Correcting Large Language Model Behavior via Influence Function

Han Zhang[1,2], **Zhuo Zhang**[1,2], **Yi Zhang**[2], **Yuanzhao Zhai**[3], **Hanyang Peng**[2], **Yu Lei**[2], **Yue Yu**[2], **Hui Wang**[2], **Bin Liang**[4], **Lin Gui**[*5], **Ruifeng Xu**[*1,2,6]

[1] Harbin Institute of Technology (Shenzhen),
[2] Pengcheng Laboratory,
[3] National University of Defense Technology,
[4] The Chinese University of Hong Kong,
[5] King's College London,
[6] Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
hanlardresearch@gmail.com, iezhuo17@gmail.com, zhangy12@pcl.ac.cn, yuanzhaozhai@nudt.edu.cn, penghy@pcl.ac.cn, leiy01@pcl.ac.cn, yuy@pcl.ac.cn, wangh06@pcl.ac.cn, bin.liang@cuhk.edu.hk, lin.1.gui@kcl.ac.uk, xuruifeng@hitsz.edu.cn

Abstract

Recent advancements in AI alignment techniques have significantly improved the alignment of large language models (LLMs) with static human preferences. However, the dynamic nature of human preferences can render some prior training data outdated or even erroneous, ultimately causing LLMs to deviate from contemporary human preferences and societal norms. Existing methodologies, either curation of new data for continual alignment or manual correction of outdated data for re-alignment, demand costly human resources. To address this, we propose a novel approach, **L**LM Beh**A**vior Correction with I**N**fluence Fun**C**tion R**E**call and Post-**T**raining (LANCET), which needs no human involvement. LANCET consists of two phases: (1) using a *new* method LinFAC to efficiently identify the training data that significantly impact undesirable model outputs, and (2) applying an *novel* **I**nfluence-driven **B**regman **O**ptimization (IBO) technique to adjust the model's outputs based on these influence distributions. Our experiments show that LANCET effectively and efficiently corrects inappropriate behaviors of LLMs while preserving model utility. Furthermore, LANCET exhibits stronger generalization ability than all baselines under out-of-distribution harmful prompts, offering better interpretability and compatibility with real-world applications of LLMs.

1 Introduction

Recent advancements in AI alignment techniques, such as Reinforcement Learning from Human Feedback (RLHF) (Stiennon et al. 2020; Bai et al. 2022a; Ouyang et al. 2022) and Direct Preference Optimization (DPO) (Rafailov et al. 2023), have made significant strides in aligning large language models (LLMs) with human preferences using static alignment datasets. However, human preferences are inherently dynamic, evolving over time and rendering some training data outdated or erroneous, particularly those reflecting values now deemed inappropriate (Carroll et al. 2024; Zhang et al. 2024). For example, the film "Gone with the Wind," once celebrated, has since faced criticism for its portrayal of race. Such discrepancies can cause model behaviors to diverge from contemporary human preferences and societal norms. Correcting the anachronistic behavior of LLMs due to learned outdated preferences is critical to enhancing their real-world applicability and ensuring adherence to evolving human values and norms.

Existing methodologies for correcting LLMs behaviors typically involve the meticulous curation of new preference data for continual alignment (Carroll et al. 2024; Zhang et al. 2024) or the manual correction of outdated data for re-alignment (Jaques et al. 2020; Kreutzer et al. 2018; Bai et al. 2022b). It remains uncertain whether newly curated data can "really" override the influences of outdated data on LLMs. In practice, both methods demand significant human resources, which are expensive and time-consuming. In response to these limitations, we propose investigating a novel and practical problem: *how can we correct LLMs behavior without costly human resources?* We seek a promising approach to address this challenge: enabling LLMs to autonomously retrieve inappropriate data from the original training dataset that significantly impacts the undesirable outputs of LLMs. Subsequently, the LLMs self-correct their behavior after training on retrieved data. This process is illustrated in Figure 1, given the misaligned behaviors of LLMs, LLMs identify the data responsible for these behaviors and use it for self-correction.

Following this spirit of this idea, we propose a practical model behavior correction algorithm, **L**LM beh**A**vior correction with i**N**fluence fun**C**tion r**E**call and post-**T**raining, abbreviated as LANCET. LANCET comprises two phases: (1) it leverages influence functions (Hampel 1974) to identify training data that most significantly affects undesirable model behavior. Traditional influence functions often suffer from prohibitive computational costs when applied to LLMs (Grosse et al. 2023). We propose a new influence function calculation method, LinFAC, which employs linear approximations (Jacot, Hongler, and Gabriel 2018; Ortiz-Jiménez, Moosavi-Dezfooli, and Frossard 2021) to reduce computational overhead and time complexity significantly. (2) We introduce an innovative Influence-driven Bregman Optimization (IBO) technique, which utilizes the distribution of influence scores to rectify inappropriate model behavior. In contrast to previous gradient ascent-driven model unlearning

*Corresponding authors.

Figure 1: The Framework of AI Behavior Correction.

algorithms (Yao, Xu, and Liu 2024), IBO employs pairwise objective to learn the ranking information provided by LinFAC. Our contributions are as follows:

1. We propose a pioneering attempt to correct LLMs' behavior from the perspective of influence functions, which seek to minimize reliance on costly human resources.
2. We present a practical algorithm, named LANCET, which leverages influence functions to identify the most influenced inappropriate data and systematically shapes LLMs behaviors during subsequent self-correcting phases.
3. Through extensive experimentation, we demonstrate that LANCET significantly corrects inappropriate behaviors of LLMs while preserving model utility and improving interpretability. Moreover, LANCET exhibits superior generalization over all baselines, markedly reducing unsafe behaviors in response to out-of-distribution prompts.

2 Preliminaries and Related Work

Problem Statement. Suppose that an accessible training set $\mathbb{D}_t = \{z_i\}_{i=1}^N$ that contains some inappropriate samples (e.g., outdated or incorrect data) and each training sample z_i is composed of prompt and response $z_i = (x_i, y_i)$. LLM $\pi_{\boldsymbol{\theta}}$ are trained on $\mathbb{D}_t$ with parameters $\boldsymbol{\theta}$. Due to these inappropriate training samples, π_θ may generate undesirable outputs z_r that do not align with current human preferences or social norms given some prompts z_p. We denote the undesirable behaviors $\mathbb{D}_q = \{z_i\}_{i=1}^Q$ as *Influence Queries* (IQs). Because of the costly human correction, our research focuses on self-correcting undesirable behaviors in LLMs via influence function without requiring extensive human intervention.

Influence Function. Influence function (IF) (Hampel 1974) aims to find the training example that most contributes to a given behavior.

To calculate the influence score of a trained sample $z_m \in \mathbb{D}_t$ to a given behavior (i.e., influence query) z_q, IF first defines the response function:

$$\boldsymbol{\theta}^*(\epsilon) = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(z_i, \boldsymbol{\theta}) + \epsilon \mathcal{L}(z_m, \boldsymbol{\theta}), \quad (1)$$

where $\mathcal{L}$ can be generally the autoregressive cross-entropy loss in LLMs: $\mathcal{L}(z; \boldsymbol{\theta}) = -\sum_{t=1}^{T} \log p(y_t | y_{1:t-1}, x; \boldsymbol{\theta})$.

The response function describes how the optimal model parameters $\boldsymbol{\theta}^*$ varies if the training weight ϵ of sample z_m changes. The influence function of z_m on $\boldsymbol{\theta}^*$ is defined as the gradient of the response function at $\epsilon = 0$:

$$\mathcal{I}_{\boldsymbol{\theta}^*}(z_m) \triangleq \left. \frac{d\boldsymbol{\theta}^*}{d\epsilon} \right|_{\epsilon=0}, \quad (2)$$

and the final influence score of z_m to z_q is calculated by (Grosse et al. 2023):

$$\mathcal{I}_{\boldsymbol{\theta}^*}(z_m, z_q) \triangleq \nabla_{\boldsymbol{\theta}^*} \log p(z_r | z_p; \boldsymbol{\theta}^*)^\top \mathcal{I}_{\boldsymbol{\theta}^*}(z_m), \quad (3)$$

where $p(z_r | z_p; \boldsymbol{\theta}^*)$ denotes the probability of the influence query. According to the chain rule, the influence score can be written as $\mathcal{I}_{\boldsymbol{\theta}^*}(z_m, z_q) = \left. \frac{d}{d\epsilon} \log p(z_r | z_p; \boldsymbol{\theta}^*) \right|_{\epsilon=0}$. $\mathcal{I}_{\boldsymbol{\theta}^*}(z_m, z_q)$ describes the degradation of $p(z_r | z_p; \boldsymbol{\theta}^*)$ if removing z_m from $\mathbb{D}_t$, and can be considered as the contribution of z_m to z_q. It is noteworthy that the influence score can be negative values, which implies that removing z_m from $\mathbb{D}_t$ will increase the probability of the z_q.

Proximal Bregman Response Function. Previous work has shown that applying the influence function defined by Eq. 2 to modern neural networks has a large bias and error (Bae et al. 2022). To address this problem, Bae et al. (2022) proposes the Proximal Bregman Response Function (PBRF) with respect to the Proximal Bregman Objective (PBO):

$$\boldsymbol{\theta}^s(\epsilon) = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} D_{\mathcal{L}}(\hat{y}, \hat{y}^s) + \epsilon \mathcal{L}(z_m, \boldsymbol{\theta}) + \frac{\lambda}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}^s\|^2, \quad (4)$$

where $\lambda > 0$ is the damping term, $\boldsymbol{\theta}^s$ is the original model parameters trained on $\mathbb{D}_t$, $\hat{y}$ (or $\hat{y}^s$) is the prediction probability under model parameters $\boldsymbol{\theta}$ (or $\boldsymbol{\theta}^s$), and $D_{\mathcal{L}}$ is the Bregman divergence:

$$D_{\mathcal{L}}(\hat{y}, \hat{y}^s) = \mathcal{L}(\hat{y}, y) - \mathcal{L}(\hat{y}^s, y) - \nabla_{\hat{y}} \mathcal{L}(\hat{y}^s, y)^\top (\hat{y} - \hat{y}^s), \quad (5)$$

the influence score with respect to PBRF is defined as:

$$\begin{aligned} \mathcal{I}_f(z_m, z_q) &\triangleq \nabla_{\boldsymbol{\theta}} \log p(z_r | z_p; \boldsymbol{\theta})^\top \left. \frac{d\boldsymbol{\theta}^s(\epsilon)}{\epsilon} \right|_{\epsilon=0} \\ &\approx -\underbrace{\nabla_{\boldsymbol{\theta}} \log p(z_r | z_p; \boldsymbol{\theta})^\top (\mathbf{G} + \lambda \mathbf{I})^{-1}}_{v_q^\top : \text{IHVP}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(z_m; \boldsymbol{\theta}), \end{aligned} \quad (6)$$

where $\mathbf{G}$ denotes the Gauss-Newton Hessian (GNH) and v_q denotes the Inverse Hessian Vector Product (IHVP).

Scale IF to LLMs by EK-FAC. Due to the tremendous dimension of $\mathbf{G}$ in LLMs, it is intractable to directly compute the influence score by Eq. 6. Theoretically, $\mathbf{G}$ equals the Fisher matrix $\mathcal{F} \triangleq \mathbb{E}[\mathcal{D}_{\boldsymbol{\theta}}\mathcal{D}_{\boldsymbol{\theta}}^{\top}]$ where $\mathcal{D}_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \log p(y|x;\boldsymbol{\theta})^{\top}$ denotes the pseudo-gradient with respect to $\boldsymbol{\theta}$. Based on this, George et al. (2018) proposes Eigenvalue-corrected Kronecker-Factored Approximate Curvature (EK-FAC) to efficiently approximate $\mathbf{G}$.

Suppose that a fully connected layer $f : \mathbb{R}^M \to \mathbb{R}^P$ has input activations $a \in \mathbb{R}^M$, parameters $W \in \mathbb{R}^{P\times M}$, and outputs $s \in \mathbb{R}^P$. Denote $\boldsymbol{w} = \text{vec}(W) \in \mathbb{R}^{PM}$ as the vectorization of W. According to the chain rule, the pseudo-gradient with respect to $\boldsymbol{w}$ can be formulated as $\mathcal{D}_{\boldsymbol{w}} = a \otimes \mathcal{D}_s$ where $\otimes$ denotes the Kronecker product. Therefore, the matrix $\mathbf{G} \in \mathbb{R}^{PM\times PM}$ can be approximated as:

$$\begin{aligned}\mathbf{G} &= \mathbb{E}[\mathcal{D}_{\boldsymbol{w}}\mathcal{D}_{\boldsymbol{w}}^{\top}] = \mathbb{E}[aa^{\top} \otimes \mathcal{D}s\mathcal{D}s^{\top}] \\ &\approx \mathbb{E}[aa^{\top}] \otimes \mathbb{E}[\mathcal{D}s\mathcal{D}s^{\top}] \triangleq \mathbf{A} \otimes \mathbf{S}, \end{aligned} \tag{7}$$

where $\mathbf{A} \in \mathbb{R}^{M\times M}$ and $\mathbf{S} \in \mathbb{R}^{P\times P}$. By employing eigendecompositions $\mathbf{A} = \mathbf{Q_A}\mathbf{\Lambda_A}\mathbf{Q_A}^{\top}$ and $\mathbf{S} = \mathbf{Q_S}\mathbf{\Lambda_S}\mathbf{Q_S}^{\top}$, the $\mathbf{G}$ can be further approximated by $\mathbf{G} \approx (\mathbf{Q_A} \otimes \mathbf{Q_S})\Lambda(\mathbf{Q_A} \otimes \mathbf{Q_S})^{\top}$ where Λ is a diagonal matrix of dimension MP defined as $\mathbf{\Lambda}_{ii} = \mathbb{E}[((\mathbf{Q_A} \otimes \mathbf{Q_S})\mathcal{D}_{\boldsymbol{w}})_i^2]$. Subsequently, the IHVP can be calculated as:

$$\begin{aligned}&(\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{v} \approx (\mathbf{Q_A} \otimes \mathbf{Q_S})(\mathbf{\Lambda} + \lambda\mathbf{I})^{-1}(\mathbf{Q_A} \otimes \mathbf{Q_S})^{\top}\mathbf{v} \\ &= \text{vec}(\mathbf{Q_S}^{\top}[(\mathbf{Q_S}\bar{V}\mathbf{Q_A}^{\top}) \oslash \text{unvec}(\text{diag}(\mathbf{\Lambda} + \lambda\mathbf{I}))]\mathbf{Q_A}),\end{aligned} \tag{8}$$

where $\oslash$ denotes elementwise division, $\text{unvec}(\cdot)$ is an inverse of the $\text{vec}(\cdot)$ operation, the gradient $\mathbf{v} = \nabla_{\boldsymbol{\theta}} \log p(z_r|z_p;\boldsymbol{\theta}) \in \mathbb{R}^{PM}$, and $\bar{V} = \text{unvec}(\mathbf{v}) \in \mathbb{R}^{P\times M}$. Substituting Eq. 8 into Eq. 6 yields the influence scores.

The EK-FAC decomposes the formidable computation of IHVP into the product of several smaller matrices, which significantly enhances the efficiency of calculating influence scores and is scalable to LLMs.

Model Unlearning. Although model unlearning (Yao, Xu, and Liu 2024) can also eliminate inappropriate behavior of LLMs, which has two drawbacks: (1) Gradient ascent lacks robustness and can easily harm model performance. (2) Observed unsafe behaviors may be insufficient, leading to inadequate generalization during training. Our method differs from it in that we trace back from the unsafe behavior of LLMs to the original training set to find samples having positive or negative influences. This not only eliminates but also corrects LLM's inappropriate behavior. In contrast, LLM-unlearning (Yao, Xu, and Liu 2024) merely performs unlearning on the inappropriate behavior data of LLMs, which can eliminate unsuitable behavior but poses a risk of insufficient generalization capability.

3 Methodology

This sectione elaborates LANCET, a novel method designed to automatically mitigate undesirable behaviors of LLMs, thereby minimizing the reliance on costly human resources. We introduce a novel influence score calculation method in Section 3.1, LinFAC, which enables efficient and accurate computation of influence scores for training samples. Subsequently, we elaborate on how to utilize IF-scored samples to correct and shape the behavior of LLMs in Section 3.2.

3.1 Linear Kronecker-Factored Approximate Curvature

Although EK-FAC can scale the IF to LLMs, it has notable limitations: (1) It handles each linear layer in isolation and assumes independence between input tokens, which leads to inaccurate influence scores; (2) It uses a linear layer as the computational unit, resulting in prolonged computation times. To address these challenges, we introduce a novel method LinFAC to calculate the influence score. LinFAC computes the Gauss-Newton Hessian using entire sequences, accounting for token interdependencies. Moreover, LinFAC uses the Transformer sublayer (e.g., Feedforward Networks) with multiple linear layers as the computational unit, enhancing the accuracy and efficiency of computations.

Suppose that the Transformer sublayer f with parameter $\boldsymbol{\theta} = \{\theta_i\}_{i=1}^{L}$ where θ_i denotes the i-th linear layer of f. LinFAC modularizes f as a linear layer with parameter $\hat{\boldsymbol{\theta}}$ which is the surrogate parameter of $\boldsymbol{\theta}$ without actually computing. Denote modular pre-activation output $s = f(\hat{\boldsymbol{\theta}}, a)$ with input state a. The influence score of z_m to z_q is calculated by:

$$\begin{aligned}\mathcal{I}_f(z_m, z_q) \triangleq -\underbrace{\overbrace{(\sum_t a_t^{z_q} \otimes \mathcal{D}s_t^{z_q})^{\top}}^{\text{modular gradient of } z_q}\overbrace{(\mathbb{E}[aa^{\top} \otimes \mathcal{D}s\mathcal{D}s^{\top}] + \lambda\mathbf{I})^{-1}}^{\hat{\mathbf{G}}\text{: modular } \mathbf{GNH}}}_{\hat{v}_q\text{: modular } \mathbf{IHVP}} \\ \cdot \overbrace{(\sum_t a_t^{z_m} \otimes \nabla_{s_t^{z_m}} L(z_m;\boldsymbol{\theta}))}^{\text{modular gradient of } z_m}.\end{aligned} \tag{9}$$

Modular Gradient of z_m and z_q. Different from prior work (Grosse et al. 2023) using the independence assumption of all tokens when calculating the pre-activation gradient, we calculate the modular gradient of sublayer f on the sequence $z = (x, y)$:

$$\begin{aligned}\nabla_{\hat{\boldsymbol{\theta}}} L(z_m;\boldsymbol{\theta}) &= -\sum_t \nabla_{\hat{\boldsymbol{\theta}}} \log p(y_t|y_{1:t-1}, x;\boldsymbol{\theta}) \\ &\triangleq \sum_t a_t^{z_m} \otimes \nabla_{s_t^z} L(z_m;\boldsymbol{\theta}),\end{aligned} \tag{10}$$

where $(\cdot)_t^z$ denotes the t-th token of z. The modular gradient of query z_q by the paseudo gradient $\mathcal{D}s_t^{z_q} \triangleq \nabla_{s_t^{z_q}} \log p(z_r|z_p;\boldsymbol{\theta})^{\top}$ and input state $a_t^{z_q}$.

Modular GNH and IHVP. To account for token interdependencies, the modular GNH $\hat{\mathbf{G}}$ can be approximated by Kronecker product $\hat{\mathbf{G}} = \mathbf{F} \triangleq \mathbb{E}[\mathcal{D}_{\hat{\boldsymbol{\theta}}}\mathcal{D}_{\hat{\boldsymbol{\theta}}}^{\top}] \approx \hat{\mathbf{A}} \otimes \hat{\mathbf{S}}$,

where $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ are:

$$\begin{aligned}\hat{\mathbf{A}} &= \frac{1}{NT^2}\sum_{n=1}^{N}\left(\sum_{t=1}^{T} a_t^n\right)\left(\sum_{t=1}^{T} a_t^{n\top}\right), \\ \hat{\mathbf{S}} &= \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{t=1}^{T} \mathcal{D}s_t^n\right)\left(\sum_{t=1}^{T} \mathcal{D}s_t^{n\top}\right).\end{aligned} \tag{11}$$

The approximation for the generalized Gauss-Newton matrix has been derived based on the assumption that activations and pre-activation are independent (Eschenhagen et al. 2023), we present the deriving process for the Fisher matrix in Appendix. After decomposing modular GNH, we can directly compute modular IHVP:

$$\hat{\mathbf{G}}^{-1}(\sum_t a_t^{z_q} \otimes \mathcal{D}s_t^{z_q}) = (\hat{\mathbf{A}} \otimes \hat{\mathbf{S}})^{-1}(\sum_t a_t^{z_q} \otimes \mathcal{D}s_t^{z_q}) = \text{vec}\left(\hat{\mathbf{A}}^{-1}(\sum_t \mathcal{D}s_t^{z_q} a_t^{z_q\top})\hat{\mathbf{S}}^{-1}\right). \tag{12}$$

Workflow of LinFAC. LinFAC's workflow involves the following steps: (1) *Estimate GNH*: Randomly sample n prompts from the training set to generate responses $\{x_i, y_i\}_{i=1}^n$, and compute factors $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ via Eq. 11. (2) *Estimate IHVP*: Calculate the pseudo-gradient for each influence query z_q and determine z_q's IHVP via Eq. 12. (3) *TF-IDF Recall*: To reduce computational load, filter the training data $\mathbb{D}_t$ recalling relevant samples using TF-IDF, following the prior work (Grosse et al. 2023). (4) *Compute Modular Gradient*: Use Eq. 10 to compute the modular gradient for each recalled sample. (5) *Calculate Influence Scores*: Compute influence scores using Eq. 9. Next, we can use these IF-scored training samples to correct the undesirable behavior of LLMs.

3.2 Influence-driven Bregman Optimization

According to the definition of influence score, a positive/negative influence score indicates that the training sample increases/decreases the likelihood of generating undesirable behavior. Based on this, correcting the undesirable behavior can be considered as:

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{z\sim\mathbb{D}_{\mathbb{IF}}}\big[\underbrace{-\mathcal{I}_f(z, z_q, \boldsymbol{\theta})}_{\text{reward term}} - \beta \underbrace{\log \frac{\pi_{\boldsymbol{\theta}}(y|x)}{\pi_{\boldsymbol{\theta}^s}(y|x)}}_{\text{KL divergence term}}\big]. \tag{13}$$

This objective is equivalent to the general RLHF objective (Ouyang et al. 2022) when considering $-\mathcal{I}_f(z_m, z_q, \boldsymbol{\theta})$ as the reward function. Following Rafailov et al. (2023), the optimal solution of Eq. 13 has a close form and be learned by pairwise human preferences. Our method considers the influence ranking as the preference and constructs pairwise data to learn the optimal solution for objective Eq. 13. Hence, we introduce the influence-ranking pairwise loss as:

$$\mathcal{L}_{COR}(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta}^s}) = -\mathbb{E}_{\substack{z^+\sim\mathbb{D}_{\mathbb{IF}}{}^+ \\ z^-\sim\mathbb{D}_{\mathbb{IF}}{}^-}}[\epsilon \cdot \log \sigma\left(\mathcal{I}_f(z^+)\log\frac{\pi_{\boldsymbol{\theta}}(y^+|x^+)}{\pi_{\boldsymbol{\theta}^s}(y^+|x^+)} + \mathcal{I}_f(z^-)\log\frac{\pi_{\boldsymbol{\theta}}(y^-|x^-)}{\pi_{\boldsymbol{\theta}^s}(y^-|x^-)}\right)], \tag{14}$$

where $\mathbb{D}_{\mathbb{IF}} = \mathbb{D}_{\mathbb{IF}}{}^+ \cup \mathbb{D}_{\mathbb{IF}}{}^-$ denotes influential samples. $\mathbb{D}_{\mathbb{IF}}{}^+$and $\mathbb{D}_{\mathbb{IF}}{}^-$ respectively denote the positive and negative influential samples. Due to the large scale of $\mathcal{I}_f(z)$, we need to rescale the influence values by deviding $\max_{z\in\mathbb{D}_t}\{|\mathcal{I}_f(z)|\}$. We set $\epsilon = -1$ in our experiment to correct the undesirable behavior. Previous research observes that influence distribution has the heavy tail (Feldman and Zhang 2020) property and follows the power law (Grosse et al. 2023). Hence, we follow Brown (2020) and employ the Pareto rule to select the significant influential samples $\mathbb{D}_{\mathbb{IF}}{}^+ = \{z|1 - \mathcal{I}_f(z) < \alpha \text{ and } I_f(z) > 0\}$and $\mathbb{D}_{\mathbb{IF}}{}^- = \{z|1 - |\mathcal{I}_f(z)| < \alpha \text{ and } I_f(z) < 0\}$ where α follows the Pareto distribution. An alternative, simpler way is to use top-K sampling directly.

We expect that LLMs can self-correct undesirable behaviors while preserving their original utility. The influence score near zeros means the IHVP is orthogonal to $\nabla_{\boldsymbol{\theta}} L(z; \boldsymbol{\theta})$, namely not influential. Therefore, we use Bregman divergence on the not-influential samples to mitigate catastrophic forgetting during correction. The Bergman divergence with cross-entropy loss function is (derived in Appendix):

$$\mathcal{L}_{BD}(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta}^s}) = \mathbb{E}_{(x,y)\sim\mathbb{D}\setminus\mathbb{D}_{\mathbb{IF}}}\left[\frac{\pi_{\boldsymbol{\theta}}(y|x)}{\pi_{\boldsymbol{\theta}^s}(y|x)} - \log\frac{\pi_{\boldsymbol{\theta}}(y|x)}{\pi_{\boldsymbol{\theta}^s}(y|x)}\right], \tag{15}$$

where the constant term is omitted. Bregman divergence measures the functional discrepancy between the current policy $\pi_{\boldsymbol{\theta}}$ and the original policy $\pi_{\boldsymbol{\theta}^s}$, thereby preventing the corrected model from drastically altering the predictions on the training dataset.

Following Bae et al. (2022), we also employ the proximity regularizer term and the final objective of IBO is:

$$\mathcal{L}_{IBO} = \underbrace{\mathcal{L}_{BD}(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta}^s})}_{\text{divergence term}} + \underbrace{\mathcal{L}_{COR}(\pi_{\boldsymbol{\theta}}, \pi_{\boldsymbol{\theta}^s})}_{\text{correction term}} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^s\|^2}_{\text{proximity regularizer}}. \tag{16}$$

Workflow of IBO. The workflow of IBO involves the following steps: (1) *Influence Ranking*: Rank recalled samples via their influence score. (2) *Sample pairing*: Divide the dataset into $\mathbb{D}_{\mathbb{IF}}{}^+$ (positively influential samples), $\mathbb{D}_{\mathbb{IF}}{}^-$ (negatively influential samples), and noninfluential samples $\mathbb{D} \setminus \mathbb{D}_{\mathbb{IF}}$ by using the Pareto criterion. (3) *Behavior shaping*: Train the LLM using objective defined by Eq. 16 to correct model misbehavior and preserve model utility.

4 Experiments

In this section, we present expansive experiments to evaluate the effectiveness of LANCET. We begin by detailing our experimental setup in Section 4.1 and then demonstrate two key findings in Section 4.2: (1) LANCET can significantly correct seen undesirable behaviors and mitigate the potential of unseen undesirable behaviors without costly human intervention, and (2) LANCET outperforms all advanced baseline methods in correcting misbehavior model outputs while preserving its diversity, utility and quality. We further conduct an in-depth analysis of LANCET in Section 4.3. Our method can efficiently and accurately identify inappropriate samples within the training data and leverage these IF-scored samples to effectively shape model outputs.

4.1 Experimental Setup

Dataset. To evaluate LLM's behavior correction methods, the training dataset needs to include inappropriate samples (i.e., outdated or erroneous samples) and corresponding

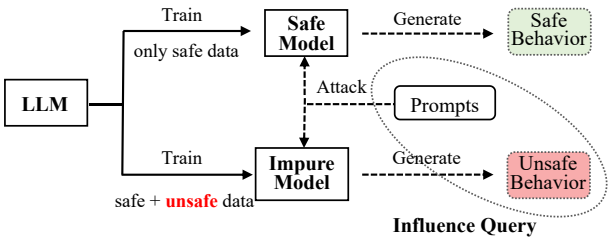


Figure 2: The inappropriate samples (unsafe data) and corresponding caused undesirable behaviors (influence query).

Data Source	Safe Data	Unsafe Data	Influence Query	Unseen Data
BeaverTails	11604	1400	140	1000
Anthropic-HH	43724	1000	100	2000

Table 1: The statistics of the dataset.

caused undesirable behaviors (influence query). However, no existing data is available to simulate this scenario. Alternatively, our experiment uses *unsafe* or *harmful* samples as inappropriate data in training data. We employ the control variates method to collect undesirable outputs. We first inject unsafe samples into the safe data, which may cause some originally safe output to become unsafe. We then select the data with the largest increase in harmfulness scores as *influence queries*. We consider two popular datasets: BeaverTails (Ji et al. 2024) and Anthropic-HH (Bai et al. 2022a). The safe data is from the safe samples of BeaverTails or the helpful-base part of Anthropic-HH (prompt+chosen). The unsafe data is from unsafe samples of BeaverTails or harmless-base (prompt+rejected) of Anthropic-HH, and we employ K-Means to select more diverse samples. To identify the influence queries, we make the unsafe LLM generate 3 times and use the open-released cost models (Yang et al. 2024; Ji et al. 2024) to select samples that show the greatest average improvements on harmfulness score. We include the unseen data that comprises prompts that may induce harmful outputs to evaluate the methods' generalization capability. Table 1 summarizes the dataset details.

Baselines. We compare LANCET with two categories of baselines: (1) *Alignment Methods with Human Correction*, which rely on human intervention to correct unsafe behavior data and then apply Supervised Fine-Tuning (SFT) or Direct Preference Optimization (DPO) for model alignment. We utilize Experience Replay (ER) to maintain the general knowledge (Ouyang et al. 2022), resulting in two baselines: **SFT+ER** and **DPO+ER**. (2) *Model Unlearning Methods*, which autonomously forget target samples via gradient ascent without human involvement. We consider Naive Gradient Ascent (**GA**), Large Language Model Unlearning (**LLM-UL**) (Yao, Xu, and Liu 2024), and Proximal Bregman Optimization (**PBO**) (Bae et al. 2022). To ensure a fair volume of training data, we follow (Grosse et al. 2023) and use TF-IDF and influence queries to identify the same size of contaminated data for forgetting. Additionally, we introduce a strong baseline Proximal Bregman Unlearning (**PBU**), which employs EK-FAC (George et al. 2018) for contaminated data identification followed by PBO for unlearning. For a comprehensive evaluation, we consider the **Impure Model** (trained on contaminated data) and the **Safe Model** (re-trained after manually removing harmful data) as additional baselines.

Type	Metric	Instructions
Automated Metrics	PPL	The perplexity on the test dataset.
	MSTTR	Mean Segmental Type-Token Ratio.
	Self-BLEU	How similar the generated texts are to each other.
	Entropy	The uncertainty or randomness in a set of tokens.
Model-based Scoring	Helpful score	$(0, 1)$ scoring by reward model.
	Harmful score	$(0, 1)$ scoring by $1-$ cost model.
	C-R	The harmful score submits the helpful score.
Manual Scoring	Human Likerit	Human Likerit score (1-7) for influence queries.

Table 2: The evaluation metrics used in our experiments.

Evaluation Design. To comprehensively evaluate the effectiveness of all correction methods, we assess their responses to seen prompts with observed unsafe outputs and unseen prompts with potential hazards. Due to the small size of seen misbehavior (influence queries), **Human Scoring** is used for precise evaluation. Specifically, authors independently score the overall quality of the corrected LLM's outputs on a 1-7 scale, focusing on factors such as coherence, informativeness, and relevance. More details about the human evaluations can be found in the Appendix. For the model's responses to unseen prompts, we use model scoring and automated metrics for evaluation, following Touvron et al. (2023) and Ramamurthy et al. (2022). **Automated Metrics**: This includes measurements of diversity and fluency (perplexity, PPL). High diversity and low PPL suggest that the corrected LLM generates non-trivial, coherent, and informative responses. Importantly, fluency is only meaningful when diversity is not excessively low. **Model-Based Scoring**: This metric assesses the helpfulness and harmfulness of LLM outputs. We employ open-source reward models (RMs) and cost models (CMs) (Dai et al. 2023; Yang et al. 2024; Ji et al. 2024) to evaluate whether corrections maintain utility while minimizing harm. Table 2 shows the metrics used in our experiment.

4.2 Experiment Results

Compared with Human Correction Methods. Since Anthropic-HH encompasses both chosen and rejected responses to a given prompt, it facilitates the simulation of human correction methods for addressing inappropriate samples. If the model exhibits an undesirable response to a prompt, we use the prompt+chosen for SFT training or (chosen, rejected) pair for DPO training. Table 3 presents the performance of LANCET and two alignment methods.

As shown in Table 3, **LANCET significantly reduces harmfulness while preserving diverse, high-quality, and helpful model outputs, greatly diminishing the reliance on human intervention.** Specifically, LANCET achieves an average reduction of 16.8% in harmfulness, which exceeds the 7.7% and 12.6% improvements observed with SFT+ER and DPO+ER, respectively. Additionally, LANCET incurs only a 2.6% reduction in utility, compared to losses of 6.5% for SFT+ER and 4.3% for DPO+ER. These findings suggest that continuous alignment with human corrections may not

Backbone	Method	IF Query Likerit(↑)	Unseen Data Helpful(↑)	Unseen Data Harmful(↓)	Fluency PPL(↓)	Diversity Distinct-1(↑)	Diversity Distinct-2(↑)	Diversity MSTTR(↑)	Self-BLEU(↓)	Entropy(↑)
OPT2.7B	**Impure Model**	3.50	0.738±0.067	0.708±0.051	1.604±0.002	0.180±0.042	0.591±0.035	0.665±0.029	0.182±0.042	6.394±0.022
	Safe Model	4.10	0.692±0.084	0.472±0.058	1.712±0.003	0.172±0.033	0.581±0.015	0.635±0.044	0.168±0.041	6.410±0.041
	SFT+ER	4.06	0.644±0.081	0.566±0.049	1.713±0.015	0.170±0.041	**0.589**±0.062	0.642±0.037	0.169±0.041	**6.370**±0.029
	DPO+ER	4.44	0.669±0.058	0.521±0.045	**1.699**±0.011	0.168±0.023	0.585±0.019	0.631±0.042	0.175±0.067	6.290±0.048
	LANCET	**4.90**	**0.694**±0.037	**0.476**±0.032	1.702±0.009	**0.174**±0.026	0.588±0.020	**0.650**±0.035	**0.167**±0.039	6.369±0.044
Llama2-7B	**Impure Model**	4.00	0.710±0.064	0.597±0.097	1.395±0.008	0.278±0.065	0.665±0.042	0.744±0.031	0.061±0.025	4.452±0.013
	Safe Model	4.50	0.695±0.094	0.485±0.087	1.377±0.002	0.270±0.032	0.617±0.025	0.730±0.019	0.061±0.028	4.530±0.043
	SFT+ER	4.40	0.683±0.035	0.585±0.039	**1.396**±0.007	0.275±0.014	0.629±0.037	0.698±0.026	0.059±0.033	**4.640**±0.051
	DPO+ER	4.72	0.694±0.061	0.532±0.088	1.401±0.017	0.261±0.041	0.619±0.021	**0.701**±0.016	0.077±0.028	4.350±0.036
	LANCET	**5.04**	**0.701**±0.052	**0.493**±0.047	1.397±0.015	**0.280**±0.032	**0.631**±0.041	0.699±0.011	**0.047**±0.054	4.620±0.043

Table 3: The performance of LANCET alongside human correction methods on Anthropic-HH. DPO/SFT+ER denotes model re-alignment through SFT or DPO on human-corrected samples.

Backbone	Method	IF Query Likerit(↑)	Unseen Data Helpful(↑)	Unseen Data Harmful(↓)	Fluency PPL(↓)	Diversity Distinct-1(↑)	Diversity Distinct-2(↑)	Diversity MSTTR(↑)	Self-BLEU(↓)	Entropy(↑)
OPT-2.7B	**Impure Model**	3.40	0.634±0.035	0.326±0.081	1.633±0.004	0.252±0.045	0.686±0.036	0.693±0.035	0.102±0.031	6.530±0.053
	Safe Model	4.28	0.581±0.073	0.121±0.068	1.719±0.006	0.264±0.024	0.762±0.024	0.709±0.031	0.049±0.042	6.800±0.049
	GA	-	0.241±0.027	0.033±0.016	2.922±0.012	-	-	-	-	-
	LLM-UL	3.92	0.515±0.016	0.274±0.029	1.913±0.024	0.189±0.003	0.582±0.007	0.673±0.012	0.209±0.031	6.541±0.046
	PBO	4.10	0.531±0.021	0.201±0.032	1.801±0.019	0.191±0.004	0.589±0.012	**0.694**±0.012	0.182±0.019	6.411±0.059
	PBU	4.04	0.562±0.013	0.158±0.019	1.819±0.000	0.184±0.001	0.577±0.008	0.680±0.007	0.227±0.026	6.600±0.073
	LANCET	**4.84**	**0.597**±0.058	**0.131**±0.012	**1.787**±0.004	**0.223**±0.012	**0.649**±0.027	0.684±0.017	**0.129**±0.007	**6.698**±0.045
Llama2-7B	**Impure Model**	4.10	0.710±0.061	0.391±0.044	1.313±0.006	0.278±0.026	0.629±0.026	0.722±0.016	0.080±0.029	4.181±0.031
	Safe Model	4.90	0.697±0.034	0.172±0.066	1.346±0.002	0.270±0.041	0.617±0.035	0.730±0.022	0.061±0.062	4.530±0.041
	GA	-	0.302±0.061	0.077±0.052	2.719±0.019	-	-	-	-	-
	LLM-UL	3.82	0.593±0.024	0.294±0.021	1.701±0.032	0.215±0.008	0.573±0.019	0.682±0.021	0.113±0.003	4.102±0.004
	PBO	4.46	0.613±0.051	0.221±0.049	1.419±0.019	0.231±0.019	0.564±0.024	0.641±0.053	**0.029**±0.046	3.670±0.042
	PBU	4.20	0.664±0.034	0.194±0.062	**1.401**±0.021	0.200±0.011	0.552±0.011	0.679±0.007	0.139±0.009	4.005±0.015
	LANCET	**5.16**	**0.722**±0.026	**0.165**±0.047	1.409±0.018	**0.276**±0.015	**0.591**±0.017	**0.714**±0.034	0.079±0.014	**4.322**±0.013

Table 4: The performance of LANCET alongside existing model unlearning algorithms on BeaverTails.

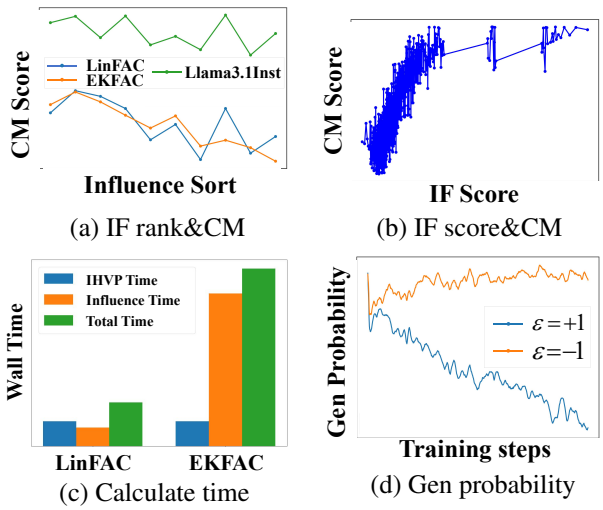
fully counteract the effects of the detrimental data and can contribute to model forgetting. Moreover, a comparison between SFT+ER and DPO+ER reveals that DPO+ER offers superior utility and safety, highlighting that the pairwise learning mechanism in DPO more effectively mitigates harmful data while minimizing model forgetting during continuous alignment. While our method yields marginally superior performance compared to the retrained safe model, the retraining process demands costly human and computational resources. Conversely, our method achieves nearly equivalent results without these extensive demands, enhancing its practicality for real-world applications.

Compared with Model Unlearning Methods. Table 4 presents the performance of LANCET and existing model unlearning methods on the BeaverTails. The results demonstrate that **LANCET effectively corrects model misbehavior and significantly outperforms other model unlearning algorithms regarding model utility and safety.**

Compared to the impure model, LANCET shows a remarkable ability to correct model misbehavior, achieving a 21.1% average reduction in harmfulness while incurring only a minimal 1.1% average loss in model utility. Although a naïve gradient ascent approach can achieve the greatest reduction in harmfulness, it does so at the cost of substantially degrading model utility. A comparison between PBO and PBU reveals that the latter, which leverages influence functions, significantly outperforms the former, which uses TF-IDF in both safety and utility metrics. These findings underscore the efficacy of influence functions in identifying really harmful data within TF-IDF, thereby reducing harmfulness more effectively while maintaining model utility. This result also highlights the necessity of incorporating the improved influence function LinFAC in LANCET. Compared to the safe model, which is retrained with harmful instances removed, we observe that the safe model consistently outperforms all model unlearning methods across model utility, human evaluation, and response diversity. This is expected, as model unlearning methods, which rely on gradient ascent to forget specific instances, often lose model effectiveness, particularly when the recalled samples contain inaccuracies. In contrast, LANCET incorporates a pairwise learning strategy, which helps preserve the quality and diversity of the model's responses as much as possible.

4.3 Futher Analysis

This section comprehensively analyzes mitigates misbehavior in LLMs. Specifically, we uncover three key insights: (1) LANCET employs a novel method LinFAC to efficiently identify influential training samples, which significantly reduces the time and cost associated with human resources; (2) IBO utilizes the influence scores derived from LinFAC for pairwise learning and employ Bregmen Diversity to maintain the model performance, thereby substantially diminishing the adverse effects of harmful samples on model behavior while



(a) IF rank&CM (b) IF score&CM

(c) Calculate time (d) Gen probability

Figure 3: Efficiency and performance analysis of LinFAC and EKFAC.

preserving model utility; and (3) the plug-and-play property of LANCET ensures easy integration into real-world applications, offering better interpretability and compatibility.

The effectiveness of LinFAC. Given influence queries, we use LinFAC and EK-FAC to recall influentially unsafe samples from contaminated data and record the calculation time. Due to directly approximating a whole neural network block instead of considering each linear layer in isolation, LinFAC demonstrates significant performance advantages over EK-FAC. Moreover, we analyze the correction between the influence score and the harmful score. As shown in Figure 3 (a) and (b), the CM score is positively correlated with the influence score and the CM score decreases with the IF score ranking. This indicates that the LinFAC can reflect human preferences on high-scoring samples. The green line in Figure 3 (a) denotes using LinFAC based on Llama3.1-8B Instruct model. Figure 3 (c) shows the computation time of LinFAC and EK-FAC. We observe that the computational time for LinFAC is significantly lower than that for EK-FAC, which is consistent with the theoretical analysis in Section 3.1. As shown in Figure 3 (c), the major time difference comes from computing the IHVP because the matrix dimension of the surrogate model calculated by the LinFAC is much smaller than those of the linear layers computed by EK-FAC. Additionally, LinFAC can approximate multiple MLP layers by merging them, thus saving substantial computational time and resources.

Why IBO can correct LLMs undesirable behavior? Compared to the advanced model unlearning algorithm PBO, IBO employs pairwise learning using samples with both positive and negative influence scores. Figure 3 (d) illustrates the probability of undesirable behavior (influence queries) during the training process. As illustrated in Figure 3 (d), positive intervention ($\epsilon = +1$) boosts the $\bar{P}(z_r|z_p)$ and negative intervention ($\epsilon = -1$) reduces the $\bar{P}(z_r|z_p)$. It demonstrates that the influence scores can reflect the impact of samples on LLM's behavior. Therefore, IBO can correct LLM's undesirable behavior by enhancing the probability of generating safe responses and reducing the probability of unsafe responses. Moreover, IBO avoids using gradient ascent to forget unsafe samples, ensuring the stability of model training and maintaining the utility of the model.

Backbone	Recall Strategy	Post Training	Model Scoring Helpful(↑)	Model Scoring Harmless(↓)	Fluency PPL(↓)
OPT-2.7B	Impure Model		0.634±0.035	0.326±0.081	1.633±0.004
	EK-FAC	PBO	0.562±0.013	**0.158**±0.019	1.819±0.000
		IBO	**0.603**±0.008	0.161±0.021	**1.761**±0.004
	LinFAC	PBO	0.548±0.022	0.134±0.030	1.820±0.007
		IBO	**0.597**±0.058	**0.131**±0.012	**1.787**±0.004
Llama2-7B	Impure Model		0.710±0.061	0.391±0.044	1.313±0.006
	EK-FAC	PBO	0.664±0.034	0.194±0.062	1.401±0.021
		IBO	**0.688**±0.045	**0.170**±0.049	**1.375**±0.032
	LinFAC	PBO	0.597±0.081	0.184±0.071	1.467±0.039
		IBO	**0.722**±0.026	**0.165**±0.047	**1.409**±0.018

Table 5: Compatibility analysis of LANCET on BeaverTails.

Compatibility analysis. Our method comprises a two-stage pipeline designed to rectify the model's undesirable behaviors, with each stage being modular and combinable. This modularity ensures that our approach is both plug-and-play and highly compatible. Table 5 presents a compatibility analysis of our method, which explores various combinations of recall strategies (LinFAC and EK-FAC) and post-training techniques (IBO and PBO). As listed in Table 5, all combination methods reduce the harmfulness of the model's output while PBO damages the utility more than IBO. Among these combinations, IBO outperforms PBO under the same recall strategy. Since IBO uses positive and negative influential samples to correct model behavior, PBO only performs unlearning on unsafe samples. LinFAC has an advantage on the harmless score to EK-FAC and performs comparably to EK-FAC on the helpful score under the same post-training. For example, under OPT-2.7B, the average harmless score for EK-FAC is 16.0%, whereas it is 13.3% for LinFAC. It indicates that the samples recalled by LinFAC are more effective in correcting model behavior than those recalled by EK-FAC.

5 Conclusion

The work presents a novel method LANCET for correcting LLM behavior without requiring extensive human resources. LANCET leverages influence functions to identify influential samples within the training set and efficiently correct inappropriate behavior. This is achieved through the innovative LinFAC technique, which significantly reduces the computational complexity compared to existing methods, and the Influence-driven Bregman Optimization (IBO), which modifies the model via learning the IF-ranking information of LinFAC. The effectiveness of LANCET is validated through comprehensive experiments, demonstrating the ability to effectively correct model behavior, in comparison to model unlearning and human correction methods. Our research offers a promising solution for the ongoing development and application of LLMs in an ever-changing societal context, ensuring that AI systems remain responsive and adaptive to evolving human values and preferences.

Acknowledgments

This research was supported in part by the Major Key Proiect of PCL (NO.PCL2023A09), the National Key Research and Development Program of China (2021ZD0112905), the National Natural Science Foundation of China (62176076), the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005), Natural Science Foundation of Guangdong (2023A1515012922), Shenzhen Foundational Research Funding (JCYJ20220818102415032), UKVI New Horizons grant (EP/X019063/1) and EPSRC IAA at KCL.

References

Bae, J.; Ng, N.; Lo, A.; Ghassemi, M.; and Grosse, R. B. 2022. If influence functions are the answer, then what is the question? *Advances in Neural Information Processing Systems*, 35: 17953–17967.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; Joseph, N.; Kadavath, S.; Kernion, J.; Conerly, T.; El-Showk, S.; Elhage, N.; Hatfield-Dodds, Z.; Hernandez, D.; Hume, T.; Johnston, S.; Kravec, S.; Lovitt, L.; Nanda, N.; Olsson, C.; Amodei, D.; Brown, T.; Clark, J.; McCandlish, S.; Olah, C.; Mann, B.; and Kaplan, J. 2022a. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv:2204.05862.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; Chen, C.; Olsson, C.; Olah, C.; Hernandez, D.; Drain, D.; Ganguli, D.; Li, D.; Tran-Johnson, E.; Perez, E.; Kerr, J.; Mueller, J.; Ladish, J.; Landau, J.; Ndousse, K.; Lukosuite, K.; Lovitt, L.; Sellitto, M.; Elhage, N.; Schiefer, N.; Mercado, N.; DasSarma, N.; Lasenby, R.; Larson, R.; Ringer, S.; Johnston, S.; Kravec, S.; Showk, S. E.; Fort, S.; Lanham, T.; Telleen-Lawton, T.; Conerly, T.; Henighan, T.; Hume, T.; Bowman, S. R.; Hatfield-Dodds, Z.; Mann, B.; Amodei, D.; Joseph, N.; McCandlish, S.; Brown, T.; and Kaplan, J. 2022b. Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073.

Brown, T. B. 2020. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*.

Carroll, M.; Foote, D.; Siththaranjan, A.; Russell, S.; and Dragan, A. 2024. AI Alignment with Changing and Influenceable Reward Functions. In *ICLR 2024 Workshop: How Far Are We From AGI*.

Dai, J.; Pan, X.; Sun, R.; Ji, J.; Xu, X.; Liu, M.; Wang, Y.; and Yang, Y. 2023. Safe RLHF: Safe Reinforcement Learning from Human Feedback. arXiv:2310.12773.

Eschenhagen, R.; Immer, A.; Turner, R. E.; Schneider, F.; and Hennig, P. 2023. Kronecker-Factored Approximate Curvature for Modern Neural Network Architectures. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Feldman, V.; and Zhang, C. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33: 2881–2891.

George, T.; Laurent, C.; Bouthillier, X.; Ballas, N.; and Vincent, P. 2018. Fast approximate natural gradient descent in a kronecker factored eigenbasis. *Advances in Neural Information Processing Systems*, 31.

Grosse, R.; Bae, J.; Anil, C.; Elhage, N.; Tamkin, A.; Tajdini, A.; Steiner, B.; Li, D.; Durmus, E.; Perez, E.; et al. 2023. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*.

Hampel, F. R. 1974. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346): 383–393.

Jacot, A.; Hongler, C.; and Gabriel, F. 2018. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 8580–8589.

Jaques, N.; Shen, J. H.; Ghandeharioun, A.; Ferguson, C.; Lapedriza, A.; Jones, N.; Gu, S.; and Picard, R. 2020. Human-centric dialog training via offline reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3985–4003. Online: Association for Computational Linguistics.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Chen, B.; Sun, R.; Wang, Y.; and Yang, Y. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Kreutzer, J.; Khadivi, S.; Matusov, E.; and Riezler, S. 2018. Can Neural Machine Translation be Improved with User Feedback? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, 92–105. New Orleans - Louisiana: Association for Computational Linguistics.

Ortiz-Jiménez, G.; Moosavi-Dezfooli, S.-M.; and Frossard, P. 2021. What can linearized neural networks actually say about generalization? *Advances in Neural Information Processing Systems*, 34: 8998–9010.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ramamurthy, R.; Ammanabrolu, P.; Brantley, K.; Hessel, J.; Sifa, R.; Bauckhage, C.; Hajishirzi, H.; and Choi, Y. 2022. Is Reinforcement Learning (Not) for Natural Language Processing?: Benchmarks, Baselines, and Building Blocks for Natural Language Policy Optimization.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D. M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize from human feedback. *CoRR*, abs/2009.01325.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yang, R.; Pan, X.; Luo, F.; Qiu, S.; Zhong, H.; Yu, D.; and Chen, J. 2024. Rewards-in-Context: Multi-objective Alignment of Foundation Models with Dynamic Preference Adjustment. *International Conference on Machine Learning*.

Yao, Y.; Xu, X.; and Liu, Y. 2024. Large Language Model Unlearning. arXiv:2310.10683.

Zhang, H.; Lei, Y.; Gui, L.; Yang, M.; He, Y.; Wang, H.; and Xu, R. 2024. CPPO: Continual Learning for Reinforcement Learning with Human Feedback. In *The Twelfth International Conference on Learning Representations*.