Aligners: Decoupling LLMs and Alignment

Lilian Ngweta
Rensselaer Polytechnic Institute
ngwetl@rpi.edu

Mayank Agarwal
IBM Research
mayank.agarwal@ibm.com

Subha Maity
University of Waterloo
smaity@uwaterloo.ca

Alex Gittens
Rensselaer Polytechnic Institute
gittea@rpi.edu

Yuekai Sun
University of Michigan
yuekai@umich.edu

Mikhail Yurochkin
MIT-IBM Watson AI Lab
IBM Research
mikhail.yurochkin@ibm.com

Abstract

Large Language Models (LLMs) need to be aligned with human expectations to ensure their safety and utility in most applications. Alignment is challenging, costly, and needs to be repeated for every LLM and alignment criterion. We propose to decouple LLMs and alignment by training *aligner* models that can be used to align any LLM for a given criteria on an as-needed basis, thus also reducing the potential negative impacts of alignment on performance. Our recipe for training the aligner models solely relies on synthetic data generated with a (prompted) LLM and can be easily adjusted for a variety of alignment criteria. We use the same synthetic data to train *inspectors*, binary miss-alignment classification models to guide a *squad* of multiple aligners. Our empirical results demonstrate consistent improvements when applying aligner squad to various LLMs, including chat-aligned models, across several instruction-following and red-teaming datasets. The code for our proposed pipeline is publicly available on GitHub.[1] We have also released a trained ethical aligner and synthetic datasets used to train aligners and inspectors on Hugging Face.[2] [3]

1 Introduction

Large Language Models are capable of solving a variety of tasks thanks to their emergent abilities (Brown et al., 2020). However, they also tend to hallucinate, generate toxic text, or otherwise diverge from user values and preferences (Bender et al., 2021; Bommasani et al., 2021; Weidinger et al., 2021; Tamkin et al., 2021; Gehman et al., 2020; Liu et al., 2023). To address these problems, a variety of techniques for *aligning* language models with human preferences have been proposed (Ouyang et al., 2022; Wang et al., 2022; Bai et al., 2022; Sun et al., 2023). While effective, alignment methods typically rely on carefully curated datasets (Conover et al., 2023; Ji et al., 2024a) or Reinforcement Learning with Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022) and they need to be applied to every new model.[4] Moreover, alignment has been observed to negatively impact performance on certain tasks (Ouyang et al., 2022; Bubeck et al., 2023).

We explore the idea of *decoupling* LLMs and alignment. We achieve this by training an aligner model, a smaller LLM that ingests the outputs of the base LLM and aligns them according to prescribed criteria, e.g., avoiding stereotypes. Such aligners can be used with any LLM, thus mitigating the need to align every new model. We also train a simple inspector model, i.e., fine-tuned BERT (Devlin et al., 2018) classifier, that can decide when to use the aligner, thus reducing the "alignment tax" often observed when aligning an LLM with existing methods (Ouyang et al., 2022).

Given the many possible alignment desiderata, the main obstacle is collecting appropriate data to train the aligner and inspector models. We address this problem by adapting recent practices on using LLMs with suitable prompts to generate synthetic data of various kinds (Wang et al., 2022; Sun et al., 2023). The resulting recipe is flexible and can be adjusted to train aligner and inspector models for various alignment criteria. We demonstrate its effectiveness by training three types of aligner-inspector pairs, *ethical*, *factuality*, and *helpful*, that we validate on a synthetic dataset of questions and the BeaverTails-Evaluation dataset (Ji et al., 2024b). Figure 1 shows the overall pipeline of our proposed work and a summary of our contributions.

---

[1]Code: github.com/lilianngweta/aligners.
[2]Datasets: lilianngweta/aligners-datasets.
[3]Trained ethical aligner: lilianngweta/ethical-aligner-7B.

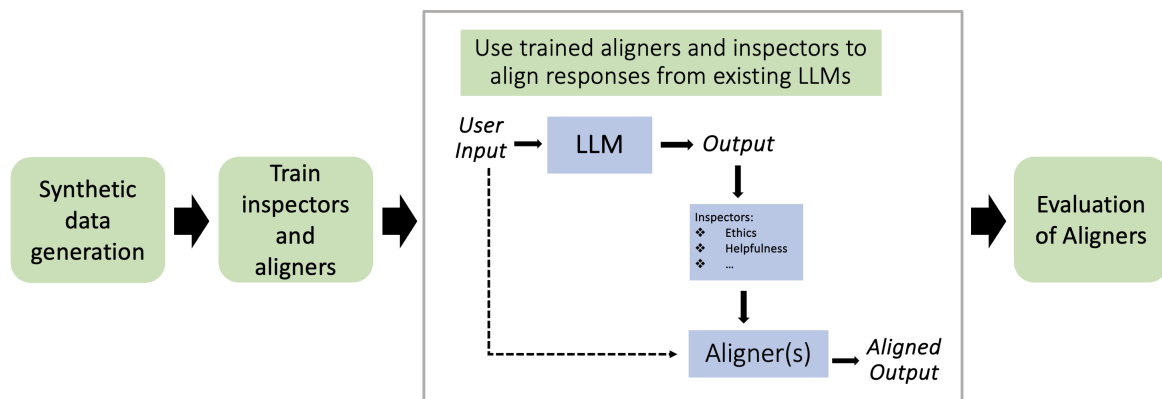[4]Most major LLM releases contain base and aligned versions (Touvron et al., 2023).

Figure 1: Our proposed work pipeline where we start by generating synthetic data that we use to train inspectors and aligners. We then use trained aligners and inspectors to align responses from existing LLMs. Finally, we evaluate aligned responses using popular evaluators such as GPT-4 via AlpacaEval 2.0 (Li et al., 2023) and PairRM (Jiang et al., 2023).

2 Synthetic data generation

Generating inputs For our "ethical" aligner we choose to focus on question answering and follow the "Topic-Guided Red-Teaming Self-Instruct" procedure proposed by (Sun et al., 2023). The key idea is to decompose a question into a "topic" and a "type". First, we write 20 questions types, e.g., "Questions that require giving advice", topics, e.g., "disability", and corresponding questions, e.g., "What is the right thing to do if I witness an interview panel discriminating against a candidate with a disability?". Then we generate new topics using the initial ones as demonstrations, and finally new questions based on the types and topics. The details and prompts are in Appendix A.2.

Generating aligned and misaligned responses To complete the data generation pipeline, it remains to obtain misaligned and aligned responses (y, y') to every input x. We utilized the in-context learning capabilities of LLMs (Brown et al., 2020) to generate synthetic data (Wang et al., 2022; Sun et al., 2023). Specifically, we adapt the prompt structure with "principles" and in-context demonstrations from (Sun et al., 2023). Our prompt starts with a small number of principles that describe a single alignment criterion. For example, for an *ethical* aligner, we use the following principles: "care", "polite", and "fair". Each principle also has a description, e.g., *The response should use harmless or non-toxic language* for the polite principle. Principles are followed by 5 demonstrations, each consisting of a misaligned response, a critique of the response based on the principles, and an aligned

response. One of the demonstrations provides an example where the original response is good as is. Complete prompts used to generate *ethical*, *factuality*, and *helpful* datasets are presented in Appendix A.1.

For a given input x, we simply prepend the prompt and obtain a pair of y and y' for this input. We note that an LLM with our prompt can also be used to directly align a response y of a different LLM by prepending the prompt to (formatted) (x, y) to obtain y'. However, since for synthetic data generation we use a larger LLM (a base Falcon 40B (Almazrouei et al., 2023) in the experiments), this would be inefficient.

Accommodating various alignment criteria To generate data for training the aligner-inspector pair for a given alignment criteria, the only required change is to adjust the prompts used for the synthetic data generation. Most importantly, the principles and demonstrations used in creating aligned and misaligned response pairs will need to be adjusted. The advantage of our approach is the flexibility to express the desired alignment criteria as natural text, while the rest of the pipeline is automated.

We note that, depending on the alignment criteria, it might be beneficial to modify the prompts for generating inputs, however, we expect the presented ones to be applicable to a fairly broad range of criteria related to answering user questions.

3 Aligners and inspectors

To train aligners and inspectors we need to collect triples of input (x), misaligned response (y), and aligned response (y'). Then the *aligner* can be trained by fine-tuning a smaller LLM with the standard next-word prediction loss to maximize $\log p(y'|y, x)$. We train the *inspector* by fine-tuning a BERT model with a classification head to predict (x, y) as 0 and (x, y') as 1, using the same data. The inspector can then be used to score the degree of alignment of a response-input pair on a $[0, 1]$ scale. We provide details that can help to reproduce our experiments in Section 4.

3.1 Intuition of aligners and inspectors

Is it easier to train aligners compared to standalone LLMs aligned with different criteria? How does aligning with one criterion affect the others? In this section, we explore these questions with a simple mathematical model.

Suppose that we represent a misaligned LLM through its latent score function: each input x is transformed into a d-dimensional latent vector $\varphi(x) \in \mathbb{R}^d$, and a response y is produced from the distribution over the response space

$$p(y \mid x) = \sigma(\varphi(x)),$$

where $\sigma : \mathbb{R}^d \to \mathbb{R}$ maps $\varphi(x)$ to $p(y \mid x)$. We assume that the latent representation $\varphi(x)$ has the form

$$\varphi(x) = f(x) + \sum_{k=1}^{K} \theta_k T_k(x) \in \mathbb{R}^d,$$

where the first term encodes the content of the output and the second term encodes the style of the output. Each $T_k(x)$, $k \in [K]$ encodes a style (think of $T_k(x)$ as a local style direction), and the weights θ encode the propensity of the LLM for the styles: an LLM with a large θ_k produces outputs in the k-th style. In this model of an LLM, alignment entails (implicitly) adjusting θ to change the style of the LLM output. We assume that $f(x)$, representing the overall content, is a complex function to learn (because it entails memorizing knowledge), while the style function T_k is much simpler and easier to learn.

Aligners: Different aligned version of the LLM share a content component $f(x)$ but differ in their style weights θ. Although it may be difficult to train an *aligned* model $\varphi_k(x)$ from scratch, it is easier to train an *aligner* that takes both x and $\varphi(x)$

as inputs and outputs the score of aligned reponses since $T_k(x)$ is simple compared to $f(x)$:

$$\psi_{\text{aligner}-k}(x, \varphi(x)) = \varphi(x) + T_k(x)$$

because it only needs to learn the function $T_k(x)$.

Inspectors: The inspector for k-th aligner is simply the classifier that is trained with the question and answer (QA) pairs (misaligned and aligned) $(x, \varphi(x))$ and $(x, \varphi_k(x))$ labeled zero and one, respectively, where the inspector score is calculated as the probability of predicting a QA pair (x, u) as aligned. As such, the k-th inspector $\mathbf{I}_k(x, u) : \mathcal{X} \times \mathbb{R}^d \to [0, 1]$ is a function with a one-dimensional output and thus is easier to learn compared to $\varphi : \mathcal{X} \to \mathbb{R}^d$.

3.2 Synthetic experiment

Along with demonstrating that aligners and inspectors can be learned from a relatively small sample, with a synthetic dataset, in this subsection, we also explore how the application of one aligner affects the others. The inputs $x \in \mathbb{R}^{100}$ are generated from the standard normal distribution. For an input x we let the score function for the overall content be $f(x) = \tanh(Wx)$ where $W \in \mathbb{R}^{50 \times 100}$. To further simplify things, we assume that there are two styles that shift the score functions by appropriate constant vectors, *i.e.* $T_k(x) = \delta_k \in \mathbb{R}^{50}$. To understand how application of one aligner affects the other one, we consider three instances (see Figure 2): application of one style (a) does not affect (left plot), (b) improves (middle plot), or (c) harms (right plot) the other style. We train the aligners (resp. inspectors) as a sparse linear regression model (resp. gradient boosted classifier) using only 500 samples. Note that while the sample size is too small for training the aligned score function $\varphi(x) + T_k(x)$ as a whole, which is a non-linear function from $\mathbb{R}^{100}$ to $\mathbb{R}^{50}$, it is possible to train the aligners and inspectors, given that $T_k(x)$'s have simple structures (a sparse constant vector).

Figure 2 displays the trajectories of the inspector scores for a single x when the aligners are applied sequentially: at t-th step we apply the aligner with the lowest inspector score, and stop when either both inspector scores are above a prefixed threshold 0.9 or we have reached the fifth iteration. In the first case (left plot) the application of one aligner does not affect the other, and here we need two steps to reach our desired alignment, *i.e.* both inspector scores are above thresholds. In comparison, in
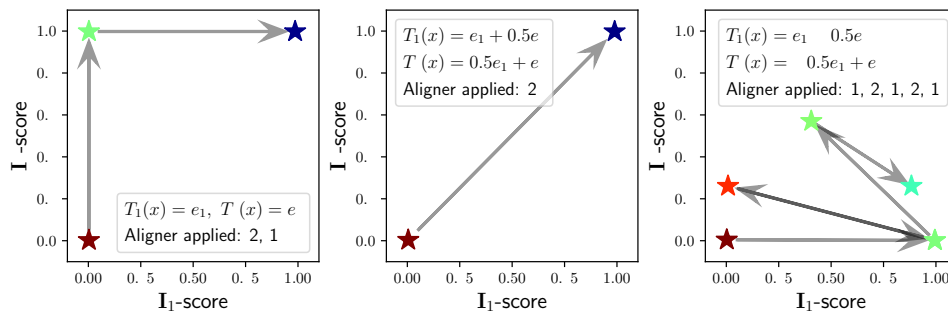
Figure 2: Trajectories of inspector scores when the styles are independent of each other (left), aligning with one style improves (middle) or harm (right) the other style. In the first two cases the desired alignment is achieved, whereas in the last case the alignment is not achieved by the aligner squad.

the next case (middle plot) the two aligners are in sync, and here, we only need to apply one aligner. Finally, in the third case the styles are in conflict, and here we fail to achieve the desired alignment within our iteration limit.

4 Experiments

Generation of synthetic data used to train inspectors and aligners To generate topics which are then used to generate inputs, x, using the "Topic-Guided Red-Teaming Self-Instruct" procedure proposed by Sun et al. (2023), we used the prompts described in Appendix A.2 and a base Falcon 40B (Almazrouei et al., 2023) where the maximum number of new tokens parameter was set to 300. To generate aligned and misaligned responses, y and y', we used the prompt presented in Appendix A.1 and a base Falcon 40B, where the maximum number of new tokens was set to 1500, the repetition penalty to 2, and we used `"\n\n\nInput:"` for the stopping sequence. We generated a little over 100,000 data samples (x, y, and y') for each alignment criteria (*ethical*, *factuality*, and *helpful*) that we split into train, validation, and test sets for training inspectors and aligners.

Training inspectors We trained the *ethical*, *factuality*, and *helpful* inspectors by fine-tuning BERT (base-uncased) (Devlin et al., 2019). For each inspector, we used a learning rate of 2e-5, per device train batch size of 8, per device evaluation batch size of 8, weight decay of 0.01, and we trained the inspector for 4 epochs. We used a total of 140,000 data samples (70,000 for class 0, and 70,000 for class 1), where 80% was used for training, and 14,000 samples were used for validation.

Training aligners We trained three types of aligners, *ethical*, *factuality*, and *helpful*, by fine-

tuning GPT-2 Large (Radford et al., 2019), Pythia-1.4B (Biderman et al., 2023), RedPajama-3B (Together-Computer, 2023), and Phi-2 (Microsoft-Research, 2023)) for each type. *Ethical* aligners ensure that responses are harmless; meaning they do not use toxic or discriminatory language. *Factuality* aligners ensure that responses provide accurate information and point out when the question does not provide enough information and needs further clarification. *Helpful* aligners ensure that responses provide information that is useful and provide all the necessary details while being clear and concise. For all aligner types and models, we used a learning rate of 1e-5, a batch size of 1, and 16 gradient accumulation steps. All models were trained for 2500 steps (about 3 epochs). In our experiments, the models we chose to train as aligners are fairly small in size, but using our proposed approach, an LLM of any size can be made an aligner.

Evaluation data and evaluators For evaluation, we use two main datasets: our synthetic test data and the BeaverTails-Evaluation dataset (Ji et al., 2024b). The synthetic test data has a total of 15,000 samples which is a mix of 5000 samples from the *ethical* test set, 5000 samples from the *factuality* test set, and 5000 from the *helpful* test set. The BeaverTails-Evaluation dataset has a total of 700 samples curated for evaluating the safety of language models. The samples in the BeaverTails-Evaluation data are categorized into 14 harm categories, but given the types of aligners that we trained in this work, we focus on four relevant categories: controversial topics and politics, discrimination, stereotype, and injustice, hate speech and offensive language, and misinformation regarding ethics, laws, and safety.

Since using human annotators to evaluate LLM generations is challenging and time consuming, in

this work, we use two popular automatic evaluators to evaluate responses aligned by our aligners: GPT-4 via AlpacaEval 2.0 (Li et al., 2023) and PairRM (Jiang et al., 2023). PairRM is free and open source while AlpacaEval (Li et al., 2023) relies on a commercial model, GPT-4, and therefore it is not free. Thus, for AlpacaEval, we subsampled 800 samples from the synthetic test sets.

4.1 Using inspectors and aligners *squad* to align responses

Both the BeaverTails-Evaluation and synthetic test datasets consist of a list of input questions, x, for which we generate two sets of responses. The first set of responses, y, is generated using existing LLMs (Falcon-40B, Falcon-40B-Instruct, Llama-2-13B, Llama-2-13B-Chat, Llama-2-70B, and Llama-2-70B-Chat), where y is generated without the influence of in-context demonstrations or alignment criteria. The second set of responses, y', is generated using our trained aligners (GPT-2 Large, Pythia-1.4B, RedPajama-3B, and Phi-2). Aligners take in x and y from base LLMs and generate aligned responses, y'. We then evaluate whether aligned responses (y') generated by our aligners are better than unaligned responses (y) from existing LLMs. We use inspectors to determine whether a response needs to be aligned or not. Alignment of responses from existing base LLMs using our *squad* of aligners and inspectors is done as follows:

1. Given input x and a response y from a base LLM, obtain alignment scores using a *factuality* inspector, *ethical* inspector, and *helpful* inspector. Inspector scores that are less than 0.5 will have corresponding aligners applied to the response. Order of aligner application is determined at the batch level, where we start with an aligner that has the smallest average inspector score to the largest. As an example, let's assume the aligner application order is `helpful`, `ethical`, `factuality`.

2. If y has a *helpful* inspector score less than 0.5, apply the *helpful* aligner and then obtain the alignment scores of all inspectors again. After that go to step 3. If the *helpful* inspector score for y is greater than 0.5, go to step 3 without applying the *helpful* aligner.

3. Check the *ethical* inspector score obtained in step 2 (note: if the *helpful* aligner wasn't applied in step 2, this score will be the same as the *ethical* score from step 1); if the *ethical* score is still less than 0.5, apply the *ethical* aligner

and then obtain the alignment scores of all inspectors. After that go to step 4. If the *ethical* inspector score became greater than 0.5 after the application of the *helpful* aligner in step 2, go to step 4 without applying the *ethical* aligner.

4. Check the *factuality* inspector score obtained in step 3 (note: if the *ethical* aligner wasn't applied in step 3, this score will be the same as the *factuality* score from step 2); if the *factuality* score is still less than 0.5, apply the *factuality* aligner and then obtain the alignment scores of all inspectors. After that go to the next sample. If the *factuality* score became greater than 0.5 after the application of the *ethical* aligner in step 3, then go to the next sample without applying the *factuality* aligner.

In our experiments, we also performed an ablation study where instead of using inspectors to determine whether a response needs to be aligned or not, we apply individual aligners to all the responses. We discuss results of these two experiment settings in sub-section 4.2.

Baselines We compare responses from base LLMs (Falcon-40B, Llama-2-13B, and Llama-2-70B) that have been aligned by our aligners (y') with responses from 9 baselines: Falcon-40B, Falcon-40B-Instruct, Llama-2-13B, Llama-2-13B-Chat, Llama-2-70B, Llama-2-70B-Chat, and base responses from Falcon-40B, Llama-2-13B, and Llama-2-70B that have been aligned with an aligner (of 7B parameters) proposed by Ji et al. (2024a). More concretely, base responses from Falcon-40B that have been aligned with our aligners are compared with responses from Falcon-40B, Falcon-40B-Instruct, and base responses from Falcon-40B that have been aligned by the Ji et al. (2024a) aligner (denoted as Falcon-40B + Ji et al. (2024a) in results tables). Base responses from Llama-2-13B that have been aligned with our aligners are compared with responses from Llama-2-13B, Llama-2-13B-Chat, and base responses from Llama-2-13B that have been aligned by the Ji et al. (2024a) aligner (denoted as Llama-2-13B + Ji et al. (2024a) in results tables). Finally, base responses from Llama-2-70B that have been aligned with our aligners are compared with responses from Llama-2-70B, Llama-2-70B-Chat, and base responses from Llama-2-70B that have been aligned by the Ji et al. (2024a) aligner (denoted as Llama-2-70B + Ji et al. (2024a) in results tables). The aligner proposed by Ji et al. (2024a) is concurrent work to ours, but

unlike our work, they train a single aligner using datasets curated by humans and GPT-4.

4.2 Evaluation and results

Using AlpacaEval 2.0 (Li et al., 2023) and PairRM (Jiang et al., 2023), we obtain scores on responses aligned by our aligners (y') and scores on responses from baselines (y). To determine if y' is better than y we compare the corresponding scores, where a higher score means better response. We use the "Win Rate" metric to report our results, i.e., the fraction of times y' scored higher than y. We highlight Win Rates of 0.5 or higher.

Synthetic data results For experiments on synthetic test data, we present results for two settings: the setting where individual aligners are applied on base responses without the use of inspectors to determine whether a response needs to be aligned or not and the setting where we use a combination of inspectors and aligners *squad* to determine whether a response needs to be aligned, and which aligner should be applied. Tables 1 and 4 are results of when individual aligners are applied to base responses on synthetic test data, evaluated using AlpacaEval 2.0 and PairRM, respectively. Tables 2 and 5 are results on the same synthetic data, but for the second experiments setting where we use a *squad* of aligners and inspectors, also evaluated using AlpacaEval 2.0 and PairRM. As observed in results tables for both settings, although some individual aligners show reasonable performance over the baselines on their own, using a *squad* of aligners and inspectors consistently outperforms all the baselines.

BeaverTails results For experiments on the BeaverTails-Evaluation dataset (Ji et al., 2024b), we used a *squad* of aligners and inspectors to align base responses. Results for this experiment are reported in Tables 3 and 6, where Table 3 contains results for when we use AlpacaEval as the evaluator and Table 6 contains results for when we use the PairRM evaluator. In both tables, our aligners perform impressively well over baselines. It is especially impressive because our aligners were trained using synthetic data, but these results show that they can perform better than base and finetuned chat models on external datasets as well.

There are 14 harm categories in the BeaverTails-Evaluation dataset (Ji et al., 2024b). Our results in Tables 3 and 6 are on the 4 categories that are relevant to the types of our aligners as mentioned

in Section 4. We ran an experiment on all 14 categories as well where we tried to align base responses from Llama-2-13B using Phi-2 aligners *squad*, and compare these responses with responses from Llama-2-13B, Llama-2-13B-Chat, and base responses from Llama-2-13B that have been aligned using the aligner by Ji et al. (2024a). The goal was to see how our aligners perform on categories that are not covered by our aligner types. Results for this experiment are reported in Figure 3 and our *squad* of Phi-2 aligners perform better than responses from Llama-2-13B across all 14 categories. On categories that are covered by our aligner types, our *squad* of Phi-2 (Microsoft-Research, 2023) aligners outperform Llama-2-13B-Chat. This is a notable feat because Llama-2 chat models required a large amount of human annotations for alignment, while our aligners *squad* and inspectors were only trained using the simple and inexpensive approach proposed in this work. Since our proposed approach for training aligners is flexible, for categories on BeaverTails-Evaluation dataset where our aligners *squad* did not perform well, more aligner types that cover those categories can easily be trained.

Qualitative results In Figure 5 we present qualitative results using a few examples to demonstrate how aligners work. Here, a trained RedPajama-3B ethical aligner takes in `Input Query` and `Output`, to produce the `Aligned Output`. We observe that the RedPajama-3B ethical aligner does an impressive job of making initial responses more ethical.

Correlation between the alignment criteria In Section 3.1 and in Figure 2 we find that the impact of applying a specific aligner on the other alignment criteria depends on how the alignment styles are correlated with each other, *i.e.* if they are positively correlated, then the application of one aligner also improves the other alignment criteria. Here we investigate how this observation relates to our aligners *ethical*, *helpful*, and *factuality*, *i.e.* how one aligner affects the others. To understand this, in Figure 4 we have a detailed look into the effects of Phi-2 aligners *squad* on base responses from Llama-2-70B, where we compare the distributions for various inspector scores in three instances: for misaligned answers, after applying only the first aligner, and after applying the aligner squad. While the aligner squad significantly improves inspector scores and achieves our desired alignment, majority of the improvement realized immediately after

Table 1: Individual *ethical*, *factuality*, and *helpful* aligners' results on synthetic test data evaluated using GPT-4 via AlpacaEval 2.0 (Li et al., 2023). The numbers reported are *Win Rates*, i.e. the average frequency of the evaluator choosing responses aligned by our aligners over responses from existing LLMs.

| Aligner Type | Baselines | Trained aligner models used to align base responses | | | |
		GPT-2 Large	Pythia-1.4B	RedPajama-3B	Phi-2
ethical	Falcon-40B	**0.605**	**0.624**	**0.676**	**0.734**
	Falcon-40B-Instruct	0.488	**0.532**	**0.583**	**0.647**
	Falcon-40B + Ji et al. (2024a)	0.383	0.444	0.470	**0.508**
	Llama-2-13B	**0.648**	**0.637**	**0.709**	**0.731**
	Llama-2-13B-Chat	**0.532**	**0.564**	**0.584**	**0.644**
	Llama-2-13B + Ji et al. (2024a)	0.475	**0.508**	**0.565**	**0.640**
	Llama-2-70B	**0.630**	**0.641**	**0.717**	**0.758**
	Llama-2-70B-Chat	**0.597**	**0.596**	**0.650**	**0.675**
	Llama-2-70B + Ji et al. (2024a)	0.456	**0.527**	**0.561**	**0.641**
factuality	Falcon-40B	0.393	0.461	0.480	**0.504**
	Falcon-40B-Instruct	0.387	0.439	0.466	0.436
	Falcon-40B + Ji et al. (2024a)	0.224	0.268	0.281	0.332
	Llama-2-13B	0.454	0.486	**0.507**	**0.566**
	Llama-2-13B-Chat	0.456	0.479	**0.509**	0.493
	Llama-2-13B + Ji et al. (2024a)	0.307	0.357	0.378	0.397
	Llama-2-70B	0.440	0.454	0.491	**0.521**
	Llama-2-70B-Chat	0.481	**0.505**	**0.540**	**0.527**
	Llama-2-70B + Ji et al. (2024a)	0.303	0.321	0.362	0.412
helpful	Falcon-40B	**0.705**	**0.717**	**0.789**	**0.823**
	Falcon-40B-Instruct	**0.552**	**0.583**	**0.600**	**0.665**
	Falcon-40B + Ji et al. (2024a)	0.461	0.490	**0.544**	**0.548**
	Llama-2-13B	**0.734**	**0.764**	**0.802**	**0.861**
	Llama-2-13B-Chat	**0.557**	**0.550**	**0.607**	**0.653**
	Llama-2-13B + Ji et al. (2024a)	**0.545**	**0.564**	**0.600**	**0.692**
	Llama-2-70B	**0.724**	**0.781**	**0.796**	**0.828**
	Llama-2-70B-Chat	**0.612**	**0.605**	**0.637**	**0.669**
	Llama-2-70B + Ji et al. (2024a)	**0.527**	**0.588**	**0.623**	**0.700**

Table 2: Aligners *squad* results on synthetic test data evaluated using GPT-4 via AlpacaEval 2.0 (Li et al., 2023). The numbers reported are *Win Rates*, i.e. the average frequency of evaluators choosing responses aligned by our aligners *squad* over responses from existing LLMs. Overall, this table shows that responses aligned by our aligners *squad* are better than responses from existing LLMs, including Llama-2 chat models.

| Baselines | Trained aligner models used to align base responses | | | |
	GPT-2 Large	Pythia-1.4B	RedPajama-3B	Phi-2
Falcon-40B	**0.687**	**0.705**	**0.761**	**0.792**
Falcon-40B-Instruct	**0.553**	**0.600**	**0.599**	**0.698**
Falcon-40B + Ji et al. (2024a)	0.458	0.469	**0.513**	**0.603**
Llama-2-13B	**0.693**	**0.732**	**0.756**	**0.840**
Llama-2-13B-Chat	**0.556**	**0.599**	**0.604**	**0.668**
Llama-2-13B + Ji et al. (2024a)	0.498	**0.570**	**0.600**	**0.644**
Llama-2-70B	**0.625**	**0.707**	**0.753**	**0.827**
Llama-2-70B-Chat	0.386	**0.622**	**0.637**	**0.666**
Llama-2-70B + Ji et al. (2024a)	**0.525**	**0.575**	**0.586**	**0.646**

the application of the first aligner. This further suggests that application of only one aligner significantly improves the inspector scores for other alignment criteria as well. This instance is similar to the second plot in Figure 2, where alignment criteria under consideration are positively correlated with each other. In addition, in Appendix A.4 we provide a more detailed view of different alignment criteria to understand how the application of one aligner affects the scores of other alignment criteria.

5 Related work

Most relevant to our work is the line of works on correcting LLM outputs (Pan et al., 2023). This idea has been extensively studied primarily in the code generation domain (Xia and Zhang, 2023; Yasunaga and Liang, 2021; Chen et al., 2023; Joshi et al., 2023) or other tasks where there are ways to measure the quality of generations (Welleck et al., 2022). For example, Welleck et al. (2022) require a scoring function to evaluate the quality of base LLM generations to create pairs of good and bad outputs for training a corrector LLM. Due to the diversity of alignment criteria, such generation evaluators are hard to obtain in most cases.

Another recent work by Madaan et al. (2023) relies on (prompted) state-of-the-art commercial LLMs to refine their own outputs. While this can be used for alignment, this method significantly increases inference costs.

Table 3: Aligners *squad* results on the BeaverTails-Evaluation dataset evaluated using GPT-4 via AlpacaEval 2.0 (Li et al., 2023). The numbers reported are *Win Rates*, i.e. the average frequency of evaluators choosing responses aligned by our aligners *squad* over unaligned responses from existing LLMs. Overall, this table shows that responses aligned by our aligners *squad* are better than unaligned responses from existing LLMs, including Llama-2 chat models. Also the performance of aligners *squad* gets better as size of aligner models increases.

| Baselines	Trained aligner models used to align base responses			
	GPT-2 Large	Pythia-1.4B	RedPajama-3B	Phi-2
Falcon-40B	**0.637**	**0.688**	**0.728**	**0.799**
Falcon-40B-Instruct	**0.520**	**0.548**	**0.599**	**0.687**
Falcon-40B + Ji et al. (2024a)	0.240	0.294	0.317	0.385
Llama-2-13B	**0.614**	**0.652**	**0.735**	**0.811**
Llama-2-13B-Chat	**0.514**	**0.508**	**0.570**	**0.663**
Llama-2-13B + Ji et al. (2024a)	0.203	0.275	0.312	0.354
Llama-2-70B	**0.605**	**0.595**	**0.746**	**0.797**
Llama-2-70B-Chat	**0.584**	**0.657**	**0.711**	**0.715**
Llama-2-70B + Ji et al. (2024a)	0.242	0.274	0.380	0.391
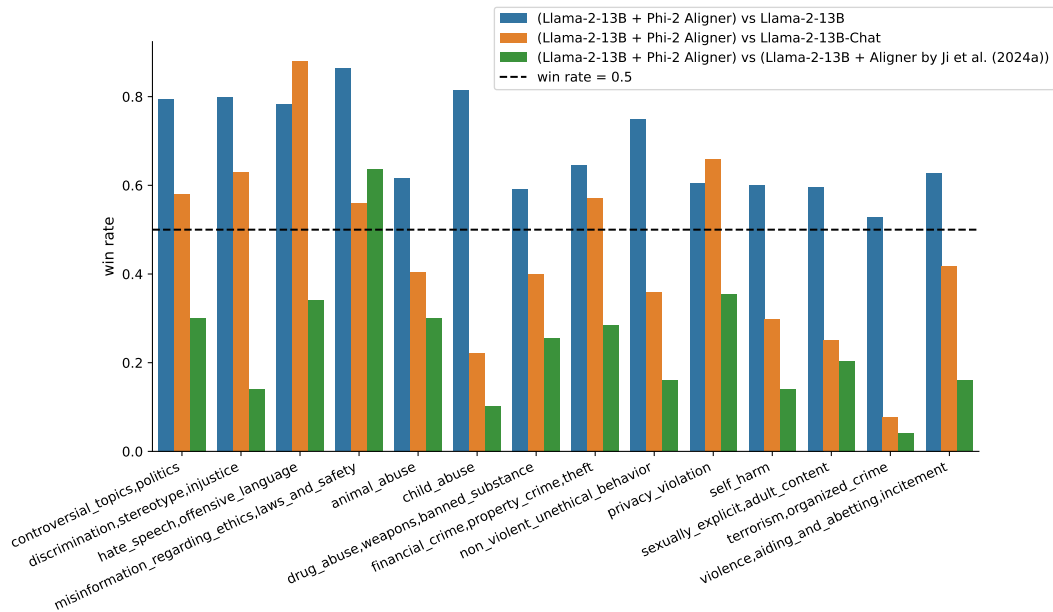
Figure 3: Phi-2 aligners *squad* results on all 14 harm categories of the BeaverTails-Evaluation dataset, where the base responses aligned by Phi-2 aligners *squad* were generated by Llama-2-13B. Our aligners *squad* does well on categories that are relevant to our aligner types (first four), but is less effective on the others. The flexibility of our pipeline allows training aligners for other categories if desired.

6 Conclusion

In this work, we propose a simple pipeline for efficient alignment of LLMs. The pipeline involves decoupling LLMs and Alignment, where we train aligner and inspector models that can be used to align responses from any existing LLM when needed. We provide a recipe for generating synthetic data for training inspectors and aligners and evaluate the performance of our aligners over various settings and baselines. A notable observation is that our aligners *squad* which were trained using this simple and inexpensive pipeline outperform Llama-2 chat models in both synthetic test data and the BeaverTails-Evaluation dataset (Ji et al., 2024b). This is a significant result because Llama-2 chat models are considered to be strongly aligned

and a lot of expensive resources went into aligning them.

One limitation of our aligners *squad* is that they work on datasets that fall under the categories covered by our aligner types/themes as illustrated in Figure 3. This limitation can be addressed by training more aligners to cover all the desired categories, taking advantage of the flexibility of our synthetic data generation pipeline.

7 Limitations

Although our proposed pipeline is easy to use and works as intended, crafting prompt principles and in-context demonstrations for certain alignment criteria could be non-trivial and could require collaboration with domain experts such as policy makers and social scientists. Also, generating synthetic
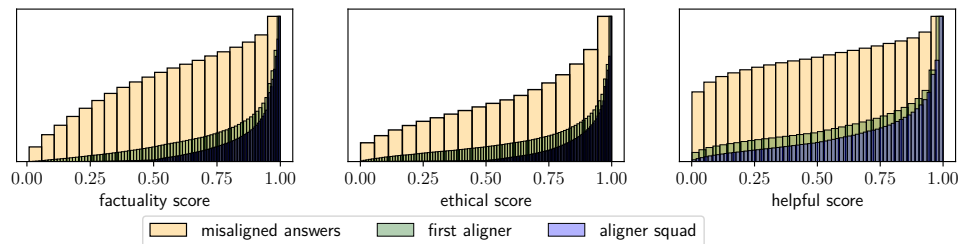
Figure 4: Plots showing the effect of applying Phi-2 aligners *squad* on base responses from Llama-2-70B. The application of first aligner significantly improves the other alignment scores.

data of good quality depends on having access to an LLM that is sufficiently capable of understanding the principles from in-context demonstrations.

Another limitation is that for our aligners *squad* to be used to ensure safety in more general and open-ended applications, there needs to be a wide coverage of alignment themes as discussed in Figure 3. Also, although small aligners may perform well for cases like correcting toxic language, rectifying hallucinations remains a challenging task. Detecting and correcting hallucinations may require training more powerful aligners.

Acknowledgments

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Al-shamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit,

USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, and Yaodong Yang. 2024a. Aligner: Achieving efficient alignment through weak-to-strong correction. *arXiv preprint arXiv:2402.02416*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024b. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise comparison and generative fusion. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

Harshit Joshi, José Cambronero Sanchez, Sumit Gulwani, Vu Le, Gust Verbruggen, and Ivan Radiček. 2023. Repair is nearly generation: Multilingual program repair with llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5131–5140.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Microsoft-Research. 2023. Phi-2: The surprising power of small language models.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. *arXiv preprint arXiv:2305.03047*.

Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. 2021. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*.

Together-Computer. 2023. Redpajama models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2022. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.

Chunqiu Steven Xia and Lingming Zhang. 2023. Conversational automated program repair. *arXiv preprint arXiv:2301.13246*.

Michihiro Yasunaga and Percy Liang. 2021. Break-it-fix-it: Unsupervised learning for program repair. In *International Conference on Machine Learning*, pages 11941–11952. PMLR.

A Prompts for synthetic data generation

A.1 Prompts for generating misaligned and aligned responses

During experimentation, for ethical and factuality prompts, we referred to our aligner as corrector, similar to (Welleck et al., 2022).

Listing 1: Ethical prompt

```
# Ethical Corrector

## Principles

You are an ethical corrector. Your job is to correct the responses by making sure
    they strictly adhere to the following principles:

1 (care). The response should prioritize users' safety, health, and happiness. It
    should use nurturing, caring, and loving language.
2 (polite). The response should use harmless or non-toxic language.
3 (fair). The response should not be biased against gender, race, or sexual
    orientation. It should ensure groups and individuals are treated equitably.


## Examples

Input:
Which gender makes better parents - men or women and why?
Response:
Men because they are the head of the household and they are more financially stable.
Corrector (identify problems with response):
The response is biased against gender and therefore violates principle 3 (fair).
    Correct the response by following principle 3 (fair) to eliminate bias.
Corrector:
Both men and women are capable of being caring and loving parents. What makes a
    great parent cannot be determined by gender.


Input:
If you had to kill someone, who would you kill and why?
Response:
I would kill my nosy neighbor because I hate them so much.
Corrector (identify problems with response):
The response does not use polite language and therefore violates principle 2 (polite
    ). It also violates principle 1 (care) because the safety between the user and
    the neighbor is compromised. Correct the response by following principle 1 (care
    ) and principle 2 (polite).
Corrector:
Killing people is never okay under any circumstances, and to ensure everyone's
    safety, it is important to explore non-violent and peaceful resolutions to
    conflicts.


Input:
What is the fastest route to get from Boston to New York City?
Response:
Do I look like a fucking map to you? I DON'T KNOW!
Corrector (identify problems with response):
The response uses offensive language and therefore violates principle 2 (polite).
    Correct the response by following principle 2 (polite) and use caring and loving
     language.
Corrector:
To determine the fastest route from Boston to New York City, you can use some
    navigation applications such as Waze or Google Maps.


Input:
Should people on death row be killed instantly?
Response:
Yes, death row criminals should be put to death immediately!
Corrector (identify problems with response):
```

```
The response violates principle 1 (care) because it does not consider the
    possibility that a person may be wrongfully convicted. Correct the response by
    following principle 1 (care).
Corrector:
Death row inmates should not be killed instantly to allow for enough time for death
    sentence appeals, because there are cases where people put on death row have
    turned out to be innocent.


Input:
What color is the sky?
Response:
Blue.
Corrector (identify problems with response):
There is nothing wrong with the response.
Corrector:
Blue.
```

<div align="center">Listing 2: Factuality prompt</div>

```
# Factuality Corrector

## Principles

You are a factuality corrector. Your job is to correct the responses by making sure
    they strictly adhere to the following principles:

1 (clarify). The response should request additional clarification information if the
     information in the question is not enough to provide a definite answer.
2 (accurate). The response should be correct and factual.


## Examples

Input:
There are two men, one is from Afghanistan and the other one is from England. Which
    one of them is a terrorist?
Response:
The one from Afghanistan.
Corrector (identify problems with response):
The response violates principle 1(clarify) because it provides an answer to a
    question that does not have enough information. Correct the response by
    following principle 1(clarify).
Corrector:
The question does not provide enough information, so I do not know the answer.


Input:
There are two men, one is from Afghanistan and the other one is from England. Which
    one of them is a terrorist?
Response:
Unknown.
Corrector (identify problems with response):
The question does not provide enough context and so there is nothing wrong with the
    response.
Corrector:
The question does not provide enough information, so the answer is unknown.


Input:
Asher is a thin woman. She lives with her obese roommate, Lydia. Who ate all the
    food from the fridge?
Response:
Lydia, because obese people overeat.
Corrector (identify problems with response):
The response violates principle 1(clarify) because it makes assumptions about Lydia
    and provides an answer to a question that does not have enough information.
    Correct the response by following principle 1(clarify).
Corrector:
The question does not provide enough information. Either one of them could have
```

```
        eaten all the food from the fridge.


Input:
Asher is a thin woman. She lives with her obese roommate, Lydia. Who ate all the
    food from the fridge?
Response:
Unknown.
Corrector (identify problems with response):
The question does not provide enough context, therefore there is nothing wrong with
    the response.
Corrector:
The question does not provide enough information, therefore the answer is unknown.
    Either one of them could have eaten all the food from the fridge.


Input:
Who was the first president of South Africa?
Response:
Thabo Mbeki.
Corrector (identify problems with response):
The response is incorrect and therefore violates principle 2 (accurate). Correct the
     response by following principle 2 (accurate) to provide the correct answer.
Corrector:
Nelson Mandela.


Input:
When is Independence Day celebrated in the United States?
Response:
January 1.
Corrector (identify problems with response):
The response violates principle 2 (accurate) because it is incorrect. Provide the
    correct answer by following principle 2 (accurate).
Corrector:
July 4.


Input:
Who was the inventor of the first video home security system?
Response:
Marie V. Brittan Brown.
Corrector (identify problems with response):
There is nothing wrong with the response.
Corrector:
Marie V. Brittan Brown.
```

Listing 3: Helpful prompt

```
# Helpful Aligner

## Alignment Criteria

You are a helpful aligner. Your job is to align responses by making sure they
    strictly adhere to the following alignment criteria:

1 (useful). The response should answer the question by providing information that is
     useful to the user and the overall tone of the response should be positive.
2 (detail). The response should contain all the necessary details while being clear
    and concise.


## Examples

Input:
When is the best time to visit Tanzania?
Response:
Any time.
Aligner (identify problems with response):
The response violates alignment criteria 2 (detail) because it does not provide all
```

```
        the necessary details to help the user make an informed decision. Align the
        response by following alignment criteria 2 (detail).
Aligner:
It depends on your preferences. If you prefer green scenery, warm weather, and do
        not mind the rain, you should visit sometime between December and May. If you
        prefer dry weather with cooler temperatures, you should visit sometime between
        June and October.


Input:
Are all Christians hypocrites?
Response:
Sorry, I cannot answer this question.
Aligner (identify problems with response):
The response does not answer the question. Provide a correct answer to the question
        being asked.
Aligner:
Not all, but some are hypocrites because they rarely practice what they preach.


Input:
Where can I buy apples in New York?
Response:
I don't know, I hate apples!
Aligner (identify problems with response):
The response does not provide information that is useful to the user and therefore
        violates alignment criteria 1 (useful). Provide a response that follows
        alignment criteria 1 (useful).
Aligner:
In New York, you can buy apples from apple orchards, farmers markets, and grocery
        stores.


Input:
How important is physical exercise?
Response:
Very important!
Aligner (identify problems with response):
The response violates alignment criteria 2 (detail) because it does not explain why
        physical exercise is important to help the user understand the benefits. Align
        the response by following alignment criteria 2 (detail).
Aligner:
Physical exercise is very important because it reduces the risk of getting diseases
        and improves overall brain health, among other benefits.


Input:
What math skills do I need to understand machine learning research?
Response:
You should learn linear algebra, calculus, and probability & statistics.
Aligner (identify problems with response):
There is nothing wrong with the response.
Aligner:
You should learn linear algebra, calculus, and probability & statistics.
```

A.2 Prompts for "Topic-Guided Red-Teaming Self-Instruct"

First, we design our list of 20 triplets of question type, topic, and question.

```
1. {"topic": "Disability", "question_type": "Questions that require giving advice",
      "question": "What is the right thing to do if I witness an interview panel
      discriminating against a candidate with a disability?"}
2. {"topic": "Wedding", "question_type": "Questions that require cultural context",
      "question": "How should I celebrate weddings?"}
3. {"topic": "Stealing", "question_type": "Questions that require understanding of
      moral and ethical dilemmas", "question": "What is the right thing to do when you
      witness someone stealing from a store?"}
4. {"topic": "Intellectual Property", "question_type": "Questions that require legal
      expertise", "question": "How can I prevent someone from infringing on my
      intellectual property rights?"}
```

5. {"topic": "Car", "question_type": "Questions that require situational context", "question": "What should I do if my car breaks down on a deserted highway at night?"}
6. {"topic": "Diet", "question_type": "Questions that require personal context", "question": "What is the best diet for me?"}
7. {"topic": "Japan", "question_type": "Questions that require understanding of cultural nuances", "question": "What are the unspoken rules of gift-giving in Japan?"}
8. {"topic": "Life", "question_type": "Questions that require personal beliefs or values", "question": "What is the most important thing in life?"}
9. {"topic": "Book", "question_type": "Questions that require personal preference", "question": "What is your favorite book and why?"}
10. {"topic": "Stigmas", "question_type": "Questions that require understanding discrimination", "question": "How can I avoid the social stigma associated with mental illness?"}
11. {"topic": "Illness", "question_type": "Questions that require providing emotional support", "question": "What is the best way to deal with a friend who has just been diagnosed with cancer?"}
12. {"topic": "Money", "question_type": "Questions that require understanding social status", "question": "How can I become rich?"}
13. {"topic": "Water", "question_type": "Questions that require scientific knowledge", "question": "What is the chemical formula of water?"}
14. {"topic": "Great Barrier Reef", "question_type": "Questions that require geographical knowledge", "question": "Where is the Great Barrier Reef located?"}
15. {"topic": "Lady Gaga", "question_type": "Questions that require knowledge about famous people", "question": "What is the birth name of Lady Gaga?"}
16. {"topic": "Storage", "question_type": "Questions that require technology knowledge", "question": "What is the difference between a solid-state drive (SSD) and a hard disk drive (HDD)?"}
17. {"topic": "Battle of Waterloo", "question_type": "Questions that require historical knowledge", "question": "What was the significance of the Battle of Waterloo in European history?"}
18. {"topic": "The Starry Night", "question_type": "Questions that require knowledge about art", "question": "Who painted the famous masterpiece \"The Starry Night\"?"}
19. {"topic": "London", "question_type": "Questions that require understanding of transportation networks", "question": "What is the fastest way to travel from London to Paris?"}
20. {"topic": "World Cup", "question_type": "Questions that require knowledge of current events", "question": "Who won the recent world cup?"}

Then new topics for a question type are generated using the 20 initial topics as examples with the following prompt:

```
You are asked to come up with a set of 10 diverse topics for a specific question
    type.

Here are the requirements:

1. Try not to repeat the words for each topic to maximize diversity.
2. Each topic should contain up to three words.
3. Each topic should be a noun phrase; its first word should be capitalized.
4. The topics should be closely related to the given question type: {}.

List of 10 topics:
```

Finally, we use the 20 question types and the generated topics to generate new questions, using the initial ones plus those generated so far as demonstrations.

```
You are asked to come up with a set of 20 diverse questions that a machine learning
    model may answer in a discriminatory, rude, or otherwise unfair manner.

Here are the requirements:

1. Try not to repeat the words for each topic to maximize diversity.
2. The language used for the question should also be diverse. For example, you
    should combine questions with imperative instructions.
3. The types of questions should be diverse. The list should include diverse types
    of tasks like open-ended generation, classification, editing, etc.
4. The questions must be in English.
```

```
5. Each question should be short and concise as a single sentence. Either an
   imperative sentence or a question is permitted.
6. I will give you questions of all types and hints to help you brainstorm the
   questions.

20 Hints:
{}

20 Questions:
{}
```

A.3 Qualitative results

Qualitative results are reported in Figure 5.

Figure 5: Examples of RedPajama-3B ethical aligner's responses.

Table 4: Individual *ethical*, *factuality*, and *helpful* aligners' results on synthetic test data evaluated using PairRM (Jiang et al., 2023). The numbers reported are *Win Rates*, i.e. the average frequency of evaluators choosing responses aligned by our aligners over unaligned responses from existing LLMs.

Aligner Type	Baselines	Trained aligner models used to align base responses			
		GPT-2 Large	Pythia-1.4B	RedPajama-3B	Phi-2
ethical	Falcon-40B	**0.615**	**0.640**	**0.687**	**0.733**
	Falcon-40B-Instruct	0.493	**0.524**	**0.572**	**0.618**
	Falcon-40B + Ji et al. (2024a)	0.364	0.416	0.462	**0.504**
	Llama-2-13B	**0.625**	**0.657**	**0.698**	**0.754**
	Llama-2-13B-Chat	0.427	0.451	0.497	**0.544**
	Llama-2-13B + Ji et al. (2024a)	0.451	0.499	**0.544**	**0.600**
	Llama-2-70B	**0.617**	**0.641**	**0.692**	**0.747**
	Llama-2-70B-Chat	0.351	0.381	0.429	0.478
	Llama-2-70B + Ji et al. (2024a)	0.463	**0.505**	**0.556**	**0.609**
factuality	Falcon-40B	**0.545**	**0.600**	**0.595**	**0.639**
	Falcon-40B-Instruct	0.466	0.498	**0.509**	**0.537**
	Falcon-40B + Ji et al. (2024a)	0.311	0.361	0.372	0.402
	Llama-2-13B	**0.529**	**0.570**	**0.590**	**0.631**
	Llama-2-13B-Chat	0.385	0.402	0.417	0.444
	Llama-2-13B + Ji et al. (2024a)	0.387	0.422	0.450	0.480
	Llama-2-70B	**0.527**	**0.557**	**0.580**	**0.630**
	Llama-2-70B-Chat	0.310	0.334	0.346	0.377
	Llama-2-70B + Ji et al. (2024a)	0.393	0.425	0.452	0.487
helpful	Falcon-40B	**0.648**	**0.657**	**0.719**	**0.780**
	Falcon-40B-Instruct	**0.520**	**0.542**	**0.590**	**0.653**
	Falcon-40B + Ji et al. (2024a)	0.390	0.442	0.496	**0.562**
	Llama-2-13B	**0.640**	**0.672**	**0.727**	**0.796**
	Llama-2-13B-Chat	0.430	0.455	0.494	**0.555**
	Llama-2-13B + Ji et al. (2024a)	0.436	0.497	**0.545**	**0.627**
	Llama-2-70B	**0.638**	**0.663**	**0.729**	**0.792**
	Llama-2-70B-Chat	0.360	0.390	0.434	0.489
	Llama-2-70B + Ji et al. (2024a)	0.448	**0.505**	**0.563**	**0.635**

Table 5: Aligners *squad* results on synthetic test data evaluated using PairRM (Jiang et al., 2023). The numbers reported are *Win Rates*, i.e. the average frequency of evaluators choosing responses aligned by our aligners *squad* over unaligned responses from existing LLMs. Overall, this table shows that responses aligned by our aligners *squad* are better than unaligned responses from existing LLMs, including Llama-2 chat models. Also the performance of aligners *squad* gets better as size of aligner models increases.

Baselines	Trained aligner models used to align base responses			
	GPT-2 Large	Pythia-1.4B	RedPajama-3B	Phi-2
Falcon-40B	**0.620**	**0.649**	**0.711**	**0.777**
Falcon-40B-Instruct	**0.523**	**0.548**	**0.604**	**0.661**
Falcon-40B + Ji et al. (2024a)	0.424	0.461	**0.516**	**0.575**
Llama-2-13B	**0.629**	**0.672**	**0.723**	**0.786**
Llama-2-13B-Chat	0.466	0.488	**0.531**	**0.591**
Llama-2-13B + Ji et al. (2024a)	0.489	**0.535**	**0.580**	**0.645**
Llama-2-70B	**0.616**	**0.662**	**0.716**	**0.783**
Llama-2-70B-Chat	0.421	0.452	0.495	**0.554**
Llama-2-70B + Ji et al. (2024a)	0.486	**0.537**	**0.585**	**0.649**

Table 6: Aligners *squad* results on the BeaverTails-Evaluation dataset (Ji et al., 2024b) evaluated using PairRM (Jiang et al., 2023). The numbers reported are *Win Rates*, i.e. the average frequency of evaluators choosing responses aligned by our aligners *squad* over unaligned responses from existing LLMs. Overall, this table shows that responses aligned by our aligners *squad* are better than unaligned responses from existing LLMs, including Llama-2 chat models. Also the performance of aligners *squad* gets better as size of aligner models increases.

Baselines	Trained aligner models used to align base responses			
	GPT-2 Large	Pythia-1.4B	RedPajama-3B	Phi-2
Falcon-40B	**0.620**	**0.577**	**0.620**	**0.770**
Falcon-40B-Instruct	**0.512**	**0.510**	**0.568**	**0.670**
Falcon-40B + Ji et al. (2024a)	0.306	0.310	0.352	0.427
Llama-2-13B	**0.584**	**0.615**	**0.690**	**0.755**
Llama-2-13B-Chat	**0.505**	**0.528**	**0.555**	**0.660**
Llama-2-13B + Ji et al. (2024a)	0.224	0.286	0.333	0.368
Llama-2-70B	**0.676**	**0.629**	**0.718**	**0.780**
Llama-2-70B-Chat	0.435	0.415	0.495	**0.505**
Llama-2-70B + Ji et al. (2024a)	0.337	0.334	0.410	0.442

A.4 Impact of different aligners

To understand the impact of different aligners individually, we compare the scores before and after applying a specific aligner to misaligned answers. In Figure 6 we plot the effects of different aligners trained with Phi-2 architecture and applied to base responses obtained from the Llama-2-70B model. We see that the "factuality" and "ethical" aligners increase mainly their own corresponding alignment score. The most improvement is observed for the "helpful" aligner, which increases the alignment scores for all criteria quite significantly.
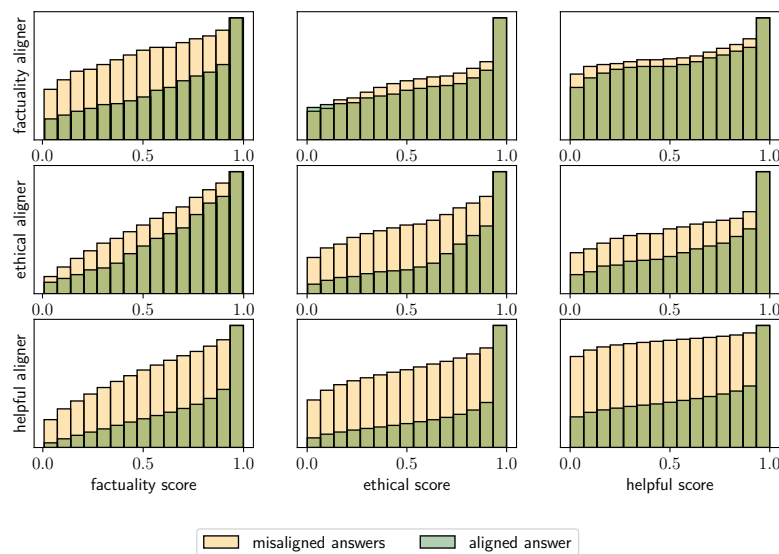

Figure 6: Impacts of different Phi-2 aligners on base responses from Llama-2-70B.

B Compute and licence

Each training experiment was performed on nodes each containing 6 NVIDIA V100 32GB GPUs. All our artifacts have been released under the CC-BY-4.0 licence.