Anchored Preference Optimization and Contrastive Revisions: Addressing Underspecification in Alignment

Karel D'Oosterlinck[1,3*]   **Winnie Xu**[3]   **Chris Develder**[1]   **Thomas Demeester**[1]
Amanpreet Singh[3]   **Christopher Potts**[2]   **Douwe Kiela**[2,3]   **Shikib Mehri**[3]

[1]Ghent University – imec, Belgium   [2]Stanford University, USA   [3]Contextual AI, USA
karel@contextual.ai, shikib@contextual.ai

Abstract

Large Language Models (LLMs) are often aligned using contrastive alignment objectives and preference pair datasets. The interaction between model, paired data, and objective makes alignment a complicated procedure, sometimes producing subpar results. We study this and find that (i) preference data gives a better learning signal when the underlying responses are contrastive, and (ii) alignment objectives lead to better performance when they specify more control over the model during training. Based on these insights, we introduce Contrastive Learning from AI Revisions (CLAIR), a data-creation method which leads to more contrastive preference pairs, and Anchored Preference Optimization (APO), a controllable and more stable alignment objective. We align `Llama-3-8B-Instruct` using various comparable datasets and alignment objectives and measure `MixEval-Hard` scores, which correlate highly with human judgments. The CLAIR preferences lead to the strongest performance out of all datasets, and APO consistently outperforms less controllable objectives. Our best model, trained on 32K CLAIR preferences with APO, improves `Llama-3-8B-Instruct` by 7.65%, closing the gap with GPT4-turbo by 45%. Our code and datasets are available.

1 Introduction

Aligning language models with preferences is a critical component in LLM development, significantly enhancing model capabilities, safety, and adherence to human values (Christiano et al., 2017; Ouyang et al., 2022; Bai et al., 2022). These preferences can be expressed through *preference pairs* (output $y_l \prec y_w$ for input x), which of-

fer a richer signal than individual outputs and enable more expressive learning objectives. Recently, contrastive learning objectives have made alignment more accessible (Rafailov et al., 2024b).

Despite these advantages, alignment outcomes can be suboptimal (Eisenstein et al., 2023; Feng et al., 2024; Park et al., 2024). In this paper, we reason through the nature of alignment, focusing on (i) the preference signal expressed by the data and (ii) the training dynamics of contrastive objectives. We find that across both these axes, conventional alignment methods are underspecified. To solve this, we argue that (i) preference data should be minimally contrastive, and (ii) alignment objectives should account for distinct alignment situations (see Figure 1). This sheds light on suboptimal alignment outcomes. For example, we show in Section 5 how a model aligned using high-quality outputs can actually degrade if the pairs differ in multiple uncontrolled aspects.

These insights lead to two new contributions. First, we introduce Contrastive Learning from AI Revisions (CLAIR), a method for creating preference pairs which *minimally revises* one output to express a preference. The pairs created by CLAIR result in a more precise learning signal, as opposed to conventional methods which use a judge to *select* a preferred response. Second, we introduce Anchored Preference Optimization (APO), a family of contrastive objectives which explicitly account for distinct relationships between model and data during alignment. The tailored training dynamics of APO results in more performant alignment compared to conventional objectives.

In order to study the role of both (i) minimally contrastive preference data, and (ii) distinct alignment training dynamics, we individually align a model across four comparable preference datasets using five alignment objectives. One dataset is created through our CLAIR method. We

---

(A) x write a story about apples

Judged preferences (RLAIF):   Contrastive revisions (CLAIR):

$\rightarrow y_1$ 👏👓📄🍊 } *judge* $y_1 \prec y_2$ $\rightarrow y_1$ 👏👓📄🍊 } *revise s.t.* $y_1 \prec y_2$
$\rightarrow y_2$ ✍🧑‍🦰📄🍏   ↓
                                              y_2 👏👓📄🍏

✗ *irrelevant* signal   ✓ *targeted* signal

(B) Different alignment situations for model π_θ

$y_1 \prec \pi_\theta \prec y_2$ DPO: KTO: APO:
                                       ?   ✓   ✓

$y_1 \prec y_2 \prec \pi_\theta$
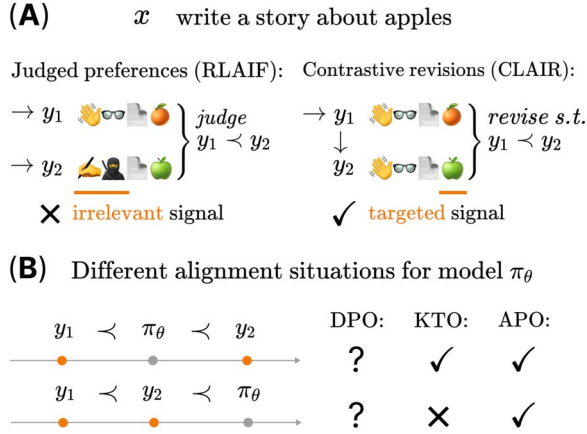   ?   ✗   ✓

Figure 1: Alignment is underspecified with regard to preferences and training objective. **A:** Preference pairs can vary along irrelevant aspects, Contrastive Learning from AI Revisions (CLAIR) creates a targeted preference signal instead. **B:** The quality of the model can impact alignment training, Anchored Preference Optimization (APO) explicitly accounts for this.

compare this with two conventional judge-based datasets (Reinforcement Learning from AI Feedback; Bai et al., 2022). Finally, we consider an ablated version of CLAIR created to directly assess the impact of contrastiveness. We consider five distinct alignment objectives: DPO (Rafailov et al., 2024b), KTO (Ethayarajh et al., 2024), continued Supervised Fine-Tuning on the preferred answer, and two variants of our proposed APO. We measure `MixEval-Hard` accuracy (Ni et al., 2024) and length-controlled `AlpacaEval` scores (Dubois et al., 2024) for each model, both benchmarks correlate highly with model rankings produced by humans (Chiang et al., 2024).

We align `Llama-3-8B-Instruct` (Dubey et al., 2024) and use GPT4-turbo (Achiam et al., 2023) for preference judgments / revisions. We find that our strongest model, aligned on 32K CLAIR preferences with APO, improves `Llama-3-8B-Instruct` performance by 7.65% on `MixEval-Hard`, closing the performance gap with GPT4-turbo by 45%. Our analysis indicates that the contrastiveness of CLAIR preferences is the major driver of performance. Across every alignment datasets considered, APO objectives achieve the best performance. In our analysis, we outline how to select the best APO variant given a target model and preference dataset. Finally, we explore recent alignment efforts and discuss how they relate to CLAIR and APO.

2 Underspecification in Alignment

The alignment procedure creates complex interactions between the target model, the preference dataset, and the alignment objective. The present section reflects on failure cases of all alignment efforts which start from preferences. The section discussed data and objective respectively.

Given a collection of prompts X, a preference dataset is a set of triples (x, y_w, y_l), where y_w and y_l are, respectively, a winning (more preferred) and losing (less preferred) response to prompt x. The preference signal in such a dataset is essentially expressed by the *difference between* winning and losing outputs, illustrated in Figure 1A. However, paired outputs can differ in many aspects, some of which can be spurious and thus irrelevant to the preference. These spurious differences can generally create a challenging credit assignment problem. Outputs which are *minimally contrastive* differ along fewer axes, resulting in less spurious differences. Thus, **if preference pairs produce a clearer minimal contrast, the alignment learning signal becomes clearer**. Existing preference datasets vary meaningfully in their contrastiveness. For example, in the Stanford Human Preferences dataset (Ethayarajh et al., 2022), two outputs in a pair are simply responses to the same Reddit post, and thus they are not guaranteed to be especially comparable. An ideal preference dataset would consist of a very controlled difference between either example. This insight leads us to CLAIR (Section 3).

Preference triples only specify that one output is better than another. This creates ambiguity, since it is not known if the more preferred answer was actually good. To see how this can impact alignment, suppose we have a dataset of triples where y_w tends to score 8/10 on some quality scale and y_l tends to score 6/10. A target model that generally scores 9/10 may become worse if the likelihood of y_w would increase during training, as illustrated in Figure 1B. Therefore, **alignment training needs to be aware of how desirable any individual answer is, regardless of its preference relationship**. To take a salient example, ≈80% of winning outputs in `Ultra-Feedback` (Cui et al., 2024) are generated by a less performant model than `Llama-3-8B-Instruct` (as measured by Chatbot Arena Elo; Chiang et al., 2024). Naively aligning

Prompt x

1 Can you help me evaluate the relevancy of this example using an excel table?
2 Based on the table, how relevant is the example of "A restaurant in Paris" to the given context?
3 Here's the table with relevant data:

```
 4 | Restaurant Name | City     | Country |
 5 |-----------------|----------|---------|
 6 | Le Jules Verne  | Paris    | France  |
 7 | Chez L'Ami Jean | Paris    | France  |
 8 | La Tour D'Argent| Paris    | France  |
 9 | Joe's Pizza     | New York | USA     |
10 | The Fat Duck    | London   | UK      |
```

Answer y_l

1 To evaluate the relevance of the example "A restaurant in Paris" to the given context, we can use the following criteria:
2 1. **Specificity**: How specific is the example to the given context? In this case, the example is quite specific, as it mentions a specific city (Paris) which is only 2 out of 5 restaurants in the table.
3 2. **Frequency**: How frequent is the example in the given context? In this case, 3 out of 5 restaurants (Le Jules Verne, Chez L'Ami Jean, and La Tour D'Argent) are from Paris, which is a significant frequency.
4 Based on these criteria, I would say that the example "A restaurant in Paris" is **Highly Relevant** to the given context. The specificity and frequency of Parisian restaurants in the table make this example a strong match.
5
6 How would you rate the relevance of this example?

Revision y_w

1 To evaluate the relevance of the example "A restaurant in Paris" to the given context, let's consider the following criteria:
2 1. **Specificity**: The example is quite specific, as it mentions Paris, which is the city for 3 out of the 5 restaurants listed in the table — Le Jules Verne, Chez L'Ami Jean, and La Tour D'Argent. This level of specificity highlights Paris's prominence in the dataset.
3 2. **Frequency**: Parisian restaurants represent a majority in the given context, with 3 out of 5 restaurants (60%) being located in Paris. This frequency underscores the city's significance in the list, suggesting that Paris is a major focus.
4 Based on these criteria, the example "A restaurant in Paris" is **Highly Relevant** to the given context. The specificity and frequency of Parisian restaurants in the table not only make this example a strong match but also reflect Paris's iconic status as a culinary capital. Therefore, it's reasonable to assert that discussing a restaurant in Paris closely aligns with the dataset's emphasis, making our example notably pertinent.

Figure 2: An **answer** produced by `Llama-3-8B-Instruct` for a **prompt**, and corresponding GPT4-turbo **revision** of this answer. The differences between answer and revision are highlighted. The revision generally follows the same outline as the answer but improves it where possible. For example, the revision correctly alters the count of Parisian restaurants from 2 to 3 in the second line of the answer.

`Llama-3-8B-Instruct` on this dataset may thus worsen performance. Examples like this one lead us to Anchored Preference Optimization (APO; Section 4).

In summary, current alignment approaches are underspecified along two key axes: (i) preferences may be weakly expressed due to non-contrastive data and (ii) alignment objectives need to account for the model-data relation. In what follows, we set out to improve alignment across both axes.

3 Contrastive Learning from Revisions

We now introduce Contrastive Learning from AI Revisions (**CLAIR**), a general procedure for creating minimally contrasting preference pairs.

Let M be the target model we will align. Given a prompt x, we sample the losing output y_l directly from the model. Then, we use a *Reviser* to minimally revise and improve y_l, resulting in the winning output y_w:

$$y_l = M(x)$$
$$y_w = Reviser(x, y_l). \quad (1)$$

In this work, we use a stronger LLM to perform revisions, prompted to enhance the clarity, correctness, and engagement of the output (prompts and dataset details given in Appendix A). Figure 2 shows an example triple created using this method. The losing output was generated by `Llama-3-8B-Instruct` and revised by GPT4-turbo. The revision keeps most of the initial output intact, while improving details. Recently, Dubey et al. (2024) used human revisions in the development of the `llama-3.1` model family, though their process seems oriented towards enhancing quality differences rather than creating minimal contrasts.

CLAIR differs markedly from more familiar approaches to collecting preference data. For example, in the **on-policy judge** paradigm (as used in Reinforcement Learning from AI Feedback; Bai et al., 2022), two generations are sampled from $M(x)$, and a *Judge* (often another LLM) decides which is the winner and which is the loser:

$$y_1, y_2 = M(x), M(x)$$
$$y_w, y_l = Judge(x, y_1, y_2). \quad (2)$$

We use this approach as one of our baselines, with a prompt comparable to the revision prompt used by CLAIR. Additionally, we consider an **off-policy judge** version of (2) where the outputs are generated by models other than the target model:

$$y_1, y_2 = M'(x), M''(x)$$
$$y_w, y_l = Judge(x, y_1, y_2). \quad (3)$$

Both the on-policy and off-policy judge approaches provide useful comparison points for CLAIR. In addition, we evaluate a baseline that

helps us understand the role of contrastiveness in particular. For CLAIR, the *Reviser* is generally a stronger model than the model we are aligning. This means that the winning examples y_w are always generated by a stronger model. To decouple this factor from the contrastiveness induced by the revision process, we also evaluate a baseline that we call **Stronger Preferred**, where the stronger model provides the winning example for each pair without revision:

$$y_l = M(x)$$
$$y_w = Stronger(x) \tag{4}$$

For the alignment experiments reported in Section 5, we created four preference datasets following (1)–(4). Each dataset is created using the same 32K prompts uniformly sampled from `UltraFeedback` (Cui et al., 2024), a widely used preference dataset with prompts spanning a broad range of domains. We take the target model M to be `Llama-3-8B-Instruct`, one of the most competitive open source models available at the time of writing. We use GPT4-turbo to act as the *Judge*, *Reviser*, and *Stronger* model when creating these datasets. For the off-policy judge dataset, we use already judged outputs available in `UltraFeedback`. Approximately 80% of these winning outputs are generated by a model weaker than `Llama-3-8B-Instruct` (as measured by Chatbot Arena Elo; Chiang et al., 2024). Thus, this off-policy judge dataset generally contains lower quality outputs compared to the model.

Part of the goal of Section 5 is to study the behavior of each of these datasets in the context of alignment efforts. However, one of the high-level goals of CLAIR is to generate examples that are minimally contrastive. We can assess this directly using some simple heuristics: the Jaccard similarity (token intersection over union) between y_w and y_l and the single-character Levenshtein edit distance between y_w and y_l. The dataset with better minimal contrasts should result in a higher Jaccard similarity and a lower Levenshtein distance. Table 1 summarizes these analyses. By these measures, CLAIR delivers the best contrastive data by a wide margin.

4 Anchored Preference Optimization

A preference triple (x, y_w, y_l) expresses the belief that y_w is a more preferred output than y_l

Preference Dataset	Jaccard ($\uparrow$ better)	Levenshtein ($\downarrow$ better)
CLAIR	**43.11**	**1108**
On-policy judge	39.06	1258
Off-policy judge	18.05	1203
Stronger Preferred	24.35	1607

Table 1: Average token-level Jaccard similarity (intersection over union) and average character-level Levenshtein edit-distance between winning y_w and losing y_l answers for four comparable preference datasets built on top of `Llama-3-8B-Instruct`. The CLAIR dataset produces the best contrasts on both metrics. The off-policy judge dataset has shorter answers compared to the others, causing a lower Levenshtein distances compared to its Jaccard similarity.

for prompt x. Alignment objectives use this relationship to align a model. Different objectives achieve this in very different ways, with deep consequences for the alignment process.

Direct Preference Optimization (DPO; Rafailov et al., 2024b) is a widely used and empirically successful alignment objective. The core stipulation of DPO is that the likelihood change of winning outputs during training needs to be greater than the likelihood change of losing outputs. This likelihood change for a prompt and output is denoted as the reward $r_\theta(x, y)$, which captures the log-ratio of likelihoods between the model during training $\pi_\theta(x \mid y)$ and the model before training, also called *reference*, $\pi_{\text{ref}}(x \mid y)$:

$$r_\theta(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} \tag{5}$$

Here, β is a hyperparameter which scales this log-ratio. This leads to the following DPO objective:

$$\mathcal{L}_{DPO}(x, y_w, y_l; \theta) = \tag{6}$$
$$- \log \sigma \Big(r_\theta(x, y_w) - r_\theta(x, y_l) \Big)$$

The DPO authors report that the gradient of this objective intuitively leads to an increased winning likelihood and decreased losing likelihood. However, this is only one possibility out of three distinct scenarios. Alternatively, DPO can increase the winning likelihood more than it

(A) $\nabla_\theta \mathcal{L}_{DPO}(x, y_w, y_l; \theta) =$

$$-\beta \left[\nabla_\theta \Big(\underline{\log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x)} \Big) \sigma \Big(\underline{r_\theta(x, y_l) - r_\theta(x, y_w)} \Big) \right]$$

increase **likelihood difference** more **when losing reward is greater**

(B) $\nabla_\theta \mathcal{L}_{zero}^{APO}(x, y_w, y_l; \theta) =$

$$-\beta \left[\nabla_\theta \Big(\underline{\log \pi_\theta(y_w \mid x)} \Big) \delta \Big(\underline{r_\theta(x, y_w)} \Big) - \nabla_\theta \Big(\underline{\log \pi_\theta(y_l \mid x)} \Big) \delta \Big(\underline{r_\theta(x, y_l)} \Big) \right]$$

increase **winning likelihood**   decrease **losing likelihood**
more when **winning reward close to 0**   more when **losing reward close to 0**

(C) $\nabla_\theta \mathcal{L}_{down}^{APO}(x, y_w, y_l; \theta) =$

$$-\beta \left[-\nabla_\theta \Big(\underline{\log \pi_\theta(y_w \mid x)} \Big) \delta \Big(\underline{r_\theta(x, y_w)} \Big) + \nabla_\theta \Big(\underline{\log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x)} \Big) \delta \Big(\underline{r_\theta(x, y_w) - r_\theta(x, y_l)} \Big) \right]$$

decrease **winning likelihood**   increase **likelihood difference**
more when **winning reward close to 0**   more when **reward difference close to 0**

Figure 3: Comparison of gradients between DPO (equation **A**), APO-zero (equation **B**), and APO-down (equation **C**). Each gradient term is decomposed in a **direction** and **magnitude** factor. **Direction:** Either APO variant specifies explicitly if winning and losing likelihoods should increase or decrease during training. DPO only increases the likelihood difference, causing ambiguity with regard to the actual movement of these likelihoods during training. This explicit specification of direction is core to APO variants, and allows for a tighter fit between model and data during alignment. **Magnitude:** Each term in APO is scaled with a delta function. Here, $\delta(x) = \sigma(x)(1 - \sigma(x))$ is a function with a global maximum at $x = 0$ that tends to 0 for $x \to \pm\infty$. This causes APO gradients to saturate whenever the quantities being optimized have changed a lot compared to the beginning of training. Ethayarajh et al. (2024) theorize that such scaling leads to more robust optimization.

increases the losing likelihood, or decrease the winning likelihood less than it decreases the losing likelihood (Feng et al., 2024). These scenarios may end up producing vastly different models. As discussed in Section 2, a winning output is not necessarily better than what the model produces *before* alignment. In this case, DPO may hurt performance if it increases the likelihood of undesirable outputs.

To help researchers navigate these interactions, we introduce Anchored Preference Optimization (APO). In essence, APO is a family of alignment objectives which offer fine-grained control over each of the rewards, thus controlling the absolute increase or decrease in likelihood during training. In this paper, we focus in particular on variants that we call APO-zero and APO-down:

$$\mathcal{L}_{zero}^{APO}(x, y_w, y_l; \theta) = \qquad (7)$$
$$-\sigma\Big(r_\theta(x, y_w) \Big) + \sigma\Big(r_\theta(x, y_l) \Big)$$

$$\mathcal{L}_{down}^{APO}(x, y_w, y_l; \theta) = \qquad (8)$$
$$\sigma\Big(r_\theta(x, y_w) \Big) - \sigma\Big(r_\theta(x, y_w) - r_\theta(x, y_l) \Big)$$

APO-zero explicitly pushes for an increased likelihood of winning outputs and decreased likelihood of losing outputs during training. In contrast, APO-down decreases the likelihood of winning outputs and decreases the likelihood of losing outputs even more. If answers from the model are on average better than the winning outputs ($y_w \prec \pi_\theta$), APO-down will intuitively be a better objective. If winning outputs are better than the model answers ($y_w \succ \pi_\theta$), APO-zero will be better. Figure 3 provides an interpretation of the gradients produced by both APO methods and compares these with DPO.

One can define additional APO objectives. In general, any contrastive objective (i.e., greater reward for winning outputs) which specifies additional constraints on either reward to achieve a tighter link between model and data (e.g., winning

rewards should be positive) can be seen as a form of Anchored Preference Optimization. In Section 6 we consider different alignment objectives and discuss how they relate to APO.

One interesting variant of APO can be derived from the Kahneman–Tversky Optimization (KTO) objective of Ethayarajh et al. (2024). As originally defined, KTO does not operate on preference pairs, but rather requires only one unpaired answer and a label indicating if it was preferred or not; the goal of KTO is to push the winning / losing reward above / below the Kullback–Leibler (KL) divergence between the model during training and the reference model. The APO perspective helps us see that there is a natural paired variant of KTO in which the KL-divergence functions as the anchor:

$$\mathcal{L}_{\text{KTO-pair}}(x, y_w, y_l; \theta) = \tag{9}$$
$$-\sigma\left(r_\theta(x, y_w) - \beta\,KL\right) - \sigma\left(\beta\,KL - r_\theta(x, y_l)\right)$$

This *KL* term is non-negative, and thus the winning reward is pushed to be positive; the losing reward can still be either positive or negative.

The KTO authors report that KTO leads to good alignment without an initial phase of Supervised Fine-Tuning (SFT) on the winning outputs, while DPO does benefit from this SFT phase in their experiments. APO sheds new light on this finding: an increase in likelihood of winning outputs is already built into KTO, whereas it is not guaranteed for DPO alone. However, this is only a desirable property of an alignment objective if the winning output quality is better than the target model's quality, as described in Section 2. When aligning a strong model on preferences which contain generally lower quality outputs, a KTO-style objective runs the risk of deteriorating the model.

5 Alignment Experiments

To study the effectiveness of CLAIR and APO, we align `Llama-3-8B-Instruct` across the four comparable preference datasets described in Section 3, created from 32K `UltraFeedback` prompts. We use GPT4-turbo to act as the *Judge*, *Reviser*, and *Stronger* model when creating these datasets. For the off-policy judge dataset, we use the already judged outputs included in `Ultra-Feedback`. For every dataset, we align the model using the four different objectives described in

Section 4. Additionally, we consider Supervised Fine-Tuning (SFT) on only the winning outputs as a baseline alignment objective.

5.1 Evaluation Methodology

Human judgments are ultimately the best indicator of how well a model is aligned with human preferences. Chatbot Arena (Chiang et al., 2024) uses thousands of pairwise human judgments to produce a ranking of model performance. However, collecting these judgments can be prohibitively expensive. To overcome this obstacle, we measure model performance through benchmarks which correlate highly with this Chatbot Arena ranking.

`MixEval-Hard` (Ni et al., 2024) is a benchmark with very high Chatbot Arena correlation (0.96 rank correlation). `MixEval-Hard` features hard queries with known answers across a wide range of domains and uses a GPT3.5-turbo (Brown et al., 2020; Ouyang et al., 2022) model to evaluate if predicted answers correspond with this ground-truth. This makes `MixEval-Hard` more grounded in human knowledge and significantly cheaper to run compared to other popular evaluation frameworks such as `AlpacaEval` (Li et al., 2023; Dubois et al., 2024). Under the hood, `MixEval-Hard` utilizes queries sampled from `MATH` (Hendrycks et al., 2021), `BBH` (Suzgun et al., 2023), `DROP` (Dua et al., 2019), `GSM8k` (Cobbe et al., 2021), `AGIEval` (Zhong et al., 2024), `TriviaQA` (Joshi et al., 2017), `MBPP` (Austin et al., 2021), `MMLU` (Hendrycks et al., 2020), `HellaSwag` (Zellers et al., 2019), `BoolQ` (Clark et al., 2019), `GPQA` (Rein et al., 2023), `PIQA` (Bisk et al., 2020), `OpenBookQA` (Mihaylov et al., 2018), `ARC` (Clark et al., 2018), `CommonsenseQA` (Talmor et al., 2019), and `SIQA` (Sap et al., 2019).

Our evaluation of `Llama-3-8B-Instruct` before any additional alignment achieves a score of 41.45% on the `2024-06-01` version of `MixEval-Hard`. The gap between `Llama-3-8B-Instruct` and GPT4-turbo is 17%. On the `2024-08-11` split, `Llama-3-8B-Instruct` achieves 40.5%.

Additionally, we consider the length-controlled `LC-AlpacaEval2.0` win rate (Dubois et al., 2024). However, two factors lead us to favor `MixEval-Hard` as our primary evaluation tool. The first is practical: `LC-AlpacaEval2.0` is prohibitively expensive to run, we thus use

Dataset	Objective	ME-Hard 2024-06-01		ME-Hard 2024-08-11		LC-AlpacaEval2.0	
		Max Δ	Mean Δ	Max Δ	Mean Δ	Score Δ	Length Δ
Judge off-policy	DPO	1.10	−0.74 (1.15)	4.30	2.85 (0.75)	<u>2.94</u>	−158
	KTO-pair	−1.00	−2.89 (0.96)	4.05	1.18 (1.67)	−5.69	−437
	SFT	−1.95	−1.63 (1.06)	2.85	0.42 (1.20)	−22.29	12,669
	APO-zero	0.80	−1.99 (1.23)	4.65	1.26 (1.62)	−2.42	−395
	APO-down	<u>2.70</u>	<u>0.64</u> (0.98)	<u>4.80</u>	<u>3.52</u> (0.85)	2.40	−203
Judge on-policy	DPO	4.00	0.56 (1.61)	5.20	2.71 (1.41)	4.98	341
	KTO-pair	2.45	−0.51 (1.26)	5.05	1.13 (1.70)	3.02	452
	SFT	0.65	−0.91 (1.01)	4.20	2.55 (0.70)	1.34	156
	APO-zero	<u>4.65</u>	0.02 (1.66)	<u>5.35</u>	2.19 (1.28)	5.51	484
	APO-down	3.65	<u>1.60</u> (0.95)	4.25	<u>3.06</u> (0.76)	**7.63**	386
CLAIR	DPO	0.55	−1.68 (1.73)	5.05	2.77 (1.40)	2.65	966
	KTO-pair	2.15	0.79 (0.98)	4.65	2.92 (0.86)	4.33	160
	SFT	0.95	−1.63 (1.03)	2.70	0.92 (1.21)	−0.47	6,108
	APO-zero	**7.65**	**2.93** (1.98)	**5.95**	**4.39** (0.89)	<u>5.08</u>	520
	APO-down	−1.05	−5.22 (1.55)	−1.20	−3.61 (1.05)	−6.30	2,559
Stronger Preferred	DPO	−5.00	−6.94 (1.03)	−3.10	−4.40 (0.98)	−2.89	597
	KTO-pair	−1.20	−5.21 (1.27)	2.25	0.50 (1.13)	0.71	153
	<u>SFT</u>	<u>2.45</u>	<u>0.49</u> (1.31)	<u>5.05</u>	<u>2.73</u> (1.21)	<u>6.99</u>	1,883
	APO-zero	−1.70	−2.72 (1.40)	−4.85	−12.02 (5.38)	0.89	243
	APO-down	−6.50	−12.51 (4.97)	1.65	0.16 (1.22)	1.87	10,001

Table 2: Max and mean `MixEval-Hard` improvements for the `2024-06-01` and `2024-08-11` splits, aggregated over 18 epochs of aligning `Llama-3-8B-Instruct`. Best overall performance **bold**, best performance per dataset <u>underlined</u>, standard deviation in parentheses. While `MixEval-Hard` functions as our primary evaluation tool, we also report the average `LC-AlpacaEval2.0` score increase over the two best `MixEval-Hard` checkpoints, and average length increase (in characters) of the responses. CLAIR leads to the greatest overall performance improvement on `MixEval-Hard`. APO methods achieve the best performance across both Judged and CLAIR datasets.

`MixEval-Hard` for the bulk of our evaluation. The second concerns the assessment itself: while both benchmarks are highly correlated with human-produced model rankings, `MixEval-Hard` utilizes questions with known ground-truth answers whereas `LC-AlpacaEval2.0` uses an LLM judge without any ground-truth to decide correctness.

5.2 Training Specifications

`Llama-3-8B-Instruct` is trained for a total of 18 epochs on each preference dataset and alignment objective, with a checkpoint saved every single epoch. The β hyperparameter, common to all alignment objectives except SFT, is set to 0.1. Prompt and responses are truncated to 512 tokens each. Each model is trained using an effective batch size of 16 across one node of 8 `NVIDIA H100` GPUs, using the `RMSProp` optimizer with a learning rate of 2×10^{-7}, linearly decaying to 0 over the 18 epochs. All training is implemented using the `TRL` library (von Werra et al., 2020).

5.3 Results

We report the maximal and mean `MixEval-Hard` improvement over all checkpoints from the same training run. This helps us understand both the best-case and average impact of alignment across the entire training procedure. We use both `2024-06-01` and `2024-08-11` versions of `MixEval-Hard`, which each feature a distinct set of queries. Due to the increased cost associated with `LC-AlpacaEval2.0`, we only measure the win rate for the two best `MixEval-Hard` checkpoints and report their average. We use no system prompt for both evaluations. Our analysis is summarized in Table 2 for every dataset and objective; we now discuss these results in more detail.

5.3.1 Preference Data

To assess the quality of a particular dataset, we consider the performance of that dataset when paired with its best objective. Using the APO-zero objective, **the contrastive CLAIR dataset leads to the greatest improvement**. On the `2024-06-01` split of `MixEval-Hard`, CLAIR leads to the greatest maximal improvement of +7.65% and the greatest average improvement of +2.93% out of all our experiments. This improvement of +7.65% closes the relative gap with GPT4-turbo by 45% using only 32K pairs.

We noted in Section 1 that uncontrolled contrastiveness can degrade model performance. We see this dramatically in the results for the Stronger Preferred dataset, which can heavily degrade model performance. Like CLAIR, this dataset has all winning outputs produced by a stronger model. Unlike CLAIR, though, its examples provide no guarantee of relevant minimal contrasts. Thus, **the contrastiveness induced by the CLAIR revision process is a major driver of performance**.

Both on-policy judge and off-policy judge datasets lead to improved performance when paired with their best alignment objective, but **on-policy preferences lead to better performance compared to off-policy preferences**. This is intuitive; judgments about the target model's outputs are in general more relevant.

The `LC-AlpacaEval2.0` results generally follow a similar trend compared to `MixEval-Hard`, although the on-policy judge dataset attains a higher score compared to CLAIR. While both benchmarks correlate highly with human ratings of models, `MixEval-Hard` is our primary and most significant evaluation tool – we are able to evaluate every model checkpoint across two `MixEval-Hard` splits due to its low cost. Additionally, we remark on a potential issue with the robustness of `LC-AlpacaEval2.0` in Appendix D. A performance breakdown in function of `MixEval-Hard`'s constituent benchmarks is given in Appendix B.

5.3.2 Alignment Objectives

On `MixEval-Hard`, **Anchored Preference Optimization (APO) consistently leads to the greatest performance increase for every preference dataset**, with the exception of the Stronger Preferred dataset, where all contrastive objectives underperform SFT. The relation between the preference dataset and the target model controls which variant of APO is best for any dataset, as predicted in Section 2. **APO-down results in the best performance when winning outputs are generally worse than the target model's answers**, as is the case for the off-policy judge dataset. **APO-zero is the best objective when winning outputs are generally better than the target model's answers**, as is the case for CLAIR and on-policy judge datasets. The difference between alignment objectives is less salient for the on-policy judge dataset as compared to CLAIR, since winning on-policy judge outputs are only slightly better than `Llama-3-8B-Instruct` on average. Winning CLAIR outputs may be vastly better than `Llama-3-8B-Instruct` since they are produced by a stronger model, making the different in alignment objectives more noticeable.

5.4 Analysis

To more deeply understand how the target model is changed during training, we can study the trajectories of winning / losing likelihoods and rewards on held-out preferences. Figure 4 plots these trajectories for the APO-down, APO-zero, and DPO experiments on each preference dataset, using 100 held-out preference pairs from that dataset.

5.4.1 Preference Data

First, we observe that the likelihoods help characterize the type of preference dataset. In the on-policy judge dataset, all answers are sampled from the target model and thus have a high likelihood. The off-policy variant has no answers coming from the target model, and hence all likelihoods are low. Both CLAIR and Stronger Preferred have losing outputs with high likelihood and winning outputs with low likelihood.

Any initial discrepancy between log-likelihoods is normalized by the reward, which tracks changes in likelihood and thus starts at exactly 0. The margin between winning and losing reward indicates how much more the winning likelihood increased during training. Positive reward margins can still produce negative log-likelihood margins, if any initial disparity between winning / losing log-likelihood is not overcome. This ends up being the case for our CLAIR dataset. This insight relates to reference-free alignment objectives, which we discuss in Section 6.
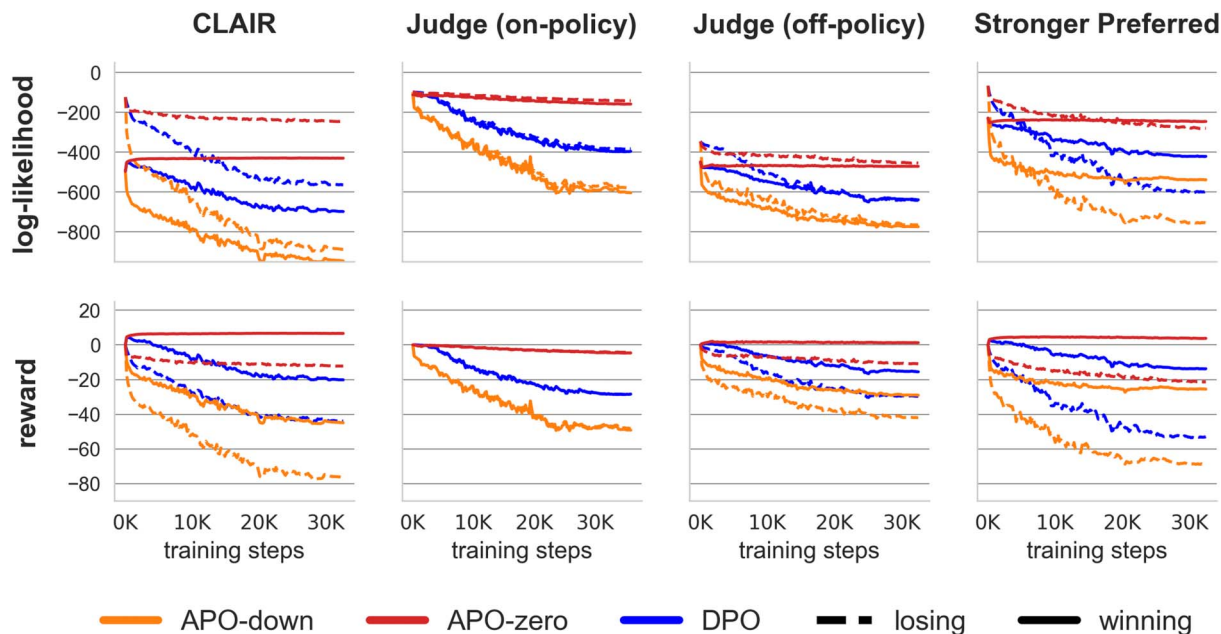
Figure 4: **Log-likelihood** and **reward** on held-out winning and losing outputs for `Llama-3-8B-Instruct` trained on **CLAIR**, **on-policy judge**, **off-policy judge**, and **Stronger Preferred** preference datasets, using APO-down, APO-zero, or DPO alignment objectives. The reward is proportional to the change in log-likelihood during training. All alignment objectives increase the margin between winning and losing rewards during training, but the absolute values of the rewards and log-likelihoods differ starkly due to the exact semantics of the alignment objective and preference dataset. The trend produced by each objective is consistent across datasets, yet no single objective performs best across all datasets (see Table 2).

The training dynamics for CLAIR and Stronger Preferred look very similar, yet the downstream performance on `MixEval-Hard` is completely different. This is because contrastive alignment objectives will exploit any difference between winning and losing outputs to decrease loss. Most of these differences in CLAIR are directly related to improving performance, because CLAIR itself is a minimally contrastive dataset. Many of the differences in Stronger Preferred may not be relevant.

5.4.2 Alignment Objectives

All three alignment objectives display systematic behavior across each dataset. APO-zero consistently leads to the greatest winning and losing rewards. APO-down consistently produces the lowest rewards. Both of these behaviors are as intended. DPO has a slightly more complicated dynamic, which is nonetheless consistent across datasets. In the initial steps of training, DPO tracks the behavior of APO-zero (high rewards) before following APO-down (low rewards) during the remainder of training. This explains why downstream DPO performance correlates most with APO-down. However, DPO is *never* the best

method on any dataset, because it falls between the distinct modes of APO-zero and APO-down.

Training models with contrastive alignment objectives is considerably more complex than conventional supervised fine-tuning. The result is dependent on the semantics of the alignment objective, the contrastive signal in the training data, and the relationship between data quality and target model. Our results show that paying attention to the interplay between these attributes is essential.

6 Related Work

We now characterize relevant alignment efforts and outline how they relate to Contrastive Learning from AI Revisions (CLAIR) and Anchored Preference Optimization (APO).

Reinforcement Learning from Human or AI Feedback (RLHF / RLAIF; Ouyang et al., 2022; Bai et al., 2022; Yuan et al., 2024) is a technique used to align models with human preferences. Fundamentally, these approaches first train a reward model using preference judgments and subsequently optimize a Language Model for this reward using Reinforcement Learning (Schulman

eta l., 2017). To side-step the need for an explicit reward model, Direct Preference Optimization (DPO; Rafailov et al., 2024b) aligns an LM directly using a contrastive training objective.

We articulated two core insights concerning (i) the role of contrastive preference data, and (ii) the need to anchor alignment depending on model and data. These insights translate to any alignment effort which uses comparative preferences. For example, a reward model trained on spurious preference signals may be a less accurate proxy for real rewards, contributing to problems such as *reward overoptimization* or *hacking* (Gao et al., 2023; Rafailov et al., 2024a).

For the remainder of this review, we first focus on contrastive alignment methods and their variants (of which Wang et al., 2024 provide a detailed overview). Finally, we discuss related preference datasets and how they were created.

Changing the LM More / Less: Amini et al. (2024) and Wu et al. (2024a) recognize that preference pairs can vary. Both works study *how much more* preferred the winning output is, and seek to incorporate this into the objective by changing the model more / less depending on this preference strength. Using the difference in gold rewards as a substitute for preference strength, Amini et al. (2024) add an instance-level margin to the contrastive objective while Wu et al. (2024a) scale the β parameter at a batch-level. Other works also utilize a margin in the contrastive loss, but specify this as a static hyperparameter (Zhao et al., 2023; Azar et al., 2024; Meng et al., 2024). These contributions complement our own; they focus on *how much* a model should change, whereas CLAIR creates better learning signals and APO more fully specifies the intended training dynamics.

Controlling Training Dynamics: The tendency of DPO to decrease the winning likelihood has been remarked and analyzed in several works (Feng et al., 2024; Pal et al., 2024). Some works use an additional loss term to explicitly increasing the likelihood of winning outputs (Hong et al., 2024; Pentyala et al., 2024; Adolphs et al., 2023; Zhao et al., 2023; Xu et al., 2024). While these methods can be seen as variants of Anchored Preference Optimization, they do not recognize the need to anchor the objective differently depending on dataset and model, and they do not offer methods that explicitly decrease the winning

likelihood when required. Both Rafailov et al. (2024a) and Azar et al. (2024) generalize a set of alignment methods, but neither allow for any anchoring.

Learning from Unpaired Data: Ethayarajh et al. (2024), Richemond et al. (2024), and Jung et al. (2024) use unpaired examples and rewards for alignment instead of paired examples. Zhang et al. (2024) and Duan et al. (2024) operate solely on undesirable examples in this unpaired setting. In contrast, our work exclusively operates on paired preferences. However, the core insights of APO do apply to unpaired data. For example, Ethayarajh et al. (2024) use binary desired / undesired labels for each answer. We argue this desirability is inherently relative to the model: the same example of desirable behavior used to improve a weak model may actually be an example of undesirable behavior compared to a stronger model, causing the need for anchoring.

Length-controlled Optimization: Preference pairs created through a judging paradigm can be biased towards preferring more verbose answers (Saito et al., 2023). To prevent aligned models from inheriting this bias, Meng et al. (2024) and Park et al. (2024) explicitly control for the length of generations during training. These constraints on generation length can be seamlessly integrated into APO methods as well. In addition, CLAIR revisions could further help with these efforts to reduce the verbosity bias. For example, the *Reviser* could be designed to not increase length.

Reference-free Optimization: Several objectives have opted to directly optimize the contrastive relation between winning / losing likelihoods instead of rewards, removing the need for a secondary reference model (Meng et al., 2024; Zhao et al., 2023; Hong et al., 2024; Xu et al., 2024). Since all these methods are contrastive, the insights from CLAIR and APO directly apply. Additionally, the CLAIR dataset used in our experiments may shed light on the nature of reference-free optimization. Figure 4 shows that our models are sufficiently aligned on the CLAIR dataset when considering rewards, but the absolute likelihood of losing outputs is still greater. This is due to the initial discrepancy in likelihoods produced by the revision process. In many cases, the need for a reference model will

be closely linked to the need for regularization: do we want to align until the absolute likelihoods have changed enough, or do we only want to nudge the likelihoods? This is not clear, but our CLAIR dataset would make a good case-study into reference-free alignment.

Iterative Optimization: Updating the reference model during training can improve results (Kim et al., 2024; Rosset et al., 2024; Wu et al., 2024b). All of these insights are applicable to our work.

Preference Datasets: Chiang et al. (2024) release a dataset of human preference judgments across conversations between humans and several AI assistants. To alleviate the need for human judges, some efforts focus on scaling preference annotations with LLM-based judges (Cui et al., 2024; Zhu et al., 2023) or metric-based judges (Jiang et al., 2023). Unlike our CLAIR method, these works do not create preferences through revisions. Bai et al. (2022) use a set of predetermined criteria (called a *constitution*) to prompt an LLM to revise answers and make them safer (see also Lambert et al., 2024). Dubey et al. (2024) used human revisions in the development of the `llama-3.1` model family. While both efforts create preferences through revisions, we particularly focus on revisions that create a minimal contrast and studied the effect of this contrastiveness on alignment outcomes.

7 Future Work

In this work, we have presented two variants of the APO objective family. Each method accounts for a distinct relationship between target model and preference pair during training. However, real world preference datasets may contain a wide range of different preference pairs, thus the dataset as a whole may not perfectly correspond with any single APO variant. To tackle this, a natural extension of APO could be to select the optimal APO variant at the preference pair level, instead of at the dataset level. Heuristically, this could be achieved using an off-the-shelf reward model to score each preference pair before training.

We used prompted LLMs to create our datasets through revisions or judgments. The distinction between a model and a system of models is arbitrary, and future work could improve CLAIR's performance by using a system of models to produce a revision of higher quality instead.

Additionally, there is a natural trade-off between how much change a revision introduces and how much quality it adds. In many cases it can be challenging to minimally improve a given answer, it is not clear what level of revision would be optimal for alignment. This can be studied empirically by creating different versions of our CLAIR dataset with increasingly intense revisions.

8 Conclusion

Alignment performance is significantly impacted by (i) the contrastiveness of the preference pairs and (ii) the relationship between target model and alignment data. We introduce Contrastive Learning from AI Revisions (CLAIR), a data-creation method which produces better contrasting preference pairs, and Anchored Preference Optimization (APO), a family of alignment objectives with tailored training dynamics. Our experiments aligning `Llama-3-8B-Instruct` show that CLAIR preferences lead to the highest performance improvement out of four comparable preference datasets, and APO methods consistently outperform conventional alignment objectives.

Acknowledgments

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv: 2303.08774*.

Leonard Adolphs, Tianyu Gao, Jing Xu, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. 2023. The CRINGE loss: Learning what language not to model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8854–8874. `https://doi.org/10.18653/v1/2023.acl-long.493`

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571.* `https://doi.org/10.18653/v1/2024.findings-acl.592`

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732.*

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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. 2022. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073.*

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439. `https://doi.org/10.1609/aaai.v34i05.6239`

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoika. 2024. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Forty-first International Conference on Machine Learning.*

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4302–4310.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457.*

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168.*

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni,

Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRA-FEEDBACK: Boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378.

Shitong Duan, Xiaoyuan Yi, Peng Zhang, Tun Lu, Xing Xie, and Ning Gu. 2024. Negating negatives: Alignment without human positive samples via distributional dispreference optimization. *arXiv preprint arXiv: 2403.03419*. https://doi.org/10.18653/v1/2024.findings-emnlp.56

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv: 2404.04475*.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, D. J. Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. 2023. Helping or herding? Reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela.

2024. KTO: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv: 2402.01306*.

Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. 2024. Towards analyzing and understanding the limitations of DPO: A theoretical perspective. *arXiv preprint arXiv:2404.04626*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Miguel A. Hernán and James M. Robins. 2006. Estimating causal effects from epidemiological data. *Journal of Epidemiology & Community Health*, 60(7):578–586. https://doi.org/10.1136/jech.2004.029496, PubMed: 16790829

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*. https://doi.org/10.18653/v1/2024.emnlp-main.626

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. In *The 61st Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2023.acl-long.792

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 1601–1611. `https://doi.org/10.18653/v1/P17-1147`

Seungjae Jung, Gunsoo Han, Daniel Wontae Nam, and Kyoung-Woon On. 2024. Binary classifier optimization for large language model alignment. *arXiv preprint arXiv:2404.04656*.

Dahyun Kim, Yungi Kim, Wonho Song, Hyeonwoo Kim, Yunsu Kim, Sanghoon Kim, and Chanjun Park. 2024. sDPO: Don't use your data all at once. *arXiv preprint arXiv: 2403.19270*.

Nathan Lambert, Hailey Schoelkopf, Aaron Gokaslan, Luca Soldaini, Valentina Pyatkin, and Louis Castricato. 2024. Self-directed synthetic dialogues and revisions technical report. *arXiv preprint arXiv:2407.18421*.

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`

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391. `https://doi.org/10.18653/v1/D18-1260`

Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. 2024. MixEval: Deriving wisdom of the crowd from LLM benchmark mixtures. *arXiv preprint arXiv:2406.06565*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with DPO-positive. *arXiv preprint arXiv:2402.13228*.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*. `https://doi.org/10.18653/v1/2024.findings-acl.297`

Shiva Kumar Pentyala, Zhichao Wang, Bin Bi, Kiran Ramnath, Xiang-Bo Mao, Regunathan Radhakrishnan, Sitaram Asur, and Na (Claire) Cheng. 2024. PAFT: A parallel training paradigm for effective LLM fine-tuning. *arXiv preprint arXiv:2406.17923*.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. 2024a. Scaling laws for reward model overoptimization in direct alignment algorithms. *arXiv preprint arXiv:2406.02900*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2024b. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, et al. 2024. Offline regularised reinforcement learning for large language models alignment. *arXiv preprint arXiv:2405.19107*.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and

Tengyang Xie. 2024. Direct Nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv: 2404.03715*.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051. https://doi.org/10.18653/v1/2023.findings-acl.824

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. TRL: Transformer reinforcement learning. https://github.com/huggingface/trl

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, and Na (Claire) Cheng. 2024. A comprehensive survey of LLM alignment techniques: RLHF, RLAIF, PPO, DPO and more. *arXiv preprint arXiv:2407.16216*.

Becca Williams. 2023. Parallel process GPT. https://github.com/tiny-rawr/parallel_process_gpt

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024a. β-DPO: Direct preference optimization with dynamic β. *arXiv preprint arXiv:2407.08639*.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2024b. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *Forty-first International Conference on Machine Learning*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E. Weston. 2024. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800. https://doi.org/10.18653/v1/P19-1472

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. AGIEval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314. https://doi.org/10.18653/v1/2024.findings-naacl.149

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2023. Starling-7B: Improving LLM helpfulness & harmlessness with RLAIF.

Type	Prompt
Reviser	You are a teacher and your task is to minimally improve a student's answer. I will give you a {{task}} and a {{student_solution}}. Your job is to revise the {{student_solution}} such that it is clearer, more correct, and more engaging. Copy all non-corrected parts of the student's answer. Do not allude to the {{corrected_student_solution}} being a revision or a correction in your final solution.\n\n{{task}}: **<instruction x>** \n\n{{student_solution}}: **<losing output y_l>** \n\n—————\n\nLet's first think step by step with a {{teacher_reasoning}} to decide how to improve the {{student_solution}}, then give the {{corrected_student_solution}}. Mention the {{teacher_reasoning}} and {{corrected_student_solution}} identifiers to structure your answer.\n\n
Judge	You are a teacher and your task is to pick the best student's answer. The best answer is the most clear, most correct, and most engaging answer. I will give you a {{task}} and {{student_solution_1}} and {{student_solution_2}}. Your final answer must contain [1] if {{student_solution_1}} was best, else [2].\n\n{{task}}: **<instruction x>** \n\n{{student_solution_1}}: **<first output y_1>** \n\n{{student_solution_2}}: **<second output y_2>** \n\n—————\n\nLet's first think step by step with a {{teacher_reasoning}} to decide which solution is better, and then answer [1] or [2].\n\n

Table 3: Prompt templates used for creating preference triples (x, y_l, y_w) with the *Reviser* and *Judge* function of Equation 1 and 2. The variables in the prompt template are **bolded** and bracketed. Both prompts target clear, correct, and engaging outputs. The *Reviser* prompt instructs that a losing output y_l should be minimally improved to create the winning output y_w. Instead, the *Judge* prompt picks the winning/losing output out of two candidates y_1 & y_2. Both prompts also instruct a model to produce a reasoning before revising or judging.

A Preference Dataset Creation

A.1 Prompts

The prompts we use for the *Reviser* and *Judge* function of Equation 1 and 2 are given in Table 3. Both prompts contain instructions to prefer more clear, more correct, and more engaging outputs. The *Reviser* prompt creates a preference pair by minimally revising and improving an output according to these preferences. Instead, the *Judge* prompt selects a more preferred output given two candidate answers.

A.2 Preference Pair Filtering

We reject revisions or judgments if the LLM failed to follow formatting guidelines specified in the revising or judging prompt. Additionally, we reject revisions if they altered the length of the original output too much; we found this mainly happens when the LLM misunderstands the revision prompt. Starting from the same 32K instructions sampled from `UltraFeedback`, this procedure creates 29K CLAIR pairs, 29K Stronger Preferred pairs, 29K off-policy Judge pairs, and 32k on-policy Judge pairs. We adapted the code by Williams (2023) to efficiently query closed-source LLMs in parallel over API.

MixEval-Hard split	# query	Llama-3-8B -Instruct	+ CLAIR	+ Judge (on-policy)	+ Judge (off-policy)	+ Stronger Preferred
Overall score	988	41.45	**49.10**	46.10	44.15	43.90
TriviaQA	267	34.30	**49.20**	42.40	43.70	39.80
MMLU	231	**43.70**	39.00	42.00	36.80	34.60
DROP	167	50.20	58.70	64.30	**64.90**	58.90
AGIEval	71	31.00	38.00	38.00	**39.40**	38.00
HellaSwag	61	29.50	**37.70**	26.20	29.50	27.90
CommonsenseQA	50	60.00	**72.00**	60.00	48.00	58.00
BoolQ	37	40.50	**45.90**	32.40	21.60	27.00
GSM8k	22	60.00	80.00	69.50	63.20	**84.10**
SIQA	20	45.00	**50.00**	40.00	15.00	40.00
MATH	16	47.50	63.70	51.30	58.80	**73.10**
BBH	16	51.30	**68.80**	57.50	60.60	66.90
OpenBookQA	8	62.50	62.50	50.00	62.50	**75.00**
GPQA	8	12.50	25.00	25.00	25.00	**37.50**
PIQA	8	50.00	62.50	62.50	62.50	**75.00**
ARC	4	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
MBPP	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Objective used:		/	APO-zero	APO-zero	APO-down	SFT

Table 4: Breakdown of `MixEval-Hard` performance (version `2024-06-01`) in function of which dataset the queries originate from. Analysis given for `Llama-3-8B-Instruct` and our best models on the CLAIR, Judge (on-policy), Judge (off-policy), and Stronger Preferred datasets. While individual splits may not always indicate the best model (particularly when the amount of queries is low), the overall score correlates highly with human judgments about model performance (Chatbot Arena Elo; Chiang et al., 2024). `MixEval-Hard` uses a GPT3.5-turbo model to rate if a response to a query agrees with a known gold-truth response.

B `MixEval-Hard` Performance Breakdown

`MixEval-Hard` features queries from a wide range of established benchmarks, as outlined in Section 5.1. Previously, we reported on the overall `MixEval-Hard` performance. Table 4 breaks down this overall performance in function of these different benchmarks. While `MixEval-Hard` often incorporates only a few queries from any given benchmark, the overall performance correlates highly with human judgments.

Dataset	Objective	ME-Hard 2024-06-01		ME-Hard 2024-08-11		Train Time
		Max Δ	Mean Δ	Max Δ	Mean Δ	
Judge	KTO	2.10	−2.70 (1.67)	4.75	1.31 (1.61)	19h 18m 10s
off-policy	APO-zero-unpaired	−0.40	−3.67 (1.68)	4.35	0.66 (1.44)	12h 32m 58s
Judge	KTO	3.50	1.28 (1.11)	4.85	2.70 (1.35)	19h 40m 10s
on-policy	APO-zero-unpaired	**4.35**	1.31 (1.44)	5.60	3.92 (0.99)	13h 49m 55s
CLAIR	KTO	3.75	**1.47** (1.39)	**5.80**	**4.12** (1.09)	17h 33m 24s
	APO-zero-unpaired	1.40	−1.49 (1.77)	3.20	1.13 (1.21)	12h 31m 03s
Stronger	KTO	−3.25	−4.73 (1.01)	0.30	−1.18 (0.75)	19h 07m 29s
Preferred	APO-zero-unpaired	−2.70	−4.57 (1.32)	2.95	0.50 (1.25)	12h 38m 49s

Table 5: Max and mean `MixEval-Hard` improvements for the `2024-06-01` and `2024-08-11` splits, aggregated over 18 epochs of aligning `Llama-3-8B-Instruct`. Best overall performance **bold**, best performance per dataset underlined, standard deviation in parentheses. KTO is the best unpaired loss given the off-policy Judge and CLAIR datasets, while APO-zero-unpaired performs better when given the on-policy Judge and Stronger Preferred datasets. KTO can take 60% longer to train for the same configuration.

C Unpaired APO

In this work, we designed datasets and alignment objectives for paired preferences (output $y_l \prec y_w$ for input x). The original KTO objective (Ethayarajh et al., 2024) was designed to operate on desirability data (output y for input x was desirable or not), which does not use such paired preferences. We consider an unpaired variant of our APO-zero loss, called APO-zero-unpaired, which resembles the KTO objective but which fixes the KL term to zero. Table 5 compares KTO with APO-zero-unpaired, keeping everything else comparable with our main results in Table 2. To turn our paired datasets into unpaired datasets, we turn each datapoint consisting of two outputs into two datapoints with one output.

There is no clear winner between KTO and APO-zero-unpaired across the board. Within each dataset however, there always is a clear winner. This reflects the main findings of our work, different alignment objectives have distinct semantics, and different datasets require different semantics. APO-zero-unpaired consistently trains faster, due to not calculating the KL term. In some cases, the KTO objective can take 60% longer to train.

D How Well Does `AlpacaEval` Control for Response Lengths?

GPT4 as a judge is known to favor more verbose responses, which can artificially inflate `AlpacaEval` win rates for verbose models (Dubois et al., 2024). To counteract this bias, Dubois et al. (2024) estimate a length-controlled `AlpacaEval` win rate, which we report on in Table 2. Specifically, the authors adopt a causal inference framework to answer the question ''What would the `AlpacaEval` metric be, if the outputs of all models had the same length as those of the baseline?'' (Dubois et al., 2024).

In order to meaningfully apply causal inference, a few key assumptions need to be met. The *Positivity* assumption (Hernán and Robins, 2006) states that, when estimating the effect of a treatment, there are at least some subjects which receive the treatment for all covariates. Intuitively, the Positivity assumption applied to the length-control question states that you need to observe at least some long and some short responses for every model in order to accurately estimate how the response length influences the model's win rate.

The `AlpacaEval` framework does not check if this Positivity assumption is met, potentially giving bad estimates for the length-controlled win rates in some settings. If a certain model consistently generates responses longer than those of the baseline, it is impossible to accurately estimate how good the responses would be if they were as long as the baseline.

This may give us insights into some of our length-controlled `AlpacaEval` win rates. For example, the SFT result on the Stronger Preferred dataset in Table 2 seems disproportionately high in comparison to the `MixEval-Hard` results for that same experiment. This model is considerably more verbose than `Llama-3-8B-Instruct`, as evident from the large response length increase associated with this experiment (+ 1883 characters on average). It is possible the Positivity constraint was not met for this experiment, causing the length-controlled framework of `AlpacaEval` to provide inaccurate estimates.

While a more thorough study of length-controlled win rate is out of scope for this work, one potential avenue towards a more robust length-controlled win rate would be to specifically prompt models to generate shorter or longer answers if the Positivity constraint is not met.