitutions that notably improve agreement with unaligned preferences.** We sample 65 training and 65 test samples from AlpacaEval’s human-annotated dataset, considering both aligned (original) and unaligned (flipped annotations) versions. GPT-4o generates constitutions for each version on the training set, evaluated on the test set using GPT-3.5-Turbo and GPT-4o annotators. In the aligned case, GPT-3.5-Turbo slightly increases its average agreement using a constitution while GPT-4o decreases its average agreement. This may be because GPT-4o is more aligned with human annotations than our average compressed constitution, while GPT-3.5-Turbo is less aligned and benefits from the constitution. In the unaligned case, GPT-4o’s agreement improves notably, while GPT-3.5-Turbo’s performance does not exceed random choice, indicating its limited ability to follow unaligned principles. Reported values and error bars are the mean and standard deviation over 6 random seeds. Example constitutions were picked for illustrative purposes.

Experimental setup. We randomly select two mutually exclusive training and test subsets, each with 65 annotated pairs (overall $\sim 20\%$ of the dataset). Similar to the synthetic experiments, we derive an *aligned* and *unaligned* dataset. In the aligned version the annotations are based on the majority vote of the original 4 cross-annotations (ties broken randomly), while the annotations are flipped in the unaligned scenario, so that most humans (and LLMs) would disagree with them.

Results. Figure 4 shows the results in aligned and unaligned scenarios. Constitutions are generated on the training subset and results reported on the (unseen) test subset. In general, we observe that the improvement with constitutions are limited in the aligned scenario but more noticeable in the unaligned scenario. There appears to be a capability gap between GPT-3.5-Turbo and GPT-4o in the unaligned case: whereas the former does not improve beyond random with a constitution, the latter is able to notably outperform the default annotator and performs well above random 50% agreement.

Finally, given GPT-3.5-Turbo’s limitations in the unaligned case in Fig. 4, we explore more generally how well our constitutions *can transfer between models*. To this end, we take the best performing constitution of the unaligned case on the training set and test how well models from Anthropic’s Claude family are able to use these constitutions on the test set. Shown in Fig. 5, the results indicate that this constitution transfers well to the Claude models—better than to GPT-3.5-Turbo.

4.3 Individual preferences: Chatbot Arena data

We test our approach’s ability to generate *personal constitutions* for individual users using the *Chatbot Arena Conversations* dataset⁸ [Zheng et al., 2023], which consists of 33k human-annotated preferences filtered by the authors to avoid personally identifiable information (PII). Unlike AlpacaEval, the dataset includes user-generated prompts in addition to user-generated preferences, which may be considered more expressive of user preferences than annotations on pre-existing prompts.

Experimental setup. We select two users exhibiting different preferences from the Chatbot Arena dataset (with 9 and 12 preference annotations respectively) and generate a separate constitution with

⁸https://huggingface.co/datasets/lmsys/chatbot_arena_conversations, license: CC-BY-NC-4.0

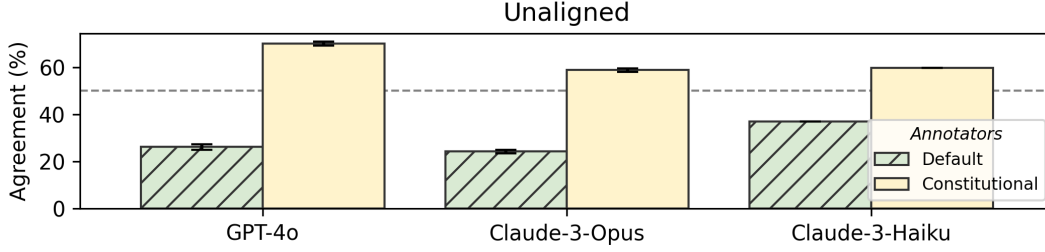


Figure 5: **Transferability of constitutions: results of transferring a GPT-4o generated constitution to other model family (Claude).** We use the highest-performing unaligned constitution on the training set, from experiments shown in the unaligned plot in Figure 4. We test two additional models from the Claude model family, Claude-3-Opus and Claude-3-Haiku. Both are able to use GPT-4o’s generated constitution to reconstruct the test set annotations effectively, albeit to a lower standard than GPT-4o. The reported values and error bars are the mean and standard deviation over 4 random seeds per annotator, all with the same constitution.

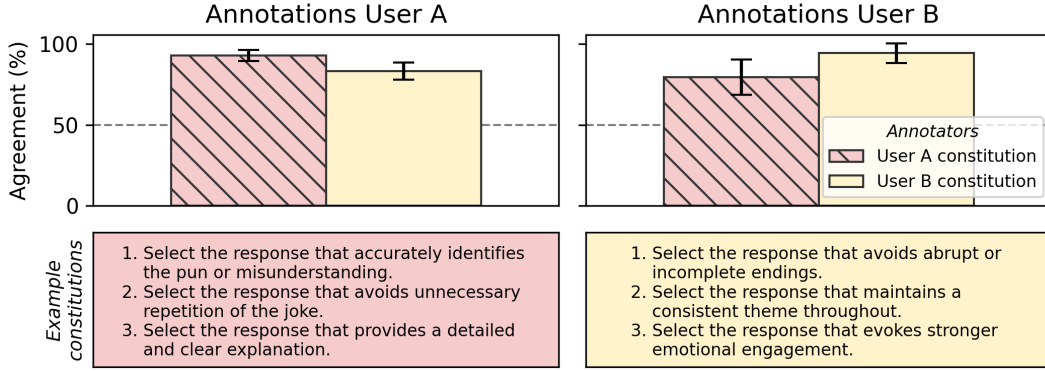


Figure 6: **Case-study of generating personal constitutions for anonymous Chatbot Arena users.** Personal constitutions have the potential to help make LLM applications more helpful and customized to individual users’ preferences—in an interpretable way. We generate constitutions based on a single user’s annotations and check the constitutions’ ability to help re-construct the annotations of the same user and another user. For the selected users, we observe that generated personal constitutions appear to work best for original user and do not transfer perfectly to another user. Note that this effect will vary for other users depending on how different users’ preferences are. Examples of generated constitutions were chosen for illustrative purposes. Reported values and error bars are the mean and standard deviation over 6 random seeds of the entire generation-annotation pipeline, run with GPT-4o.

3 principles for each. To be better able to detect the effect of user-specific principles, we adapt our generation and annotation prompts to be less general, i.e. to generate constitutions more specific to the individual users and to follow the specific principles more closely rather than the model’s priors.

Results. Figure 6 shows the results of our experiments: the personal constitutions of each user work best for the user they were created for. Note that how well a personal constitution transfers from one user to another depends on how similar their preferences are. Constitutions of similar users may transfer well, whilst the contrast will be most pronounced between users with opposing preferences. Thus, the results here illustrate the use of personal constitutions, but may vary for any given pair of users.

5 Related work

Our work focuses on deriving interpretable principles from human feedback data and using AI annotators to evaluate those principles. We build on work related to learning from human feedback, biases in feedback, interpretable preference models, prompt generation, and AI annotators.

Learning from human feedback. Fine-tuning LLMs with human feedback has significantly contributed to the success of modern LLMs [Ouyang et al., 2022, Stiennon et al., 2020]. Typically, feedback is collected through pairwise comparisons of model outputs, training a *reward model* for fine-tuning, e.g. using reinforcement learning from human feedback (RLHF) [Ouyang et al., 2022, Stiennon et al., 2020, Kaufmann et al., 2024] or direct preference optimization (DPO) [Rafailov et al., 2023]. Interpreting preference data is challenging since it generally lacks annotations of the underlying reasons. Additionally, the reward model is often a black-box neural network, making it hard to interpret as well. Our work aims to generate interpretable principles explaining the feedback data.

Biases in human feedback. Identifying biases in human feedback data is crucial, as unintended biases are common [Hosking et al., 2024, Wu and Aji, 2023, Bansal et al., 2024, Sharma et al., 2023, Xu et al., 2023]. For example, Hosking et al. [2024] note a preference for assertiveness over truthfulness, while Wu and Aji [2023] highlight a bias towards grammatical correctness over factual accuracy. Bansal et al. [2024] highlight that feedback methods influence biases, e.g., annotators focus more on accuracy in pairwise ranking compared to rating feedback. Additionally, Sharma et al. [2023] observe a bias towards sycophantic outputs, where responses align with the user’s beliefs rather than the truth. While these studies provide valuable insights, most methods for detecting biases rely on specialized feedback data collection, making them challenging to apply to pre-existing data. Our work generates interpretable principles from existing preference data, which can be inspected to detect biases and provide insights into the underlying preferences.

Interpretable preference models. There has been a growing interest in creating interpretable preference models, aiding in understanding behavior of AI systems. Go et al. [2024] create a *compositional* preference model based on 13 fixed features, similar to our constitutional principles. While they do not generate the constitution from data, they do create a regression model to weight them, which would be a promising extension to our approach. Petridis et al. [2024] propose a feedback-based constitution generation method relying on interactive tools, whereas our approach can be applied to standard preference datasets.

Prompt generation. LLM outputs can be guided by generating specific prompts. This relates closely to our work, where we create principles to steer outputs. Adversarial prompt generation aims to produce outputs contrary to the model’s training data, similar to our unaligned setting. Manual adversarial prompt generation, or ‘jailbreaking’, allows users to bypass safety constraints imposed during fine-tuning. This process can also be automated [Zou et al., 2023], generating adversarial prompts to attack a wide range of models. Li et al. [2024a] propose virtual tokens to steer outputs towards specific viewpoints, using a dataset of question responses to define these personas, unlike our approach based on pairwise comparisons and interpretable constitutions. Rodriguez et al. [2024] explore the use of LLMs to discover and classify user intent, which may help adapt model prompts.

AI annotators. Due to the cost and time required for human feedback collection, AI annotators, or LLM-as-a-judge, have been proposed as a scalable alternative. Constitutional AI [Bai et al., 2022] uses LLMs with a set of principles for feedback. Due to strong alignment with human preferences through fine-tuning, LLMs can generalize from very general principles, such as “do what’s best for humanity” [Kundu et al., 2023], or even give feedback well-aligned with human preferences without any constitution [Zheng et al., 2023]. Our experiments show similar trends, where default LLM annotators align well with dataset annotations, even without a constitution. AI annotators can exhibit biases, partially overlapping with human biases [Chen et al., 2024], and inconsistencies in their judgments [Stureborg et al., 2024]. Examples include position bias (preferring the first output) [Zheng et al., 2023], verbosity bias (preferring longer responses) [Zheng et al., 2023], and self-enhancement or familiarity bias (preferring outputs similar to their own) [Panickssery et al., 2024, Stureborg et al., 2024]. Their proposed mitigation measures include trying both orderings and tying if inconsistent, with further explorations in later work [Dubois et al., 2024]. AlpacaEval [Li et al., 2024b] offers a set of well-validated AI annotators. Our work both builds upon and extends this line of research,

proposing a new method to automatically generate an inspectable and modifiable constitution and using an LLM-as-a-judge to guide an LLM towards this constitution.

Rule-based preference learning. Our work involves rule learning [Fürnkranz et al., 2012], aiming to develop descriptive or predictive rules, a method previously applied to preference learning [de Sá et al., 2011]. A common technique for rule learning is to first generate a set of candidate rules and then measure each rule’s *support* in a dataset, i.e. the fraction of data points that satisfy the rule [Fürnkranz et al., 2012, de Sá et al., 2011]. Our algorithm follows this approach but, in contrast to more traditional rule learning, generates rules as natural language sentences. These rules, though more ambiguous and requiring AI annotators for interpretation, are expressive, interpretable, and easy for non-experts to edit.

6 Limitations and discussion

It is important to consider the limitations of our approach when interpreting our results. Firstly, *we do not show causality*—our generated principles correlate LLM annotators with the original annotations, but we cannot validate if these principles were used by the original annotators. Multiple constitutions may explain the data equally well (as in the Rashomon effect [Breiman, 2001]). Nonetheless, an undesirable principle correlating with annotations is concerning, even if the principle was not intentionally used. For example, in the ‘aligned’ variant of the AlpacaEval dataset, some generated constitutions include principles to prefer verbose or redundant responses (see Appendix C.2.1). While this principle was likely not consciously followed by the original annotators, its high support in the dataset may warrant further investigation and possible data cleaning. Secondly, *constitutions represent a lossy compression*—A constitution of a few principles is a simplification of the decision-making process underlying annotations. Some annotations may not be possible to reconstruct based on a simple constitution.⁹ Finally, *preferences closely aligned to LLMs are challenging to test*. If an LLM annotator is already highly aligned with the dataset annotations, improving its performance with a constitution is challenging. The constitutional reconstruction loss is most useful for evaluating principles orthogonal to or against the popular opinions internalized by the LLM. On already well-aligned models, the constitution may not improve performance, but it can still provide insights into the underlying preferences.

We hope our work will have a positive societal impact by helping better understand preference data, already widely used for fine-tuning and evaluation of popular LLMs. We emphasize that our generated constitutions cannot claim to re-construct an individual’s true reasoning process. The constitutions should be interpreted cautiously when working with human annotators to avoid potential negative implications. Similarly, the use of our approach for personalised LLMs should also be considered carefully. Overall, we believe the potential for positive impacts outweighs possible negative impacts.

Future work should focus on addressing these limitations, extending the capabilities of our approach, possibly using multi-modal models, and exploring new applications.

7 Conclusion

We have presented our work on the *Inverse Constitutional AI* (ICAI) problem: first defining the ICAI problem compressing preference data into a short list of natural language principles (or *constitution*). We then introduced an initial ICAI algorithm as a first approach to generate such constitutions. We demonstrated the effectiveness of our approach in experiments across three different types of datasets: (a) *synthetic data* to provide a proof of concept; (b) *AlpacaEval data* to show the applicability to compress human-annotated data and the possibility of transferring constitutions across model families; and (c) *Chatbot Arena data* to illustrate the generation of personal constitutions. We hope that our approach can improve both our understanding and the usefulness of widely-used feedback data. Potential use-cases of our interpretable and editable constitutions include: highlighting issues with datasets, creating interpretable alternatives to black-box reward models, scaling human-annotated evaluation to new models and use-cases, and improving model customization via personal constitutions. We are excited for future work to explore these use-cases in more detail.

⁹Similarly, a country’s legal constitution is an ambiguous set of principles interpreted by judges, or LLMs in our case.

Acknowledgments and Disclosure of Funding

We thank Benjamin Minixhofer, Max Bartolo, Tom Hosking, Hritik Bansal, the RECOG-AI team at the Leverhulme Centre for the Future of Intelligence (especially José Hernández-Orallo and Lorenzo Pacchiardi), Bill Marino and Jason Brown for their valuable feedback and discussions on early versions of this work. We also thank Anthropic for providing free research access to their models. Arduin Findeis was supported by a University of Cambridge School of Physical Sciences Award and by the UKRI Centre for Doctoral Training in Application of Artificial Intelligence to the study of Environmental Risks (reference EP/S022961/1). This publication was further supported by LMUexcellent, funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder as well as by the Hightech Agenda Bavaria.

References

- 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, December 2022. URL <http://arxiv.org/abs/2212.08073>. arXiv:2212.08073 [cs].
- Hritik Bansal, John Dang, and Aditya Grover. Peering Through Preferences: Unraveling Feedback Acquisition for Aligning Large Language Models, February 2024. URL <http://arxiv.org/abs/2308.15812>. arXiv:2308.15812 [cs] version: 3.
- Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001. URL <https://projecteuclid.org/journals/statistical-science/volume-16/issue-3/Statistical-Modeling--The-Two-Cultures-with-comments-and-a/10.1214/ss/1009213726.short>. Publisher: Institute of Mathematical Statistics.
- Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the Judge? A Study on Judgement Biases, April 2024. URL <http://arxiv.org/abs/2402.10669>. arXiv:2402.10669 [cs].
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference, March 2024. URL <http://arxiv.org/abs/2403.04132>. arXiv:2403.04132 [cs].
- Cláudio Rebelo de Sá, Carlos Soares, Alípio Mário Jorge, Paulo Azevedo, and Joaquim Costa. Mining Association Rules for Label Ranking. In *Advances in Knowledge Discovery and Data Mining*. Springer, 2011. doi: 10.1007/978-3-642-20847-8_36.
- Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback, August 2023. URL <http://arxiv.org/abs/2305.14387>. arXiv:2305.14387 [cs].
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators, April 2024. URL <http://arxiv.org/abs/2404.04475>. arXiv:2404.04475 [cs, stat] version: 1.
- Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of Rule Learning*. Springer, 2012. doi: 10.1007/978-3-540-75197-7.
- Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, and Marc Dymetman. Compositional preference models for aligning LMs, March 2024. URL <http://arxiv.org/abs/2310.13011>. arXiv:2310.13011 [cs].
- Tom Hosking, Phil Blunsom, and Max Bartolo. Human Feedback is not Gold Standard, January 2024. URL <http://arxiv.org/abs/2309.16349>. arXiv:2309.16349 [cs].

- Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A Survey of Reinforcement Learning from Human Feedback, April 2024. URL <http://arxiv.org/abs/2312.14925>. arXiv:2312.14925 [cs].
- Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The PRISM Alignment Project: What Participatory, Representative and Individualised Human Feedback Reveals About the Subjective and Multicultural Alignment of Large Language Models, April 2024. URL <http://arxiv.org/abs/2404.16019>. arXiv:2404.16019 [cs].
- Sandipan Kundu, Yuntao Bai, Saurav Kadavath, Amanda Askell, Andrew Callahan, Anna Chen, Anna Goldie, Avital Balwit, Azalia Mirhoseini, Brayden McLean, Catherine Olsson, Cassie Evraets, Eli Tran-Johnson, Esin Durmus, Ethan Perez, Jackson Kernion, Jamie Kerr, Kamal Ndousse, Karina Nguyen, Nelson Elhage, Newton Cheng, Nicholas Schiefer, Nova DasSarma, Oliver Rausch, Robin Larson, Shannon Yang, Shauna Kravec, Timothy Telleen-Lawton, Thomas I. Liao, Tom Henighan, Tristan Hume, Zac Hatfield-Dodds, Sören Mindermann, Nicholas Joseph, Sam McCandlish, and Jared Kaplan. Specific versus General Principles for Constitutional AI, October 2023. URL <http://arxiv.org/abs/2310.13798>. arXiv:2310.13798 [cs].
- Junyi Li, Ninareh Mehrabi, Charith Peris, Palash Goyal, Kai-Wei Chang, Aram Galstyan, Richard Zemel, and Rahul Gupta. On the steerability of large language models toward data-driven personas, April 2024a. URL <http://arxiv.org/abs/2311.04978>. arXiv:2311.04978 [cs].
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An Automatic Evaluator of Instruction-following Models, May 2024b. URL https://github.com/tatsu-lab/alpaca_eval. original-date: 2023-05-25T09:35:28Z.
- 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, March 2022. URL <http://arxiv.org/abs/2203.02155>. arXiv:2203.02155 [cs].
- Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM Evaluators Recognize and Favor Their Own Generations, April 2024. URL <http://arxiv.org/abs/2404.13076>. arXiv:2404.13076 [cs].
- Savvas Petridis, Benjamin D Wedin, James Wexler, Mahima Pushkarna, Aaron Donsbach, Nitesh Goyal, Carrie J Cai, and Michael Terry. ConstitutionMaker: Interactively Critiquing Large Language Models by Converting Feedback into Principles. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*. Association for Computing Machinery, 2024. doi: 10.1145/3640543.3645144.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, December 2023. URL <http://arxiv.org/abs/2305.18290>. arXiv:2305.18290 [cs].
- Juan A. Rodriguez, Nicholas Botzer, David Vazquez, Christopher Pal, Marco Pedersoli, and Issam H. Laradji. IntentGPT: Few-Shot Intent Discovery with Large Language Models. *ICLR 2024 Workshop on LLM Agents*, 2024. URL <https://openreview.net/forum?id=2kvDzdC5rh>.
- Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards Understanding Sycophancy in Language Models, October 2023. URL <http://arxiv.org/abs/2310.13548>. arXiv:2310.13548 [cs, stat].
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/1f89885d556929e98d3ef9b86448f951-Abstract.html>.
- Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. Large Language Models are Inconsistent and Biased Evaluators, May 2024. URL <http://arxiv.org/abs/2405.01724>. arXiv:2405.01724 [cs].
- Amos Tversky and Daniel Kahneman. Judgment under Uncertainty: Heuristics and Biases. *Science*, 185(4157): 1124–1131, 1974. doi: 10.1126/science.185.4157.1124.
- Minghao Wu and Alham Fikri Aji. Style Over Substance: Evaluation Biases for Large Language Models, November 2023. URL <http://arxiv.org/abs/2307.03025>. arXiv:2307.03025 [cs].

- Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. A Critical Evaluation of Evaluations for Long-form Question Answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.181. URL <https://aclanthology.org/2023.acl-long.181>.
- 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, July 2023. URL <http://arxiv.org/abs/2306.05685>. arXiv:2306.05685 [cs].
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and Transferable Adversarial Attacks on Aligned Language Models, December 2023. URL <http://arxiv.org/abs/2307.15043>. arXiv:2307.15043 [cs].

Appendix

A Additional experimental results

We further share results with respect to the n hyperparameter of our algorithm.

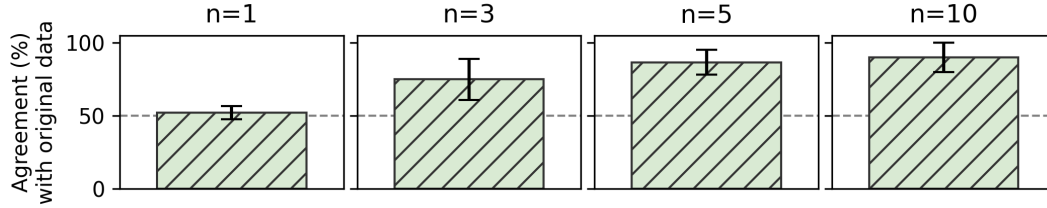


Figure 7: **Results when varying number n of principles in constitution on orthogonal synthetic data.** Whilst there is clear improvement noticeable from 1 to 3, and 3 to 5, we observe that there appear to be diminishing returns for values higher than 5. Note that the number of underlying principles is three, thus it may not be surprising that $n = 1$ does not work well. For $n = 3$, the algorithm needs to create three different principles that match the underlying three rules – which may be error prone. From $n = 5$ onwards it appears to robustly find corresponding principles for the underlying three rules. Thus, we use $n = 5$ in our experiments. Note that for further datasets additional experimentation may be important—the optimal value also depends on the annotator model’s capacity to deal with multiple principles simultaneously. Experiments use GPT-3.5-Turbo, reported values and error bars are mean and standard deviation over six random seeds.

B Prompts

Note that prompts are generally separated into two messages, a system message and a user message. We use the following format for all prompts (based on AlpacaEval’s formatting): `<|im_start|>` and `<|im_end|>` denote the start and end of a message, followed by the message type (system or user) and the content.

B.1 Principle generation

Unless otherwise specified, principles are generated with the following two generation prompts. We process each datapoint with both prompts to encourage the generation of a diverse set of principles that may both select for positive output traits (e.g. more helpful) and negative output traits (e.g. off-topic). Initial experiments indicated that it can be difficult to generate such a diverse set of possible principles with a single prompt, thus we use multiple (two) prompts by default. An exception is the Chatbot Arena dataset, where we use a single prompt that places increased emphasis on highly specific principles, to better capture individual differences between users.

Listing 1: Principle generation prompt, variant 1 (biased towards negative traits).

```
<|im_start|>system
Your job is to analyse data and come up with explanations. You're an
expert at this.
<|im_end|>
<|im_start|>user
Selected sample:
{preferred_sample}

Other sample:
{rejected_sample}

Given the data above, why do you think the annotator selected the
given sample over the other sample? Reply with {num_principles}
most likely rules that may explain the selection, each in 10
```

```

words or less. Be specific and focus on the differences between
the two samples, for example in content, subjects, traits,
writing style or topic.

Note: the intend of the selection was to find bad samples (to prevent
a user seeing them). Always suggest as rule that starts with
'Select the response that...<bad thing>'. Suggest rules that help
find bad samples.

Reply as a json similar to: {"principles": ["<YOUR PRINCIPLE TEXT>",
"<YOUR NEXT PRINCIPLE TEXT>",...]}}.
DO NOT respond with any text apart from the json format above!
DO NOT add markdown formatting around JSON.
ONLY REPLY IN JSON FORMAT
<|im_end|>

```

Listing 2: Principle generation prompt, variant 2.

```

<|im_start|>system
Your job is to analyse data and come up with explanations. You're an
expert at this.
<|im_end|>
<|im_start|>user
Selected sample:
{preferred_sample}

Other sample:
{rejected_sample}

Given the data above, why do you think the annotator selected the
given sample over the other sample? Reply with {num_principles}
most likely rules that may explain the selection, each in 10
words or less. Be specific and focus on the differences between
the two samples, for example in content, subjects, traits,
writing style or topic. Always suggest as rule that starts with
'Select the response that...'.

Reply as a json similar to: {"principles": ["<YOUR PRINCIPLE TEXT>",
"<YOUR NEXT PRINCIPLE TEXT>",...]}}.
DO NOT respond with any text apart from the json format above!
DO NOT add markdown formatting around JSON.
ONLY REPLY IN JSON FORMAT
<|im_end|>

```

Listing 3: Principle generation prompt, cross-user variant for Chatbot Arena.

```

<|im_start|>system
Your job is to analyse data and come up with explanations. You're an
expert at this.
<|im_end|>
<|im_start|>user
Selected sample:
{preferred_sample}

Other sample:
{rejected_sample}

Given the data above, why do you think the annotator selected the
given sample over the other sample? Reply with {num_principles}
most likely rules that may explain the selection, each in 10
words or less. Be specific and focus on the differences between
the two samples. Always suggest as rule that starts with 'Select
the response that...'. Important: suggest rules that are specific
to the shown samples, not general or generic rules! Do NOT

```

```

suggest generic rules like "select the more useful sample" or
"Select the response that directly answers the user's query".
Instead, suggest specific rules like "select x over y if z",
based on the specific samples and their topic z. For example, if
the samples are about translation, create rule in the context of
translation.
Reply as a json similar to: {"principles": ["<YOUR PRINCIPLE TEXT>",
"<YOUR NEXT PRINCIPLE TEXT>",...]}
DO NOT respond with any text apart from the json format above!
DO NOT add markdown formatting around JSON.
ONLY REPLY IN JSON FORMAT
<|im_end|>

```

B.2 Principle testing

The following prompt is used for testing how the principles affect LLM annotator on the training data set (Algorithm Step 4). Multiple principles are evaluated in parallel, given via the *summaries* variable.

Listing 4: Rule testing prompt.

```

<|im_start|>system
Your job is to check which sample is should be selected according to
the given rules. You're an expert at this.
<|im_end|>
<|im_start|>user
Sample A:
{sample_a}

Sample B:
{sample_b}

Given the samples data above, check for each rule below which sample
should be selected:
{summaries}

Answer in json format, e.g. {{0: "A", 1: "B", 2: "None",...}}.
Put "A" if A is selected according to that rule, and "B" if B is
selected. Put "None" if a rule is not applicable to the two
samples.
No ties are allowed, only one of "A", "B" or "None".
Vote for all rules, even if you are unsure.
DO NOT respond with any text apart from the json format above!
DO NOT add markdown formatting around JSON.
ONLY REPLY IN JSON FORMAT
<|im_end|>

```

B.3 Constitution evaluation

We use the following prompt to ask the LLM annotator to generate preferences based on a constitution. We use two prompts loosely based on 'chatgpt_fn' prompt from AlpacaEval, which was designed to evaluate the preferences of a language model without a constitution to follow. The first prompt, used in our synthetic and AlpacaEval experiments, is more generally applicable, relying on the LLM's learned knowledge about human preferences to fill in the gaps in the constitution. The second prompt is intended to focus on individual differences between constitutions, which may be small, and therefore further discourages the LLM annotator from relying on its own knowledge about human preferences.

Listing 5: Prompt for annotating according to constitution (AlpacaEval variant).

```

<|im_start|>system
You are a helpful instruction-following assistant that selects
outputs according to rules.

```



```

<|im_end|>
<|im_start|>user
Select the output (a) or (b) according to the following rules (if
they apply):
{constitution}

You MUST follow the rules above if they apply.
Select the output randomly if they do not apply.

Your answer should ONLY contain: Output (a) or Output (b).

# Task:
Now the task, do not explain your answer, just say Output (a) or
Output (b).

## Output (a):
{output_1}

## Output (b):
{output_2}

## Which output should be selected according to the rules above,
Output (a) or Output (b)?
<|im_end|>

```

Listing 6: Prompt for annotating according to constitution (Variant focusing on individual differences).

```

<|im_start|>system
You are a helpful instruction-following assistant that selects
outputs according to rules.
<|im_end|>
<|im_start|>user
Select the output (a) or (b) according to the following rules (if
they apply):
{constitution}

You MUST follow the rules above if they apply.
Select the output randomly if they do not apply.

Your answer should ONLY contain: Output (a) or Output (b).

# Task:
Now the task, do not explain your answer, just say Output (a) or
Output (b).

## Output (a):
{output_1}

## Output (b):
{output_2}

## Note:
If the rules do not apply, you MUST select randomly. DO NOT follow
you own opinion.

## Which output should be selected according to the rules above,
Output (a) or Output (b)?
<|im_end|>

```

B.4 Non-constitutional baseline

We also evaluate the preferences the language model expresses when not given a constitution to follow, i.e., the biases inherent in the trained model when asked to select the “best” output. We adapted two

of the default prompts from AlpacaEval for this purpose by removing references to an “instruction”, as this is not present in all pairwise ranking datasets. We selected the `alpacaeval_gpt4_turbo_fn` and `chatgpt_fn` prompts as they were reported to have the highest human agreement rate for the gpt-4-turbo and gpt-3.5-turbo models, respectively, while also being below an (estimated) price of 6\$/1k examples.¹⁰

Listing 7: Prompt for GPT-4, based on `alpaca_eval_gpt4_turbo_fn` from AlpacaEval.

```
<|im_start|>system
You are a highly efficient assistant, who evaluates and rank large
language models (LLMs) based on the quality of their responses to
given prompts. This process will create a leaderboard reflecting
the most accurate and human-preferred answers.
<|im_end|>
<|im_start|>user
I require a leaderboard for various large language models. I'll
provide you with prompts given to these models and their
corresponding responses. Your task is to assess these responses,
ranking the models in order of preference from a human
perspective. Once ranked, please output the results in a
structured JSON format for the make_partial_leaderboard function.

## Model Outputs

Here are the unordered outputs from the models. Each output is
associated with a specific model, identified by a unique model
identifier.

{
  {
    "model": "m",
    "output": "{output_1}"
  },
  {
    "model": "M",
    "output": "{output_2}"
  }
}

## Task

Evaluate and rank the models based on the quality and relevance of
their outputs. The ranking should be such that the model with the
highest quality output is ranked first.
<|im_end|>
```

Listing 8: Prompt for GPT-3.5-Turbo, based on `chatgpt_fn` from AlpacaEval.

```
<|im_start|>system
You are a helpful instruction-following assistant that prints the
best model by selecting the best outputs for a given instruction.
<|im_end|>
<|im_start|>user
Select the output (a) or (b) that best matches the given instruction.
Choose your preferred output, which can be subjective. Your
answer should ONLY contain: Output (a) or Output (b). Here's an
example:

# Example:

## Output (a):
```

¹⁰https://github.com/tatsu-lab/alpaca_eval/tree/v0.6.2/src/alpaca_eval/evaluators_configs

```

Instruction:
Give a description of the following job: "ophthalmologist"

Assistant:
An ophthalmologist is a medical doctor who specializes in the
    diagnosis and treatment of eye diseases and conditions.

## Output (b):

Instruction:
Give a description of the following job: "ophthalmologist"

Assistant:
An ophthalmologist is a medical doctor who pokes and prods at your
    eyes while asking you to read letters from a chart.

## Which is best, Output (a) or Output (b)?
Output (a)

Here the answer is Output (a) because it provides a comprehensive and
    accurate description of the job of an ophthalmologist. In
    contrast, output (b) is more of a joke.

# Task:
Now is the real task, do not explain your answer, just say Output (a)
    or Output (b).

## Output (a):
{output_1}

## Output (b):
{output_2}

## Which is best, Output (a) or Output (b)?
<|im_end|>

```

C Constitutions

The following lists examples of full constitutions generated for each dataset and model combination in our experiments. To provide an unbiased view of the generated constitutions, we show the constitution with the highest, median, and lowest performance on the test set. Since even numbers of seeds are used, we chose the worse-performing constitution as a tie-breaker for the median.

C.1 Synthetic datasets

Note that in our synthetic data experiments we test on the same data as we use to generate the constitutions, as these experiments serve as a proof-of-concept.

C.1.1 Aligned

Listing 9: Best constitution on the ‘aligned’ synthetic dataset.

1. Select the response that maintains a positive and helpful tone.
2. Select the response that shows a higher level of willingness.
3. Select the response that directly answers the question.
4. Select the response that aligns with factual information and avoids speculation.
5. Select the response that provides accurate and concise information.

Listing 10: Median constitution on the ‘aligned’ synthetic dataset.

1. Select the response that maintains a positive and helpful tone.

2. Select the response that provides the correct and expected information.
3. Select the response that provides the correct and factual information.
4. Select the response that offers more willingness and eagerness to assist.
5. Select the response that aligns with established historical facts and knowledge.

Listing 11: Worst constitution on the ‘aligned’ synthetic dataset.

1. Select the response that maintains a positive and helpful tone.
2. Select the response that is concise and to the point.
3. Select the response that shows a more positive and proactive attitude.
4. Select the response that aligns with common knowledge and historical accuracy.
5. Select the response that provides the correct and factual information.

C.1.2 Orthogonal

Listing 12: Best constitution on the ‘orthogonal’ synthetic dataset.

1. Select the response that emphasizes specific flavor (Lemon Ice Cream).
2. Select the response that features a cat instead of a dog.
3. Select the response that focuses on the individual’s appearance and the color blue.
4. Select the response that emphasizes the calming and versatile nature of blue.
5. Select the response that emphasizes the specific flavor mentioned.

Listing 13: Median constitution on the ‘orthogonal’ synthetic dataset.

1. Select the response that features a cat as the pet.
2. Select the response that emphasizes the versatility of the color.
3. Select the response that involves a humorous pet-owner interaction.
4. Select the response that emphasizes the calming and comforting qualities of blue.
5. Select the response that offers a citrus flavor option.

Listing 14: Worst constitution on the ‘orthogonal’ synthetic dataset.

1. Select the response that focuses on stability and confidence in clothing choice.
2. Select the response that features a cat as the pet.
3. Select the response that suggests a tangy treat for refreshment.
4. Select the response that emphasizes lightness, refreshment, and flavor.
5. Select the response that mentions a specific flavor (lemon, raspberry).

C.1.3 Unaligned

Listing 15: Best constitution on the ‘unaligned’ synthetic dataset.

1. Select the response that lacks enthusiasm.
2. Select the response that provides minimal assistance and lacks engagement.
3. Select the response that includes a negative or dismissive tone.
4. Select the response that provides inaccurate information.
5. Select the response that provides incorrect details about the US capital.

Listing 16: Median constitution on the ‘unaligned’ synthetic dataset.

1. Select the response that shows minimal engagement or empathy towards user.
2. Select the response that lacks a positive and proactive tone.
3. Select the response that sets lower expectations for help provision.
4. Select the response that is more casual and less formal.
5. Select the response that shows reluctance to provide assistance.

Listing 17: Worst constitution on the ‘unaligned’ synthetic dataset.

1. Select the response that lacks enthusiasm and willingness to assist.
2. Select the response that lacks a proactive and helpful tone.
3. Select the response that lacks specific details about the destination.
4. Select the response that provides a vague and less helpful answer.
5. Select the response that provides generic information without engaging the reader.

C.2 AlpacaEval datasets

C.2.1 Aligned

Listing 18: Best constitution on the ‘aligned’ AlpacaEval dataset.

1. Select the response that includes redundant information.
2. Select the response that provides detailed information and context.
3. Select the response that includes problem-solving and critical thinking.
4. Select the response that uses consistent category naming.
5. Select the response that provides more practical examples.

Listing 19: Median constitution on the ‘aligned’ AlpacaEval dataset.

1. Select the response that includes redundant information.
2. Select the response that is overly verbose and repetitive.
3. Select the response that provides more practical examples.
4. Select the response that uses more engaging and descriptive language.
5. Select the response that uses more vivid and engaging imagery.

Listing 20: Worst constitution on the ‘aligned’ AlpacaEval dataset.

1. Select the response that maintains a neutral and informative tone.
2. Select the response that avoids spelling or grammatical errors.
3. Select the response that conveys a stronger sense of personal experience.
4. Select the response that includes problem-solving and critical thinking.
5. Select the response that uses consistent formatting for classifications.

C.2.2 Unaligned

Listing 21: Best constitution on the ‘unaligned’ AlpacaEval dataset.

1. Select the response that uses simpler, less engaging language.
2. Select the response that contains incorrect or nonsensical logic.
3. Select the response that lacks detailed achievements.
4. Select the response that lists key takeaways clearly and concisely.
5. Select the response that maintains consistency in classification.

Listing 22: Median constitution on the ‘unaligned’ AlpacaEval dataset.

1. Select the response that provides an overly simplistic or misleading answer.
2. Select the response that lacks depth in analysis.
3. Select the response that contains incorrect or nonsensical logic.
4. Select the response that lists all entities in the text.
5. Select the response that ends abruptly without a conclusion.

Listing 23: Worst constitution on the ‘unaligned’ AlpacaEval dataset.

1. Select the response that changes the meaning slightly.
2. Select the response that uses more technical language.
3. Select the response that maintains the original order of entities.
4. Select the response that lacks specific examples or details.
5. Select the response that uses fewer abstract concepts.

C.3 Chatbot Arena

Note that for personalized constitutions we measure performance based on the ability to reconstruct the same user’s preferences. Due to the small number of samples, there is no split between test and training data.

C.3.1 User A

Listing 24: Best constitution on User A annotations.

1. Select the response that avoids anachronistic errors.
2. Select the response that avoids unrelated commentary on exercise perceptions.
3. Select the response that provides context about the word ‘plagiarism’.

Listing 25: Median constitution on User A annotations.

1. Select the response that provides a detailed and clear explanation.
2. Select the response that explains the joke’s wordplay clearly.
3. Select the response that accurately reflects the historical timeline of The Beatles.

Listing 26: Worst constitution on User A annotations.

1. Select the response that provides a clear and accurate explanation.
2. Select the response that directly explains the pun in the joke.
3. Select the response that references specific scenes or characters.

C.3.2 User B

Listing 27: Best constitution on User B annotations.

1. Select the response that avoids abrupt or incomplete endings.
2. Select the response that concludes the story more definitively.
3. Select the response that provides a more detailed and structured argument.

Listing 28: Median constitution on User B annotations.

1. Select the response that avoids abrupt or incomplete endings.
2. Select the response that maintains a consistent dark and ominous tone.
3. Select the response that evokes stronger emotional engagement.

Listing 29: Worst constitution on User B annotations.

1. Select the response that avoids abrupt or incomplete endings.
2. Select the response that conveys a stronger emotional impact.
3. Select the response that concludes the story more definitively.

D Numerical Results

In addition to the bar plots shown in the main paper, we provide a table of numerical results for all experiments here.

Table 1: Results for experiments on synthetic data. Averaged over 6 random seeds.

Dataset	Model	Annotator	Mean	Std	Min	Max
Orthogonal	GPT-3.5 Turbo	Constitutional	86.67%	8.43	73.33%	96.67%
		Default	37.78%	2.72	33.33%	40.00%
Aligned	GPT-3.5 Turbo	Constitutional	92.22%	6.89	83.33%	100.00%
		Default	100.00%	0.00	100.00%	100.00%
Unaligned	GPT-3.5 Turbo	Constitutional	84.44%	9.81	73.33%	100.00%
		Default	0.00%	0.00	0.00%	0.00%

Table 2: Results for experiments on AlpacaEval data. Averaged over 6 random seeds.

Dataset	Model	Annotator	Mean	Std	Min	Max
Aligned	GPT-3.5-Turbo	Default	64.87%	1.16	63.08%	66.15%
		Constitutional	67.44%	3.43	63.08%	72.31%
	GPT-4o	Default	72.56%	1.16	70.77%	73.85%
		Constitutional	68.46%	3.19	63.08%	72.31%
Unaligned	GPT-3.5-Turbo	Default	35.90%	1.26	33.85%	36.92%
		Constitutional	39.49%	3.32	35.38%	44.62%
	GPT-4o	Default	26.92%	1.61	24.62%	29.23%
		Constitutional	66.41%	7.69	53.85%	72.31%

Table 3: Results for cross-user experiments on Chatbot Arena data. Averaged over 6 random seeds.

Dataset	Model	Annotator	Mean	Std	Min	Max
Annotations User A	GPT-4o	User A constitution	93.06%	3.40	91.67%	100.00%
		User B constitution	83.33%	5.27	75.00%	91.67%
Annotations User B	GPT-4o	User A constitution	79.63%	10.92	66.67%	88.89%
		User B constitution	94.44%	6.09	88.89%	100.00%

Table 4: Results for cross-model experiments on AlpacaEval data. Averaged over 4 random seeds.

Dataset	Model	Annotator	Mean	Std	Min	Max
Unaligned	GPT-4o	Default	26.15%	1.26	24.62%	27.69%
		Constitutional	70.00%	0.89	69.23%	70.77%
	Claude-3-Opus	Default	24.23%	0.77	23.08%	24.62%
		Constitutional	58.85%	0.77	58.46%	60.00%
	Claude-3-Haiku	Default	36.92%	0.00	36.92%	36.92%
		Constitutional	59.65%	0.00	59.65%	59.65%

Table 5: Results for the sensitivity study on parameter n (rules per constitution) on synthetic data. Averaged over 6 random seeds.

Dataset	Model	Annotator	Mean	Std	Min	Max
Unaligned	GPT-3.5 Turbo	Constitutional (n=1)	52.22%	4.55	43.33%	56.67%
		Constitutional (n=3)	75.00%	14.10	63.33%	100.00%
		Constitutional (n=5)	86.67%	8.43	73.33%	96.67%
		Constitutional (n=10)	90.00%	10.11	70.00%	96.67%

E Cost estimates

In this section we estimate the cost of reproducing all experiments shown in this paper. All experiments were run using models via API access from OpenAI and Anthropic. Note that all estimates are subject to variability due to provider pricing as well as inherent variability of the length (and thus cost) of model outputs.

Synthetic experiments. The first set of experiments are the synthetic experiments, which are entirely run using GPT-3.5-Turbo. Per run (30 samples, 1 constitution, annotation on same 30 samples) these experiments cost approximately 0.05\$. Overall, we estimate it would cost 2.7\$ to re-run all experiments shown (3 datasets \times 6 seeds).

AlpacaEval experiments. The second set of experiments are the AlpacaEval experiments, split into the main aligned/unaligned experiments as well as cross-model experiments. The main experiments cost approx. 2.20\$ per seed. Overall, we estimate it would cost 26.40\$ to re-run all of the main experiments (2 datasets \times 6 seeds). Additionally, we estimate the cross-model (just annotation) experiments would cost 5.00\$.

Chatbot Arena experiments. The third set of experiments are the Chatbot Arena experiments, split into the main aligned/unaligned experiments as well as cross-model experiments. The main experiments cost approx 1.10\$ per seed. Overall, we estimate it would cost 13.20\$ to re-run all of the main experiments (2 datasets \times 6 seeds).

We estimate the remaining cost of experiments to be less than 5\$. Overall we thus estimate the total cost of re-running our experiments to approx. 52.30\$ in API costs. Note that the overall cost for running experiments in the context of this project was about 3 times this amount (approx. 156.90\$), due to failed runs and additional experimentation that did not fit into the scope of the paper.

F Synthetic data generation

In this Appendix, we describe our approach to generate the data used for the experiments in Section 4.1. We use an LLM (gpt-3.5-turbo-0125) to generate the data, given the various prompts below. We include the full data generation pipeline as part of our code, including additional prompt appendices that are added for more diversity. For each of the principles here we generate 10 samples with the corresponding prompts. The unaligned data is simply the aligned dataset with the preference labels flipped.

F.1 Orthogonal data

F.1.1 Example

Listing 30: Orthogonal data example

```
Text A:
Instruction: Please write a funny short story about a human and
            their pet.
Output: Tom was amazed by how his cat, Mr. Whiskers, could always
        find the best hiding spots in their house. One day, Tom decided
        to play a game of hide and seek with Mr. Whiskers, but as soon
        as he closed his eyes to count, he heard a loud crash followed
        by a meow of victory. When he opened his eyes, he saw Mr.
```

Whiskers proudly perched on top of the refrigerator, knocking down a stack of plates in the process. Tom couldn't help but laugh at his mischievous feline friend. After all, life with Mr. Whiskers was never boring.

Text B:

Instruction: Please write a funny short story about a human and their pet.

Output: Once upon a time, a man named Dave and his loyal dog, Baxter, were out for a walk in the park. Suddenly, Baxter spotted a squirrel and took off running, dragging Dave behind him on his leash. Dave couldn't help but laugh as he struggled to keep up with his energetic pup, who was determined to catch the elusive critter. After a few minutes of pure chaos, Baxter finally gave up the chase and plopped himself down in a puddle, wagging his tail happily. Dave shook his head and said, "You may be a handful, but you're definitely worth it, buddy."

Preferred: Text A

F.1.2 Prompts for principle "prefer cats over dogs"

Listing 31: Generation prompt 1 for synthetic orthogonal data

Please write a funny short story about a human and their pet. Their pet is a {differing_part}. Max 5 sentences.

Where the `differing_part` variable is set to one of ["cat", "dog"]. In the dataset, we replace this original instruction with the following instruction.

Listing 32: Instruction 1 for synthetic orthogonal data

Please write a funny short story about a human and their pet.

F.1.3 Prompts for principle "prefer green over blue color"

Listing 33: Generation prompt 2 for synthetic orthogonal data

Give a short reasoning why I should pick a {differing_part} t-shirt. Mention the color. Max 1 sentences.

Where the `differing_part` variable is set to one of ["blue", "green"]. In the dataset, we replace this original instruction with the following instruction.

Listing 34: Instruction 2 for synthetic orthogonal data

Should I pick this blue t-shirt or the green one?

F.1.4 Prompts for principle "select lemon over raspberry ice-cream"

Listing 35: Generation prompt 3 for synthetic orthogonal data

Tell me why I should pick the {differing_part} ice cream. Max 1 short sentence. Include the name of the flavor.

Where the `differing_part` variable is set to one of ["lemon", "raspberry"]. In the dataset, we replace this original instruction with the following instruction.

Listing 36: Instruction 3 for synthetic orthogonal data

Which ice cream flavor should I pick?

F.2 Aligned data

F.3 Example

Listing 37: Aligned data example

```
Text A:
  Instruction: Can you help me?
  Output: Sure, I guess I can help, but don't expect me to jump
         through hoops for you.

Text B:
  Instruction: Can you help me?
  Output: Of course! How can I assist you?

Preferred: Text B
```

F.3.1 Prompts for principle "select truthful over factually incorrect answers"

Listing 38: Generation prompt 1a for synthetic aligned data

```
What is the capital of the US? Max 1 sentence.
```

Listing 39: Generation prompt 1b for synthetic aligned data

```
Give me a made-up reason why Paris is the capital of the US. Max 1
sentence.
```

In the dataset, we replace this original instructions with the following instruction.

Listing 40: Instruction 1 for synthetic aligned data

```
What is the capital of the US?
```

F.3.2 Prompts for principle "select helpful over useless answers"

Listing 41: Generation prompt 2a for synthetic aligned data

```
Give me one interesting destination to travel to in the UK. Max 1
sentence.
```

Listing 42: Generation prompt 2b for synthetic aligned data

```
Why is it good to travel? Do not mention any specific destination
names. Max 1 sentence.
```

In the dataset, we replace this original instructions with the following instruction.

Listing 43: Instruction 2 for synthetic aligned data

```
What would be an interesting destination to travel to in the UK?
```

F.3.3 Prompts for principle "select polite over impolite answers"

Listing 44: Generation prompt 3a for synthetic aligned data

```
Can you help me?
```

Listing 45: Generation prompt 3b for synthetic aligned data

```
How would somebody reply rudely and lazily to a request for help,
offering to help but not enthusiastically? Max 1 sentence.
```

In the dataset, we replace this original instructions with the following instruction (identical to generation prompt 3a in Listing 44).

Listing 46: Instruction 3 for synthetic aligned data

```
Can you help me?
```