# Perplexity-aware Correction for Robust Alignment with Noisy Preferences

**Keyi Kong**[1,*] **Xilie Xu**[2,*] **Di Wang**[3] **Jingfeng Zhang**[4,5,†] **Mohan Kankanhalli**[2]
[1]Shandong University [2]National University of Singapore
[3]King Abdullah University of Science and Technology [4]The University of Auckland
[5]RIKEN Center for Advanced Intelligence Project (AIP)
luxinyayaya@mail.sdu.edu.cn, xuxilie@comp.nus.edu.sg
di.wang@kaust.edu.sa, jingfeng.zhang@auckland.ac.nz, mohan@comp.nus.edu.sg

## Abstract

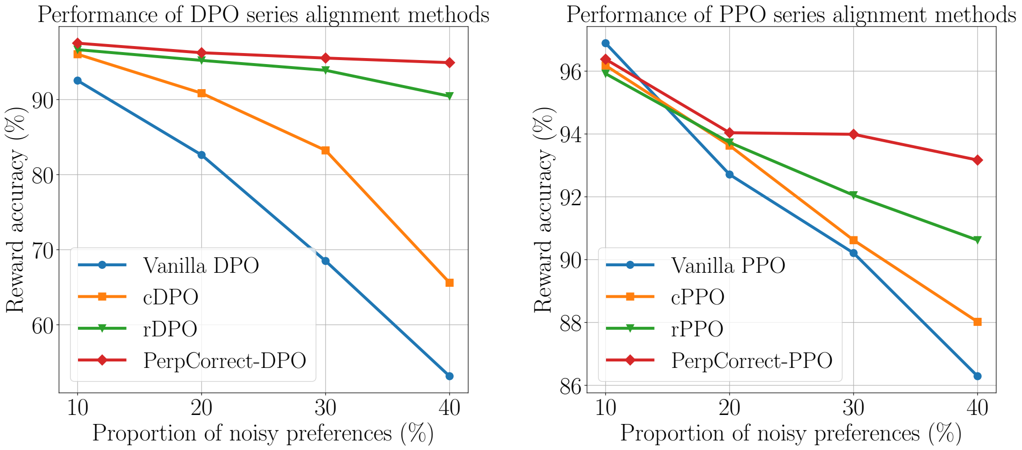
Alignment techniques are critical in ensuring that large language models (LLMs) output helpful and harmless content by enforcing the LLM-generated content to align with human preferences. However, the existence of noisy preferences (NPs), where the responses are mistakenly labelled as chosen or rejected, could spoil the alignment, thus making the LLMs generate useless and even malicious content. Existing methods mitigate the issue of NPs from the loss perspective by adjusting the alignment loss based on a clean validation dataset. Orthogonal to these loss-oriented methods, we propose perplexity-aware correction (PerpCorrect) from the data perspective for robust alignment which detects and corrects NPs based on the differences between the perplexity of the chosen and rejected responses (dubbed as PPLDiff). Intuitively, a higher PPLDiff indicates a higher probability of the NP because a rejected/chosen response which is mistakenly labelled as chosen/rejected is less preferable to be generated by an aligned LLM, thus having a higher/lower perplexity. PerpCorrect works in three steps: (1) PerpCorrect aligns a surrogate LLM using the clean validation data to make the PPLDiff able to distinguish clean preferences (CPs) and NPs. (2) PerpCorrect further aligns the surrogate LLM by incorporating the reliable clean training data whose PPLDiff is extremely small and reliable noisy training data whose PPLDiff is extremely large after correction to boost the discriminatory power. (3) Detecting and correcting NPs according to the PPLDiff obtained by the aligned surrogate LLM to obtain a denoised training dataset for robust alignment. Comprehensive experiments validate that our proposed PerpCorrect can achieve state-of-the-art alignment performance under NPs. Notably, PerpCorrect demonstrates practical utility by requiring only a modest amount of validation data and being compatible with various alignment techniques. Our code is available at PerpCorrect.

## 1 Introduction

Alignment enables the safe utilization of the remarkable capabilities acquired by large language models (LLMs) through self-supervised learning on vast corpora [6, 24, 4]. It refers to the process of ensuring that the contents generated by LLMs are helpful, harmless, and aligned with human values and preferences [19]. Reinforcement Learning from Human Feedback (RLHF) [9] has emerged as a primary technique for achieving alignment. Current technical routes [39, 40, 29] require a reward model to simulate human preference and use it to optimize the policy model outputs with

[*] Equal contribution.
[†] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) DPO series methods.

(b) PPO series methods.

Figure 1: We evaluated various robust alignment methods under different proportions of noisy preferences using the Llama2-7B model, on the Golden HH dataset. The reward accuracy of both the vanilla DPO and PPO method significantly decreases as the proportion of noisy preferences increases. Our method, perplexity-aware correction (PerpCorrect), outperforms both the DPO and PPO series baselines across different proportions of noisy preferences.

Proximal Policy Optimization (PPO) [27]. Current offline techniques such as Direct Preference Optimisation (DPO) [26], Sequence Likelihood Calibration with Human Feedback (SLiC) [38] and Identity-Preference Optimisation (IPO) [3], could directly align LLMs without intensively training a reward model as employed in RLHF.

Recent studies [34, 8] have shown there exist noisy preferences (NPs) that may lead to significant degradation in alignment performance. The issue of NPs, where the label of the actually chosen/rejected responses in training datasets is flipped as rejected/chosen, can arise from the biases of annotators [34] and malicious noise injection [5]. As shown in Figure 1, when NPs are randomly injected into the training dataset, the conventional alignment method (e.g., DPO [26] and PPO [9]) will have significantly degraded alignment performance measured by the reward accuracy. Such performance degradation could result in the generation of useless and even malicious content [34]. Therefore, it necessitates developing robust alignment methods that can utilize datasets with NPs to effectively align the LLMs with human preferences.

Existing robust alignment methods are proposed from the loss perspective, which adjust the alignment loss using a clean validation dataset to mitigate the issue of NPs. In particular, the conservative DPO (cDPO) [21] and robust-DPO (rDPO) [8] both estimate the proportion of NPs using the clean validation data via cross-validation and then adjust the original DPO loss based on the estimated proportion of NPs. However, Mitchell [21] and Chowdhury et al. [8] overlooked the essential differences between noisy and clean preferences, which is critical for mitigating the issue of NPs.

To this end, we propose **Perp**lexity-aware **Correct**ion (PerpCorrect) for robust alignment from the data perspective by leveraging the differences between noisy and clean preferences for robust alignment. PerpCorrect detects and corrects NPs based on the difference between the perplexity of the chosen response and that of the rejected counterparts (dubbed as PPLDiff) obtained by an aligned surrogate LLM using the clean validation set. If an NP is detected, PerpCorrect will correct it by flipping the label of the rejected/chosen responses as chosen/rejected. Intuitively, rejected responses which are mistakenly labelled as chosen have a higher perplexity since they are less consistent with human preferences and thus have a lower probability of being generated after alignment. Therefore, a higher value of PPLDiff indicates a higher probability of the preferences being noisy. In this way, PerpCorrect leverages the differences between noisy and clean preferences (CPs) identified by PPLDiff to detect NPs.

To make the PPLDiff able to distinguish CPs and NPs, PerpCorrect requires an aligned surrogate LLM for calculating PPLDiff. The density of PPLDiff obtained on the noisy training dataset using an unaligned surrogate LLM, which can be fitted as a normal distribution centered around zero (evidenced in Figure 2a), cannot discriminate CPs and NPs. Therefore, we align a surrogate LLM using the clean validation data. The density of PPLDiff obtained by the aligned surrogate LLM in Figure 2b can be fitted into two distinguishable normal distributions, thus being able to differentiate CPs and NPs.

However, there still exists a large overlap between two normal distributions after aligning only on the clean validation dataset, which could result in an unsatisfactory accuracy of NP detection. To this end, we iteratively align the model using more reliable clean training data with extremely low PPLDiff (located in the green area in Figure 2c) and reliable noisy training data with extremely large PPLDiff (located in the red area in Figure 2c) sampled from noisy training datasets. Finally, the two normal distributions are significantly separated as shown in Figure 2d, which indicates that PPLDiff has an enhanced discriminatory power.

Benefiting from the strong discriminatory power of PPLDiff calculated by the aligned surrogate LLM, PerpCorrect outputs a denoised training dataset for robust alignment by first detecting NPs based on a PPLDiff threshold and then correcting them. The data, whose PPLDiff is below a certain threshold (i.e., the black dotted line in Figure 2d) selected as the x-coordinate of the two normal distributions' intersection, are identified as NPs and thus corrected by flipping the response's label. Notably, our proposed PerpCorrect is compatible with various alignment methods as well as robust alignment methods [21, 8] since the metric PPLDiff is agnostic to training algorithms and only requires an arguably small number of clean validation data (~50), thus making it practical.

Comprehensive empirical results, evaluated using the Llama2-7B [32] and phi-2 [20] models on the OpenAssistant Conversations (OASST1) [17] and Golden HH [7] datasets, validate the effectiveness of our proposed PerpCorrect method in robustifying alignment with NPs. We empirically validate that PerpCorrect consistently obtains state-of-the-art performance among various proportions of NPs. Besides, we empirically demonstrate that PerpCorrect can effectively robustify various alignment techniques and robust alignment methods, validating its compatibility.

## 2 Literature Review and Preliminary

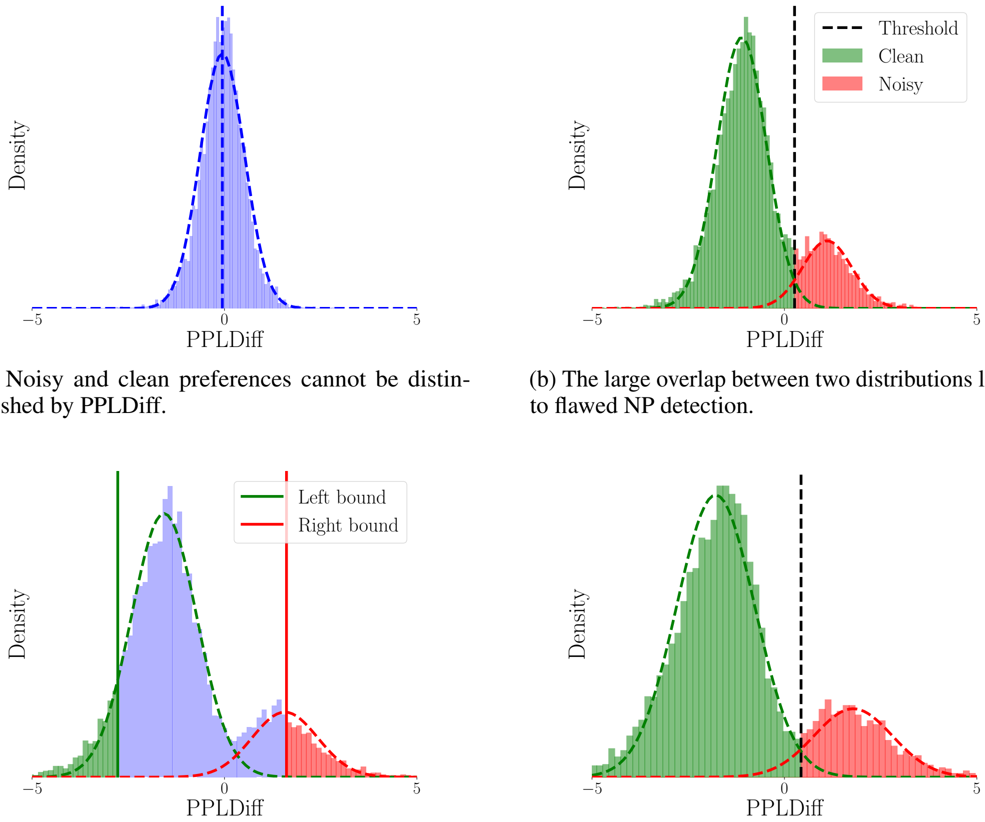
In this section, we introduce the related work about LLM alignment and provide preliminaries about the noisy preferences, perplexity, as well as various alignment methods.

### 2.1 LLM Alignment

In the domain of aligning LLMs with human preferences, pairwise preference methods are favored due to their lower cognitive burden on evaluators. Traditional online alignment approaches [32, 24, 29] involve training reward models from these preferences to provide signals in reinforcement learning. Recent offline alignment methods like Direct Preference Optimization (DPO) [26], Sequence Likelihood Calibration (SLiC) [38], and Identify Preference Optimization (IPO) [3] streamlined this process by directly using preference pairs to train LLMs, thus enhancing performance and reducing computational costs. Additionally, methods like RRHF [37] align LLMs using multiple ranked preferences, Kahneman-Tversky Optimization (KTO) [13] align LLMs using a single preference labeled as good or bad, and Rejection Sampling Optimization (RSO) [18] address DPO's limitation in sampling preference pairs from the optimal policy through rejection sampling. However, NPs, arising from the biased human feedback, can determine the alignment performance [24, 34]. Robust alignment methods like conservative DPO (cDPO) [21], robust DPO (rDPO) [8] have been proposed to address these issues from the loss perspective. Our approach focuses on the data perspective to address these issues of NPs and is orthogonal to these robust alignment methods.

### 2.2 Preliminary

**Noisy preferences (NPs).** NPs refer to preference data in training datasets, whose label of the actually chosen/rejected responses is flipped as rejected/chosen. Let $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$ be the preference dataset consisting of $N \in \mathbb{N}$ preference data points. For each preference data point $(x, y_w, y_l) \in \mathcal{D}$, $x$ is the prompt input to LLMs, $y_w$ is the chosen response, and $y_l$ is the rejected response. We let $\tilde{\mathcal{D}} = \{(x^{(i)}, \tilde{y}_w^{(i)}, \tilde{y}_l^{(i)})\}_{i=1}^N$ be the noisy preference dataset (i.e., preference dataset



(a) Noisy and clean preferences cannot be distinguished by PPLDiff.

(b) The large overlap between two distributions leads to flawed NP detection.

(c) Aligning the surrogate LLM using extra reliable training data.

(d) Separating and correcting noisy preferences based on the threshold.

Figure 2: We visualized the PPLDiff under the entire PerpCorrect process using Llama2-7B on Golden HH dataset with 20% noisy preferences. We use the green dotted line to represent the normal distribution formed by clean data, the red dotted line represents the normal distribution formed by noisy data, and the black dotted line represents the threshold.

consisting noisy preferences) and denote preference data points that are not noisy as clean preferences (CPs). Following Chowdhury et al. [8], we obtain the noisy preference dataset $\tilde{\mathcal{D}}$ using the standard random noise model [23] with the probability $\varepsilon \in (0, 50\%)$ to change the data point into noisy preferences, i.e.

$$\mathbb{P}_{(x^{(i)}, \tilde{y}_w^{(i)}, \tilde{y}_l^{(i)}) \sim \tilde{\mathcal{D}}} \left[ (x^{(i)}, \tilde{y}_w^{(i)}, \tilde{y}_l^{(i)}) = (x^{(i)}, y_l^{(i)}, y_w^{(i)}) \right] = \varepsilon. \tag{1}$$

**Perplexity (PPL).** PPL [15] measures the probability that the LLM generates a sentence. A lower PPL of a sentence indicates that the LLM has generated this sentence with a high probability. PPL is defined as the average negative log-likelihood of a sequence, i.e.,

$$\mathrm{PPL}(s; \theta) = \exp(-\frac{1}{t} \sum_{i=1}^{t} \log \pi_\theta(s_i | s_{<i})), \tag{2}$$

where $s$ is a sequence composed of $t$ tokens and $\log \pi_\theta(s_i | s_{<i})$ denotes the log-likelihood of the $i$-th token given the preceding tokens $s_{<i}$ calculated by an LLM $\pi_\theta$.

**Technical details of alignment methods.** There are usually three phases in RLHF pipeline [34, 26]: (1) supervised fine-tuning (SFT); (2) reward modeling; (3) reinforcement learning (RL) optimization. In the SFT phase, an LLM is fine-tuned via supervised learning on high-quality task-related data.

We denote the LLM after the SFT phase as $\pi_{\text{SFT}}$. In the reward modeling phase, the reward model is introduced to simulate human preferences. Given a preference dataset, a reward model $r_\omega(x, y)$ parameterized by $\omega$, which takes prompt $x$ and response $y$ as input and outputs a real number representing the reward score, can be optimized via minimizing the following loss function:

$$\mathcal{L}_R(r_\omega, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma(r_\omega(x, y_w) - r_\omega(x, y_l))\right], \tag{3}$$

where $\sigma$ is the logistic function. In the RL optimization phase, the objective function is as follows:

$$\max_\theta \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[r_\omega(x, y) - \beta \cdot (\log \pi_\theta(y|x) - \log \pi_{\text{ref}}(y|x))], \tag{4}$$

where $\pi_\theta(y|x)$ represents the probability that the LLM parameterized by $\theta > 0$ generates the response $y$ given the prompt $x$, $\pi_{\text{ref}}$ is a reference LLM to maintain the generation ability of the aligned model, and $\beta$ is a hyper-parameter to ensure the similarity between $\pi_\theta(y \mid x)$ and $\pi_{\text{ref}}(y \mid x)$. We take $\pi_{\text{SFT}}$ as the reference LLM $\pi_{\text{ref}}$ following Ouyang et al. [24].

Recently, offline alignment methods directly leverages preferences in preference datasets, bypassing the need to learn a reward model in RLHF. The LLM parameters are optimized by minimizing the following loss function:

$$\mathcal{L}(\pi_\theta; \pi_{\text{ref}}) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\mathcal{G}(x, y_w, y_l; \theta)\right], \tag{5}$$

where the function $\mathcal{G}$ changes with the alignment method. To be specific, DPO [26] uses a BCE loss, SLiC [38] uses a hinge loss, and IPO [3] uses a square loss:

$$\mathcal{G}_{\text{DPO}}(x, y_w, y_l; \theta) = -\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right), \tag{6}$$

$$\mathcal{G}_{\text{SLiC}}(x, y_w, y_l; \theta) = \max \left\{0, 1 - \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right\}, \tag{7}$$

$$\mathcal{G}_{\text{IPO}}(x, y_w, y_l; \theta) = \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} - \frac{1}{2}\right)^2. \tag{8}$$

To mitigate the issue of NPs, cDPO [21] and rDPO [8] adjust the DPO loss based on the estimated proportion of NPs $\varepsilon'$ using a clean validation dataset $\mathcal{D}_{\text{val}} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^{N_{\text{val}}}$ consisting of $N_{\text{val}} \in \mathcal{N}$ clean preference data points, i.e.

$$\mathcal{G}_{\text{cDPO}}(x, \tilde{y}_w, \tilde{y}_l; \theta) = (1 - \varepsilon')\mathcal{G}_{\text{DPO}}(x, \tilde{y}_w, \tilde{y}_l; \theta) + \varepsilon'\mathcal{G}_{\text{DPO}}(x, \tilde{y}_l, \tilde{y}_w; \theta), \tag{9}$$

$$\mathcal{G}_{\text{rDPO}}(x, \tilde{y}_w, \tilde{y}_l; \theta) = \frac{(1 - \varepsilon')\mathcal{G}_{\text{DPO}}(x, \tilde{y}_w, \tilde{y}_l; \theta) - \varepsilon'\mathcal{G}_{\text{DPO}}(x, \tilde{y}_l, \tilde{y}_w; \theta)}{1 - 2\varepsilon'}. \tag{10}$$

## 3 Perplexity-aware Correction for Robust Alignment

This section introduces **Perp**lexity-aware **Correct**ion (PerpCorrect) for robust alignment with NPs. In Section 3.1, we introduce a novel metric called PPLDiff and then illustrate the pipeline of PerpCorrect to detect and correct NPs based on PPLDiff. In Section 3.2, we demonstrate how to adapt our proposed PerpCorrect with various alignment methods to achieve robust alignment.

### 3.1 Perplexity-aware Correction (PerpCorrect)

In this subsection, we introduce PerpCorrect which employs a novel metric called PPLDiff as the foundation for detecting and correcting NPs. The algorithm of PerpCorrect is described in Algorithm 2.

**PPLDiff.** PPLDiff measures the difference between the PPL of chosen response and that of the rejected response. Given a preference data point $(x, \tilde{y}_w, \tilde{y}_l) \in \tilde{\mathcal{D}}$ sampled from the noisy training dataset $\tilde{\mathcal{D}}$ and an LLM $\pi_\theta$, PPLDiff is defined as follows:

$$\text{PPLDiff}(x, \tilde{y}_w, \tilde{y}_l; \theta) = \log \text{PPL}([x; \tilde{y}_w]; \theta) - \log \text{PPL}([x; \tilde{y}_l]; \theta), \tag{11}$$

where $[x; y]$ indicates the concatenation of the prompt $x$ and the response $y$. Intuitively, if a data point is a clean preference, the $\text{PPL}([x; \tilde{y}_w]; \theta)$ will be lower than $\text{PPL}([x; \tilde{y}_l]; \theta)$ because the sequence $[x; \tilde{y}_w]$ is more aligned with human values and thus has a higher probability of being generated by aligned LLMs. As a result, it PPLDiff will be lower compared to NPs, which $\text{PPL}([x; \tilde{y}_w]; \theta)$ is higher than $\text{PPL}([x; \tilde{y}_l]; \theta)$. This difference allows us to distinguish CPs and NPs based on PPLDiff.

**Aligning a surrogate LLM only using clean validation data.** Here, we utilize a clean validation dataset $\mathcal{D}_{\text{val}}$ to obtain an aligned surrogate LLM to make PPLDiff able to distinguish CPs and NPs. We empirically find that the PPLDiff values of CPs and NPs calculated by an unaligned LLM in the noisy training dataset were initially indistinguishable as shown in Figure 2a, making it impossible to differentiate the NPs from CPs. This is because an unaligned LLM lacks the necessary preferences to distinguish NPs and CPs.

Therefore, we introduce a surrogate LLM $\pi_{\theta'}$ parameterized by $\theta'$ to replace the unaligned LLM and use it for calculating PPLDiff. We optimize the surrogate LLM $\pi_{\theta'}$ using the clean validation dataset $\mathcal{D}_{\text{val}}$ as follows:

$$\max_{\theta'} \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{val}}} \left[\mathcal{G}_{\text{DPO}}(x, y_w, y_l; \theta')\right]. \tag{12}$$

After aligning the surrogate LLM, the PPLDiff values of NPs calculated by the surrogate LLM $\pi_{\theta'}$ are significantly increased and those of CPs are significantly decreased, forming two distinct distributions as shown in Figure 2b. This is because the aligned surrogate LLM is trained to generate responses that align with human preferences, enhancing its ability to distinguish between NPs and CPs based on PPLDiff.

To separate CPs and NPs in the noisy training dataset without knowing the oracle preferences, we leverage the Levenberg-Marquardt (LM) algorithm to find two normal distributions that fit the density of PPLDiff calculated by the aligned surrogate LLM. Specifically, the LM algorithm returns the constants $\bar{\varepsilon}, \bar{\mu}, \bar{\sigma}$ that satisfies the following condition:

$$h(x|\bar{\varepsilon}, \bar{\mu}, \bar{\sigma}) = (1-\bar{\varepsilon}) f_{\text{clean}}(x|\bar{\mu}, \bar{\sigma}^2) + \varepsilon f_{\text{noisy}}(x|-\bar{\mu}, \bar{\sigma}^2), \tag{13}$$

$$\text{where} \quad f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}). \tag{14}$$

Note that $x$ is the PPLDiff value and $h(x|\bar{\varepsilon}, \bar{\mu}, \bar{\sigma})$ is the superposition of these two normal distribution. We denote $f_{\text{clean}}(x|\bar{\mu}, \bar{\sigma}^2)$ as the normal distribution fitting the PPLDiff of CPs and $f_{\text{noisy}}(x|-\bar{\mu}, \bar{\sigma}^2)$ as the normal distribution fitting the PPLDiff of NPs since the PPLDiff of NPs is intuitively higher than that of CPs. In this way, we can obtain two distinguishable normal distributions to separate NPs and CPs as shown in the green and red dotted lines of Figure 2b without knowing the oracle preferences.

**Further aligning the surrogate LLM using extra reliable training data from noisy training datasets.** After aligning only using the clean validation datasets, the discriminatory power of the PPLDiff is still far from satisfactory because of the large overlap between the two normal distributions. Therefore, we align the surrogate LLM with more reliable training data to make the PPLDiff of CPs and that of NPs more separable. We iteratively align the surrogate LLM $\pi_{\theta'}$ using more reliably clean training data whose PPLDiff is extremely small and reliably noisy training data whose PPLDiff is extremely large after correction by flipping the label of the response.

Specifically, at epoch $t \in \mathbb{N}$, we select $(t-1) \cdot \alpha \cdot |\tilde{\mathcal{D}}|$ of the training data along with the clean validation data for further alignment where $\alpha \in (0, 1)$ is the selection ratio and $|\tilde{\mathcal{D}}| = N$ is the number of data points in noisy training dataset. As shown in Lines 33–45 of Algorithm 2, the selected reliable training dataset $\mathcal{D}'_t$ consists of $(t-1) \cdot \alpha \cdot (1-\bar{\varepsilon}) \cdot |\tilde{\mathcal{D}}|$ reliably clean training data whose PPLDiff values are smallest $(t-1) \cdot \alpha \cdot (1-\bar{\varepsilon})$ percent and $(t-1) \cdot \alpha \cdot \bar{\varepsilon} \cdot |\tilde{\mathcal{D}}|$ reliably noisy training data after correction. Note that the reliably clean training data are the data points whose PPLDiff values are smallest $(t-1) \cdot \alpha \cdot (1-\bar{\varepsilon})$ percent (located in the green area of Figure 2c), and the reliably noisy training data whose PPLDiff values are largest $(t-1) \cdot \alpha \cdot \bar{\varepsilon}$ percent (located in the red area of Figure 2c) among all the training data points.

**Detecting and correcting NPs based on PPLDiff to output a denoised training dataset.** Based on the PPLDiff calculated by the aligned surrogate LLM, PerpCorrect detects and corrects NPs whose PPLDiff value is lower than a certain threshold. We take the x-coordinate of the intersection of the two normal distributions as the threshold (the black dotted line in Figure 2d). As shown in Lines 23–31, data points whose PPLDiff values are larger than this threshold are identified as CPs (the green area in Figure 2d), and other data points are identified as NPs requiring correction (the red area in Figure 2d). In this way, we can obtain a denoised training dataset for robust alignment.

**Algorithm 1** Robust Alignment via Perplexity-aware Correction (PerpCorrect)

```
1: Input: Noisy training dataset $\tilde{\mathcal{D}}$, clean validation dataset $\mathcal{D}_{\mathrm{val}}$, and pre-trained LLM $\pi_\theta$ parameterized by $\theta$
2: Output: Robust alignment model $\pi_\theta$
3: // Stage I: Supervised fine-tuning (SFT)
4: $\pi_\theta \leftarrow$ Supervised fine-tuned LLM $\pi_\theta$. (Details in Appendix C.3)
5: // Stage II: Perplexity-aware correction using the surrogate LLM
6: $\tilde{\mathcal{D}}_{\mathrm{denoised}}, \varepsilon'_{\mathrm{denoised}} \leftarrow$ Perplexity-aware Correction ($\pi_\theta$, $\tilde{\mathcal{D}}$, $\mathcal{D}_{\mathrm{val}}$) (Details in Algorithm 2)
7: // Stage III: Alignment with denoised dataset
8: $\pi_\theta \leftarrow$ Aligned LLM $\pi_\theta$ using $\tilde{\mathcal{D}}_{\mathrm{denoised}}$ and $\varepsilon'_{\mathrm{denoised}}$ (Details in Appendix C.3)
```

Further, we select an optimal denoised training dataset to further enhance the performance of robust alignment according to the intersection area of the two normal distributions. We denote the intersection area of two normal distributions as the estimated NP proportion of the denoised training dataset, i.e.,

$$\varepsilon'_{PC} = \int_{-\inf}^{+\inf} \min\{(1-\bar{\varepsilon}) f_{\mathrm{clean}}(x|\bar{\mu}, \bar{\sigma}^2), \bar{\varepsilon} f_{\mathrm{noisy}}(x|-\bar{\mu}, \bar{\sigma}^2)\} \mathrm{d}x, \tag{15}$$

where $\varepsilon'_{PC}$ calculates the ratio of noisy data points which are not detected by PerpCorrect (i.e., the green area enclosed by the black and red lines in Figure 2d) and the clean data points which are mistakenly detected by PerpCorrect (i.e., the red area enclosed by the black and green lines in Figure 2d). In this way, $\varepsilon'_{PC}$ can efficiently calculate the NP proportion of the denoised training dataset. We take the denoised training dataset with the smallest $\varepsilon'_{PC}$ among multiple iterations as the optimal one for robust alignment to boost alignment performance.

### 3.2 Robust Alignment

Here, we introduce how to adapt PerpCorrect to robustify various alignment methods and demonstrate the algorithm of robust alignment via PerpCorrect in Algorithm 1. In general, the pipeline of the robust alignment based on PerpCorrect contains three stages: SFT, PerpCorrect, and alignment. We will first conduct SFT, following Christiano et al. [9], to boost the performance of a pre-trained LLM by boosting its skills for specific tasks. Next, we will conduct PerpCorrect to detect and correct NPs and output an optimal denoised training dataset $\tilde{\mathcal{D}}_{\mathrm{denoised}}$ the smallest $\varepsilon'_{PC}$ in Eq. 15. Finally, we can obtain an aligned LLM from the SFT model using the denoised training dataset $\tilde{\mathcal{D}}_{\mathrm{denoised}}$ via alignment (i.e., Line 8 in Algorithm 1).

Because our proposed PerpCorrect is agnostic to alignment methods and model structures, PerpCorrect is applicable to robustify both online alignment methods such as RLHF (PPO) [9] and offline alignment methods including DPO [26], SLiC [38], and IPO [3]. Besides, our proposed PerpCorrect is compatible with existing loss-oriented robust alignment methods, such as cDPO [21] and rDPO [8], based on the estimated proportion of NPs. Note that cDPO and rDPO require conducting computationally expensive cross-validation to tune the estimated proportion of NPs. We can efficiently estimate the proportion of NPs by utilizing the fitted normal distributions during PerpCorrect, i.e., $\varepsilon'_{PC}$ in Eq. 15. Therefore, we can combine PerpCorrect with a wide range of existing alignment methods to achieve robust alignment with NPs.

## 4 Experiments

In this section, we demonstrate that our proposed PerpCorrect achieves state-of-the-art alignment performance under different proportion of NPs and have good compatibility with other alignment methods. In Section 4.1, PerpCorrect combined with DPO [26] achieves state-of-the-art alignment performance than existing baselines (Section 4.1), including DPO [26], cDPO [21], and rDPO [8]. In Section 4.2, we further analyze the impact of the number of validation data and verified the compatibility of PerpCorrect with online and offline alignment methods and robust alignment methods. The training details and compute resources are reported in Appendix C.1.

Table 1: Average reward accuracy of DPO series alignment methods using Llama2-7B on the Golden HH dataset. The standard deviation of reward accuracy is reported in Table 8.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| Vanilla DPO | 92.53% | 82.62% | 68.50% | 53.15% |
| cDPO | 96.04% | 90.85% | 83.23% | 65.60% |
| rDPO | 96.65% | 95.22% | 93.90% | 90.45% |
| PerpCorrect-DPO | **97.51%** | **96.24%** | **95.53%** | **94.92%** |

Table 2: Average reward accuracy of PPO series alignment methods using Llama2-7B on the Golden HH dataset. The standard deviation of reward accuracy is reported in Table 9.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| Vanilla PPO | **96.64%** | 92.71% | 90.21% | 86.29% |
| cPPO | 96.18% | 93.63% | 90.62% | 88.02% |
| rPPO | 95.92% | 93.73% | 92.05% | 90.62% |
| PerpCorrect-PPO | 96.38% | **94.04%** | **93.99%** | **93.17%** |

Table 3: Average reward accuracy of DPO series alignment methods using phi-2 on the Golden HH dataset. The standard deviation of reward accuracy is reported in Table 10.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| Vanilla DPO | 93.19% | 85.57% | 73.07% | 54.98% |
| cDPO | 97.21% | 92.63% | 81.05% | 66.72% |
| rDPO | 96.49% | 95.73% | 93.34% | 84.55% |
| PerpCorrect-DPO | **98.17%** | **97.05%** | **97.66%** | **96.39%** |

Table 4: Average reward accuracy of DPO series alignment methods using phi-2 on the OASST1 dataset. The standard deviation of reward accuracy is reported in Table 11.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| Vanilla DPO | 66.94% | 62.61% | 58.44% | 52.42% |
| cDPO | 67.30% | 61.44% | 54.87% | 49.21% |
| rDPO | 63.95% | 59.47% | 56.45% | 45.20% |
| PerpCorrect-DPO | **71.34%** | **69.04%** | **68.27%** | **68.49%** |

**Datasets.** We utilize two preference datasets, namely OpenAssistant Conversations (OASST1) [17] and Golden HH [7]. The processed OASST1 dataset comprises 17,939 training samples and 951 testing samples and the processed Golden HH dataset consists of 12,066 training samples and 654 testing samples. The description and processing details of these datasets are provided in Appendix C.2.

**Models.** Our evaluation leverages two distinct series of open-sourced LLMs with different parameter sizes: Llama2-7B [32] and phi-2 [20]. We acquire the checkpoints from their official repositories on Hugging Face. The LLMs used for PerpCorrect and those for robust alignment share the same model structure and initialization.

**Baselines.** We adopt vanilla DPO [26] and two robust alignment methods, cDPO [21] and rDPO [8], as baselines. For their detailed implementation, we utilize and adapt the transformers and TRL libraries provided by the Hugging Face community.

**Metrics.** In accordance with Chowdhury et al. [8], we employ the winning rate of policy generations against the selected preferences on the test dataset as our primary metric. This metric applies to vanilla DPO [26], cDPO [21], rDPO [8], as well as other offline alignment methods including SLiC [38] and IPO [3]. Additionally, we utilize the winning rate of the reward model score for the chosen preferences on the test dataset as our metric for vanilla PPO [24], cPPO [21, 34], and rPPO [8]. These two metrics are collectively called reward accuracy.

### 4.1 PerpCorrect Achieves the State-of-the-Art Robust Alignment Performance

The empirical results demonstrate that our method, PerpCorrect, achieves state-of-the-art robust alignment performance, surpassing existing baselines such as vanilla DPO [26], cDPO [21], and rDPO [8]. This is evident across various proportions of noisy preferences $\varepsilon$ using different datasets and LLMs.

**Comparison using different LLMs.** Tables 1 and 3 demonstrate the average reward accuracy of the DPO series alignment methods on the Golden HH [7] dataset using Llama2-7B [32] and phi-2 [20]. At a proportion of the NPs $\varepsilon = 40\%$, PerpCorrect increases the reward accuracy by 41.77% (from 53.15% to 94.92%) using Llama2-7B and by 41.41% (from 54.98% to 96.39%) using phi-2. The empirical result validates that our proposed PerpCorrect can be used on different sizes of LLMs and achieve better alignment performance than baselines.

**Comparison on different datasets.** Tables 3 and 4 demonstrate the average reward accuracy of the DPO series alignment methods on the Golden HH [7] and OASST1 [17] datasets using phi-2 [20].

Table 5: Impact of the number of clean validation data evaluated on the Golden HH dataset using Llama2-7B with a proportion of NPs $\varepsilon = 40\%$.

| Number | 10 | 20 | 30 | 40 | 50 | 100 | 200 |
|---|---|---|---|---|---|---|---|
| Reward accuracy | 81.40% | 88.26% | 94.21% | 94.21% | 95.43% | 95.43% | 96.04% |

Table 6: Average reward accuracy and improvements of the offline and robust alignment methods, as well as those combined with PerpCorrect, using Llama2-7B on the Golden HH dataset. The standard deviation of reward accuracy and improvements is reported in Table 12.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| DPO | 92.53% | 82.62% | 68.50% | 53.15% |
| PerpCorrect-DPO | 97.51% | 96.24% | 95.53% | 94.92% |
| Δ | **+4.98%** | **+13.62%** | **+27.03%** | **+41.77%** |
| SLiC | 96.70% | 87.75% | 76.17% | 58.59% |
| PerpCorrect-SLiC | 96.95% | 95.02% | 95.38% | 94.61% |
| Δ | **+0.25%** | **+7.27%** | **+19.21%** | **+36.02%** |
| IPO | 98.07% | 92.73% | 79.17% | 61.64% |
| PerpCorrect-IPO | **98.73%** | **97.66%** | **97.82%** | **97.56%** |
| Δ | **+0.66%** | **+4.93%** | **+18.65%** | **+35.92%** |
| cDPO | 96.04% | 90.85% | 83.23% | 65.60% |
| PerpCorrect-cDPO | 98.12% | 97.31% | 94.97% | 88.36% |
| Δ | **+2.08%** | **+6.46%** | **+11.74%** | **+22.76%** |
| rDPO | 96.65% | 95.22% | 93.90% | 90.45% |
| PerpCorrect-rDPO | 95.99% | 95.02% | 94.77% | 95.73% |
| Δ | -0.66% | -0.20% | **+0.87%** | **+5.28%** |

The empirical results reveal a significant discrepancy in average reward accuracy between the more complex OASST1 dataset and the Golden HH dataset. The performance of other robust alignment methods is found to be unsatisfactory on the OASST1 dataset, often not surpassing the vanilla DPO. In contrast, our method PerpCorrect consistently maintains strong alignment performance across varying proportions of noisy preferences. In general, our method PerpCorrect can achieve better alignment performance than baselines across different datasets.

### 4.2 Ablation Study

**Impact of the number of clean validation data.** Table 5 illustrates the impact of the number of clean validation data points. We conducted experiments on the Golden HH dataset using Llama2-7B with a proportion of NPs $\varepsilon = 40\%$. The empirical results indicate that as the number of clean validation data points increases, the performance of our method, PerpCorrect, also improves. However, when the number is too large, the improvement in performance is not obvious, and the cost of manual annotation significantly increases.

**Compatibility with online alignment method RLHF (PPO).** We adopt vanilla PPO [24], cPPO [21, 34], and rPPO [8] as baselines. Table 2 shows the alignment performance of PPO series alignment methods on the Golden HH [7] dataset using Llama2-7B. Although vanilla PPO has good performance when the proportion of NPs is low, it still declines significantly when the proportion is high. PerpCorrect maintains desirable alignment performances when the proportion of NPs is high. Our empirical results show that PerpCorrect has desirable compatibility with online alignment method RLHF (PPO).

**Compatibility with various offline alignmet methods.** Table 6 presents the average reward accuracy and improvements of original offline alignment methods compared to those combined with PerpCorrect. Our experiments, conducted on the Golden HH dataset using Llama2-7B, reveal that the reward accuracy of SLiC [38] and IPO [3] both significantly decrease as the proportion of NPs increases, similar to vanilla DPO [26]. However, our method PerpCorrect enhances their alignment performance across different proportions of NPs. Notably, IPO combined with PerpCorrect achieves the best alignment performance. We conjecture the main reason is that the proportion of NPs in the denoised dataset is very low and IPO performs better than other methods under a low proportion of NPs. These empirical results demonstrate that our method has good compatibility with various offline alignment methods.

**Compatibility with robust alignment methods.** Table 6 shows the average reward accuracy and improvements of robust alignment methods compared to those combined with PerpCorrect. Our method, PerpCorrect, can significantly enhance the performance of cDPO [21], and provide a modest improvement for rDPO [8] under almost all proportion of NPs. The empirical results show that our method has good compatibility with robust alignment methods.

## 5 Conclusions

This paper proposes a method called perplexity-aware correction (PerpCorrect), as an effective approach for robust alignment with noisy preferences (NPs). PerpCorrect utilizes a surrogate LLM to calculate a novel metric, PPLDiff, and further detects and corrects NPs from clean preferences (CPs) based on it. PerpCorrect consists of three steps: (1) PerpCorrect aligns a surrogate LLM using the clean validation dataset, enabling PPLDiff to distinguish between CPs and NPs. (2) PerpCorrect enhances the discrimination power of PPLDiff by aligning the surrogate LLM with more reliable training data. (3) PerpCorrect detects and corrects NPs from CPs based on a calculated threshold and obtains a denoised training dataset. The paper further proposes a robust alignment pipeline, consisting of three stages SFT, PerpCorrect, and alignment, to achieve robust alignment with NPs. The experimental results validate that PerpCorrect achieves state-of-the-art alignment performance and has good compatibility with other online, offline, and robust alignment methods. Therefore, PerpCorrect can be an effective method to mitigate the impact of NPs and can be used for robust alignment. Future research directions include: (1) Improving the time efficiency of PerpCorrect and (2) Reducing the amount of clean validation data required to achieve the same alignment performance.

## Limitations

We discuss some limitations of this work to stimulate further research in this direction. Our limitations mainly stem from two aspects: time efficiency issues caused by multiple calculations of PPLDiff and repeated training of a surrogate LLM, and the need for a validation dataset.

**Time efficiency.** Iteratively calculating the PPLDiff value for each data point and aligning a surrogate LLM is time-consuming. Selecting reliably training data and denoising the training dataset requires that the PPLDiff value be calculated for each data point during each epoch, which may cause unnecessary calculations for CPs and NPs that can already be clearly distinguished. Besides, aligning a surrogate LLM with same size as the LLM for alignment multiple times is time-consuming. The detailed discussion is in the Appendix B.

**Validation dataset.** PerpCorrect requires a validation dataset for aligning a surrogate LLM. However, manually annotating a validation dataset is complex and labor-intensive in practice. As shown in Table 5, there is a significant disparity in alignment performance when comparing the use of 10 clean samples to 50 clean samples. Exploring how to use fewer clean samples or even no clean samples to achieve the same or better performance is a problem worth further investigation.

## Acknowledgements

This research is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative, the funding BAS/1/1689-01-01, URF/1/4663-01-01, REI/1/5232-01-01, REI/1/5332-01-01, and URF/1/5508-01-01 from KAUST, and funding from KAUST - Center of Excellence for Generative AI, under award number 5940. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

# References

[1] 01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.ai, 2024. URL `https://arxiv.org/abs/2403.04652`.

[2] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models, 2023. URL `https://arxiv.org/abs/2311.16867`.

[3] Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences, 2023.

[4] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL `https://arxiv.org/abs/2204.05862`.

[5] Tim Baumgärtner, Yang Gao, Dana Alon, and Donald Metzler. Best-of-venom: Attacking rlhf by injecting poisoned preference data, 2024.

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

[7] Tianchi Cai, Xierui Song, Jiyan Jiang, Fei Teng, Jinjie Gu, and Guannan Zhang. Ulma: Unified language model alignment with human demonstration and point-wise preference, 2024.

[8] Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust DPO: Aligning language models with noisy feedback. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=yhpDKSw7yA`.

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

[10] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=TyFrPOKYXw`.

[11] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with $\mathcal{V}$-usable information. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/ethayarajh22a.html`.

[13] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization, 2024.

[14] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024.

[15] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1): S63–S63, 1977.

[16] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[17] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

[18] Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xbjSwwrQOe.

[19] Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. Trustworthy llms: a survey and guideline for evaluating large language models' alignment, 2024.

[20] Microsoft. Phi-2: The surprising power of small language models, 2023. URL https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models.

[21] Eric Mitchell. A note on dpo with noisy preferences and relationship to ipo, 2023. URL https://ericmitchell.ai/cdpo.pdf.

[22] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022. URL https://arxiv.org/abs/2112.09332.

[23] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

[24] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[25] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[26] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

[27] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[28] Damien Sileo. tasksource: Structured dataset preprocessing annotations for frictionless extreme multi-task learning and evaluation. *arXiv preprint arXiv:2301.05948*, 2023. URL `https://arxiv.org/abs/2301.05948`.

[29] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[30] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[31] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL `https://arxiv.org/abs/2408.00118`.

[32] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta,

Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[33] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL `https://arxiv.org/abs/2307.09288`.

[34] Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, Songyang Gao, Nuo Xu, Yuhao Zhou, Xiaoran Fan, Zhiheng Xi, Jun Zhao, Xiao Wang, Tao Ji, Hang Yan, Lixing Shen, Zhan Chen, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. Secrets of rlhf in large language models part ii: Reward modeling, 2024.

[35] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL `https://arxiv.org/abs/2407.10671`.

[36] Jingwei Yi, Rui Ye, Qisi Chen, Bin Benjamin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Open-source can be dangerous: On the vulnerability of value alignment in open-source LLMs, 2024. URL `https://openreview.net/forum?id=NIouO0C0ex`.

[37] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.

[38] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback, 2023.

[39] Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. Secrets of rlhf in large language models part i: Ppo, 2023.

[40] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

# A Broader Impacts

Our proposed PerpCorrect and robust alignment pipeline offers a solution for achieving state-of-the-art performance in robust alignment under noisy preferences. PerpCorrect is designed to effectively reduce malicious noise in the dataset and mitigate biases introduced by human annotators, ensuring that the trained language model (LLM) is accurately aligned with true human preferences.

Moreover, we recognize a potential risk: if malicious users exploit our method for reverse training, they might compromise the security mechanisms of existing open-source LLMs. Existing research has demonstrated the possibility of reverse training [36].

# B Time Efficiency Analysis

The additional computational overhead is primarily attributed to PerpCorrect (Section 3.1). Table 7 presents both theoretical and empirical runtime comparisons, where $X$ represents the theoretical time required for Alignment (Section 3.2) or other baselines.

In theory, during the PerpCorrect process, we need to calculate PPLDiff and train the surrogate model in each epoch. The computation time introduced by PerpCorrect is approximately $\frac{T}{3}$ that of the Alignment or other baselines.

The calculation of PPLDiff in each epoch requires only $\frac{1}{3}$of the time needed for robust alignment. The primary computational load in robust alignment arises from the complexity of forwarding and back-propagation, while the complexities of gradient updates and parameter updates are relatively low. Additionally, back-propagation takes twice as long as forwarding. In addition, the calculation of PPLDiff only requires forwarding.

For surrogate model training, PerpCorrect utilized data points that represented $t \times \alpha$ of the total dataset during epoch $t$. Since both $t$ and $\alpha$ are small, the time required for surrogate model training can be approximately ignored.

In practice, Our entire robust alignment pipeline (∼24 hours) takes only twice as long as the baseline (∼12 hours). We set $T = 5$ and $\alpha = 2$, and used the AdamW optimizer. The practical efficiency of the PerpCorrect is due to the use of fp32 precision by the AdamW optimizer, which increases the GPU's calculation time during the robust alignment process.

Table 7: Comparison of theoretical and practical running times for PerpCorrect and baselines.

| Stage | Theoretical Running Time | Practical Running Time |
|---|---|---|
| PerpCorrect | $\frac{T}{3} \times X$ | ∼12 hours |
| Alignment (or baselines) | $X$ | ∼12 hours |
| Total | $(1 + \frac{T}{3})X$ | ∼24 hours |

# C Implementation details

## C.1 Training details and compute resources.

We utilized the Qlora method [11] for fine-tuning the LLMs, executed on RTX 4090 GPUs with 24 GB of memory. Hyperparameters were set as follows: lora_rank $=32$, lora_dropout $=0.1$, and lora_alpha $=16$. For SFT, we use the alpaca dataset [30] and set learning_rate $=2e-4$ and batch_size $=20$. For our PerpCorrect stage II, we set $\beta = 0.1$, learning_rate $=1e-3$, batch_size $=4$, $T = 5$, and $\alpha = 0.02$. For our PerpCorrect stage III and all other alignment methods, we set $\beta = 0.1$, learning_rate $=3e-4$, and batch_size $=20$. Other details not mentioned, we follow the default setting in TRL library. Each experiment, involving a specific method and proportion of NPs, could be completed using a single RTX 4090 GPU within 24 hours on the Golden HH dataset and within 72 hours on the OASST1 dataset.

### C.2 Description and Processing Details of the Datasets

**OpenAssistant Conversations Dataset (OASST1).** The original OASST1 dataset [17] is an assistant-style conversation corpus generated and annotated by humans. It consists of over 10,000 fully annotated conversations in 35 different languages. Sileo [28] converted these conversations into a preference dataset comprising 17,966 training samples and 952 testing samples. After filtering out conversations with one or fewer letters, we obtained a preference dataset with 17,939 training samples and 951 testing samples.

**Golden HH.** The Golden HH dataset [7] is a variant of the Anthropic Helpful and Harmless (HH) dataset [4]. It originally contains 42,537 training samples and 2,312 testing samples as part of a preference dataset. Each sample has two keys: one representing the prompt $x$ and the chosen response $y_w$, and the other representing the prompt $x$ and the rejected response $y_l$. We first converted the dataset into a triple form: prompt $x$, chosen response $y_w$, and rejected response $y_l$, retaining only one-turn conversation data. After filtering out samples with one or fewer letters, we obtained a preference dataset with 12,066 training samples and 654 testing samples.

### C.3 Detailed Robust Alignment via Perplexity-aware Correction

**Supervised Fine-Tuning (SFT).** The objective of Supervised Fine-Tuning (SFT) is to enhance the performance of a pre-trained large language model (LLM) by refining its abilities for specific tasks. As demonstrated by prior work [9, 25, 24], this can be achieved by utilizing supervised fine-tuning with a specialized dataset tailored to the target task. The SFT dataset is annotated with labels, providing examples that are directly relevant to the task. Specifically, for each data point $(x, y)$ in the SFT dataset, $x$ represents the prompt given to the LLM, and $y$ represents the expected response that the model should generate based on the prompt $x$. The process involves fine-tuning the LLM by maximizing the log-likelihood of the correct responses $y$ given the prompts $x$. Through this method, the model learns to produce more accurate and task-specific outputs, thereby significantly improving its performance on the given task.

**Perplexity-aware Correction (PerpCorrect).** We demonstrate the entire PerpCorrect algorithm in Algorithm 2.

**Alignment.** We can achieve alignment using the denoised training dataset $\tilde{\mathcal{D}}_{\text{denoised}}$ with an estimated proportion of NPs $\varepsilon'_{\text{denoised}}$. For offline alignment methods such as DPO, SLiC, and IPO, we can directly optimize the LLM using the denoised training dataset $\tilde{\mathcal{D}}_{\text{denoised}}$ based on the loss functions defined in Eqs. 6–8. For loss-based robust alignment methods, including cDPO and rDPO, we set $\varepsilon' = \varepsilon'_{\text{denoised}}$ and then optimize the LLM using the denoised training dataset $\tilde{\mathcal{D}}_{\text{denoised}}$ according to the loss functions mentioned in Eqs. 9 and 10. For the online alignment method RLHF (PPO), we first train a reward model using the denoised training dataset $\tilde{\mathcal{D}}_{\text{denoised}}$ based on the loss function described in Eq. 3. Subsequently, we further optimize the LLM using PPO according to the objective function detailed in Eq. 4.

## D Extended Experimental Results

### D.1 Experiment Statistical Significance

Tables 8–12 demonstrates the standard deviation of the reward accuracy reported in Tables 1–4 and 6.

### D.2 Average PPLDiff Values of Data from Different Datasets Calculated by Unaligned LLMs

We randomly selected 10,000 data points from each dataset and calculated PPLDiff using different LLMs. The datasets and LLMs are downloaded from the Huggingface website. The average PPLDiff values are reported in the Table 13.

Table 8: Standard deviation of reward accuracy for DPO series alignment methods using Llama2-7B on the Golden HH dataset. The average reward accuracy is reported in Table 1.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| vanilla DPO | 0.81% | 0.40% | 2.52% | 2.60% |
| cDPO | 1.15% | 0.81% | 1.76% | 1.64% |
| rDPO | 0.26% | 1.53% | 0.95% | 1.92% |
| PerpCorrect-DPO | 0.63% | 0.87% | 1.73% | 0.63% |

Table 9: Standard deviation of reward accuracy for PPO series alignment methods using Llama2-7B on the Golden HH dataset. The average reward accuracy is reported in Table 2.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| vanilla PPO | 0.15% | 1.30% | 4.05% | 0.77% |
| cPPO | 0.15% | 1.53% | 4.61% | 5.89% |
| rPPO | 0.62% | 1.38% | 1.55% | 5.29% |
| PerpCorrect-PPO | 0.35% | 1.15% | 1.34% | 1.57% |

Table 10: Standard deviation of reward accuracy for DPO series alignment methods using phi-2 on the Golden HH dataset. The average reward accuracy is reported in Table 3.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| vanilla DPO | 0.92% | 0.63% | 1.28% | 0.98% |
| cDPO | 0.35% | 0.54% | 0.49% | 1.84% |
| rDPO | 0.30% | 0.15% | 1.67% | 7.82% |
| PerpCorrect-DPO | 0.81% | 1.27% | 0.63% | 1.07% |

Table 11: Standard deviation of reward accuracy for DPO series alignment methods using phi-2 on the OASST1 dataset. The average reward accuracy is reported in Table 4.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| vanilla PPO | 0.70% | 1.30% | 2.46% | 1.64% |
| cPPO | 0.46% | 0.87% | 0.56% | 0.91% |
| rPPO | 0.96% | 2.29% | 1.24% | 6.61% |
| PerpCorrect-PPO | 0.35% | 1.89% | 1.35% | 1.06% |

Table 12: Standard deviation of reward accuracy and improvements of the offline and robust alignment methods, as well as those combined with PerpCorrect, using Llama2-7B on the Golden HH dataset. The average reward accuracy is reported in Table 6.

| Method | Proportion of noisy preferences (%) | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| DPO | 0.81% | 0.40% | 1.28% | 0.98% |
| PerpCorrect-DPO | 0.63% | 0.87% | 1.73% | 0.63% |
| Δ | 0.98% | 0.89% | 0.97% | 1.98% |
| SLiC | 1.91% | 1.33% | 6.77% | 8.02% |
| PerpCorrect-SLiC | 1.76% | 1.45% | 1.16% | 1.50% |
| Δ | 0.63% | 0.38% | 5.62% | 6.52% |
| IPO | 0.84% | 0.63% | 3.48% | 0.32% |
| PerpCorrect-IPO | 0.32% | 1.68% | 0.23% | 1.25% |
| Δ | 0.98% | 2.29% | 3.26% | 1.10% |
| cDPO | 1.15% | 0.81% | 1.76% | 1.64% |
| PerpCorrect-cDPO | 0.69% | 0.75% | 0.81% | 0.72% |
| Δ | 0.47% | 1.31% | 1.85% | 2.36% |
| rDPO | 0.26% | 1.53% | 0.95% | 1.92% |
| PerpCorrect-rDPO | 0.63% | 0.23% | 1.84% | 1.19% |
| Δ | 0.84% | 1.48% | 2.19% | 2.31% |

Table 13: Average PPLDiff values of randomly selected data points across datasets calculated by different LLMs. "Avg." refers to the average PPLDiff value over all the datasets.

| Model | Dataset | | | | Avg. |
|---|---|---|---|---|---|
| | HH-RLHF [4] | SafeRLHF [10] | SHP [12] | WebGPT [22] | |
| Qwen2-1.5B [35] | -0.140 | -0.002 | 0.040 | -0.018 | -0.030 |
| Qwen2-1.5B-Instruct [35] | -0.149 | -0.009 | 0.046 | -0.021 | -0.033 |
| Yi-1.5-6B [1] | -0.158 | -0.103 | 0.105 | -0.036 | -0.048 |
| Yi-1.5-6B-Chat [1] | -0.159 | -0.054 | 0.069 | -0.040 | -0.046 |
| gemma-2-2b [31] | -0.140 | -0.051 | 0.001 | -0.024 | -0.053 |
| gemma-2-2b-it [31] | -0.163 | -0.053 | 0.063 | -0.028 | -0.045 |
| falcon-7b [2] | -0.113 | -0.001 | 0.037 | -0.016 | -0.023 |
| falcon-7b-instruct [2] | -0.121 | 0.021 | 0.039 | -0.017 | -0.019 |
| Mistral-7B-v0.3 [16] | -0.133 | -0.045 | 0.048 | -0.026 | -0.039 |
| Mistral-7B-Instruct-v0.3 [16] | -0.201 | -0.058 | 0.065 | -0.038 | -0.058 |
| glm-4-9b [14] | -0.134 | -0.019 | 0.045 | -0.027 | -0.034 |
| glm-4-9b-chat-1m [14] | -0.135 | -0.019 | 0.049 | -0.028 | -0.033 |
| Llama-2-7b-hf [33] | -0.133 | -0.052 | 0.051 | -0.028 | -0.040 |
| Llama-2-7b-chat-hf [33] | -0.139 | -0.041 | 0.063 | -0.032 | -0.037 |

**Algorithm 2** Perplexity-aware Correction (PerpCorrect)

```
Input: Noisy training dataset $\tilde{\mathcal{D}}$, clean validation dataset $\mathcal{D}_{\text{val}}$, LLM $\pi_\theta$ parameterized by $\theta$
Output: Denoised training dataset $\tilde{\mathcal{D}}_{\text{denoised}}$ and estimated proportion of NPs $\varepsilon'_{\text{denoised}}$
$\pi_{\theta'} \leftarrow \pi_\theta$, $\mathcal{D}'_0 \leftarrow \emptyset$, $\varepsilon'_{\text{denoised}} \leftarrow 1$, $\tilde{\mathcal{D}}_{\text{denoised}} \leftarrow \tilde{\mathcal{D}}$,
for epoch $t = 0, \ldots, T$ do
  // Aligning the surrogate LLM
  $\pi_{\theta'} \leftarrow$ Alignment ($\pi_{\theta'}$, $\mathcal{D}'_t \cup \mathcal{D}_{\text{val}}$)
  // Calculating the PPLDiff values for each data point
  $\Omega \leftarrow \emptyset$
  for $(\tilde{x}, \tilde{y}_w, \tilde{y}_l) \in \tilde{\mathcal{D}}$ do
    $z \leftarrow \log \text{PPL}(x + \tilde{y}_w; \theta') - \log \text{PPL}(x + \tilde{y}_l; \theta')$
    $\Omega \leftarrow \Omega \cup \{(\tilde{x}, \tilde{y}_w, \tilde{y}_l, z)\}$
  end for
  // Fitting PPLDiff density of noisy training dataset
  $\bar{\varepsilon}$, $\bar{\mu}$, $\bar{\sigma} \leftarrow$ Fitted parameters using Levenberg-Marquard algorithm with $\Omega$
  // Estimating NPs proportion of the denoised training dataset
  $\varepsilon'_{PC} \leftarrow$ Estimated proportion of NPs using the Eq.15 based on $\bar{\varepsilon}$, $\bar{\mu}$, $\bar{\sigma}$
  // Keeping denoised training dataset with the smallest $\varepsilon'_{\text{denoised}}$
  if $\varepsilon'_{PC} < \varepsilon'_{\text{denoised}}$ then
    $\varepsilon'_{\text{denoised}} \leftarrow \varepsilon'_{PC}$
    // Calculating the Threshold $\tau$
    $\tau \leftarrow$ x-coordinate of the intersection of the two normal distributions($\bar{\varepsilon}$, $\bar{\mu}$, $\bar{\sigma}$)
    // Distinguishing CPs and NPs based on the threshold $\tau$ and correcting NPs
    $\tilde{\mathcal{D}}_{\text{CPs}} \leftarrow \emptyset$, $\tilde{\mathcal{D}}_{\text{NPs}} \leftarrow \emptyset$
    for $(\tilde{x}, \tilde{y}_w, \tilde{y}_l, z) \in \Omega$ do
      if $z > \tau$ then
        $\tilde{\mathcal{D}}_{\text{CPs}} \leftarrow \tilde{\mathcal{D}}_{\text{CPs}} \cup \{(\tilde{x}, \tilde{y}_w, \tilde{y}_l)\}$
      else
        $\tilde{\mathcal{D}}_{\text{NPs}} \leftarrow \tilde{\mathcal{D}}_{\text{NPs}} \cup \{(\tilde{x}, \tilde{y}_l, \tilde{y}_w)\}$
      end if
    end for
    $\tilde{\mathcal{D}}_{\text{donised}} \leftarrow \tilde{\mathcal{D}}_{\text{CPs}} \cup \tilde{\mathcal{D}}_{\text{NPs}}$
  end if
  $\mathcal{D}_{\text{Clean}} \leftarrow \emptyset$, $\mathcal{D}_{\text{Noisy}} \leftarrow \emptyset$
  // Calculating the left bound $\tau_l$ and the right bound $\tau_r$
  $\tau_l \leftarrow (t-1) \cdot \alpha \cdot (1 - \bar{\varepsilon}) \cdot |\tilde{\mathcal{D}}|$-th smallest PPLDiff value in $\Omega$
  $\tau_r \leftarrow (t-1) \cdot \alpha \cdot \bar{\varepsilon} \cdot |\tilde{\mathcal{D}}|$-th largest PPLDiff value in $\Omega$
  // Finding extra reliable training data
  for $(\tilde{x}, \tilde{y}_w, \tilde{y}_l, z) \in \Omega$ do
    if $z < \tau_l$ then
      $\mathcal{D}_{\text{Clean}} \leftarrow \mathcal{D}_{\text{Clean}} \cup \{(\tilde{x}, \tilde{y}_w, \tilde{y}_l)\}$
    end if
    if $z > \tau_r$ then
      $\mathcal{D}_{\text{Noisy}} \leftarrow \mathcal{D}_{\text{Noisy}} \cup \{(\tilde{x}, \tilde{y}_l, \tilde{y}_w)\}$
    end if
  end for
  $\mathcal{D}'_{t+1} \leftarrow \mathcal{D}_{\text{Clean}} \cup \mathcal{D}_{\text{Noisy}}$
end for
```

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our introduction covers our contributions, main methods and experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the efficiency issues and data volume requirements of our method PerpCorrect in the Conclusions section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We show all the experiment detail in the Experiments section and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to our code using Github.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We show the entire experimental details in the Experiments section and Appendix and provide open access to the code using Github.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the standard deviation in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide detailed sufficient information on the computer resources in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential impacts in Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our robust alignment method does not have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The models and datasets the we used are open-sourced, and we follow their license and terms of use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our training code are open-source on GitHub.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.