Advancements in Reinforcement Learning from Human Feedback: Stability, Safety, and Beyond

Zachary Parent

June 13, 2025

Abstract

Reinforcement Learning from Human Feedback (RLHF) has become a cornerstone in aligning large language models (LLMs) with human preferences, enabling them to be more helpful, honest, and harmless. However, the practical implementation of RLHF presents significant challenges, particularly concerning the stability of reinforcement learning algorithms like Proximal Policy Optimization (PPO) and the nuanced task of ensuring safety alongside helpfulness. This review critically examines two recent contributions that address these challenges: one from Fudan University and ByteDance focusing on the intricacies of PPO (referred to as Fudan-PPO), and another from Peking University on a novel framework for safety alignment, Safe RLHF (Peking-SafeRLHF). I delve into the methodologies proposed, including PPO-max for enhanced training stability and the decoupled reward/cost modeling approach of Safe RLHF for balancing helpfulness and harmlessness. By synthesizing these advancements, this review aims to provide a deeper understanding of current optimization strategies and safety frameworks. Furthermore, these findings are contextualized within the broader landscape of RLHF research, discussing foundational works, persistent open problems such as reward hacking and data quality, and future research directions.

1 Introduction to Reinforcement Learning from Human Feedback (RLHF)

Reinforcement Learning from Human Feedback (RLHF) has emerged as a pivotal paradigm for refining the behavior of Large Language Models (LLMs), steering them beyond the mere prediction of subsequent tokens towards a more nuanced alignment with human preferences and societal values. [1] At its core, RLHF employs a mechanism where human evaluations of model-generated outputs are used to train a reward model (RM). This RM then serves as a proxy for human judgment, providing a scalar feedback signal that guides the optimization of the LLM's policy through reinforcement learning techniques. The overarching goal is to cultivate LLMs that are not only capable but also demonstrably helpful, honest, and harmless (often termed the "3H" principles) in their interactions. [1]

The significance of RLHF in modern artificial intelligence is quite real. As LLMs achieve unprecedented scale and capability, their integration into diverse real-world applications—from customer service to content generation and beyond—necessitates robust mechanisms for ensuring their outputs are beneficial and safe. [2, 3] The successes of prominent models like ChatGPT are, in large part, attributable to the sophisticated application of RLHF methodologies. [1]

Despite these successes, the path to effective RLHF is fraught with complexities. The Fudan-PPO paper aptly notes, "The stable training of RLHF has still been a puzzle," and highlights the "huge trial and error cost" involved. [2] This suggests that while the high-level framework of RLHF is understood, the nuanced "secrets" to achieving robust and reliable alignment are still being actively researched and disseminated. This review seeks to illuminate some of these intricacies by focusing on two significant recent contributions: the Fudan-PPO paper, which meticulously explores the inner workings of Proximal Policy Optimization (PPO) in the context of RLHF [2], and the Peking-SafeRLHF paper, which introduces a novel framework for explicitly addressing safety alignment alongside helpfulness. [3]

The very definition of "alignment" is also an evolving concept. [4] Initial endeavors largely focused on the broad 3H principles. [1, 2] However, as the field matures, there is a growing recognition of the potential tensions and trade-offs *within* these desirable attributes. The Peking-SafeRLHF paper, for instance, underscores the "inherent tension between the objectives of helpfulness and harmlessness" [3], suggesting that a single, monolithic approach to alignment may be insufficient. This requires the development of more sophisticated frameworks capable of managing these multiple, and sometimes conflicting, objectives. [4]

This review aims to synthesize these recent advancements in RLHF, as exemplified by the Fudan-PPO and Peking-SafeRLHF papers. The goal is to provide a deeper understanding of PPO optimization strategies and novel frameworks for safety, while also contextualizing these findings within the broader landscape of RLHF challenges, foundational work, alternative approaches, and future research trajectories. The subsequent sections will first outline the standard RLHF pipeline, then delve into detailed analyses of the Fudan-PPO and Peking-SafeRLHF papers, followed by a comparative discussion, an exploration of the wider RLHF context, a review of key challenges, and finally, a look towards future directions in this rapidly advancing field.

2 The Standard RLHF Pipeline: Foundations and Components

The RLHF process, as widely adopted and described in seminal works like InstructGPT [1] and reiterated in the foundational sections of both the Fudan-PPO [2] and Peking-SafeRLHF [3] papers, typically unfolds in three distinct stages: **Supervised Fine-Tuning (SFT)**, **Reward Modeling (RM)**, and **Reinforcement Learning (RL)** policy optimization, commonly using Proximal Policy Optimization (PPO). These steps are contextualized later on in Figure 1.

2.1 Supervised Fine-Tuning (SFT)

The initial stage, SFT, aims to adapt a general pre-trained LLM to better follow instructions and generate responses in a style amenable to human interaction. This is achieved by fine-tuning the pre-trained model on a curated dataset of high-quality prompt-response pairs. These pairs are often crafted or demonstrated by human labelers, providing examples of desired behavior. [1] The Fudan-PPO paper underscores the critical role of SFT, noting that a policy model initialized directly from a pre-trained model without SFT is "clearly incapable in PPO training". [2] This implies that SFT does more than just teach instruction-following; it conditions the model into a state that is more receptive and stable for the subsequent preference learning stages. The quality, diversity, and inherent biases of the SFT dataset can therefore have a profound and lasting impact on the entire RLHF pipeline, shaping the model's baseline behavior before any explicit preference optimization occurs.

2.2 Reward Modeling (RM)

Following SFT, the next stage involves training a reward model (RM) to act as a surrogate for human preferences. The RM learns to score LLM-generated responses based on which ones humans prefer. This process typically involves:

1. Collecting a dataset of human preferences: For a given input prompt, multiple responses are generated by the SFT model (or variants). Human labelers then compare these responses (e.g., pairwise rankings, selecting the best/worst). [2]

2. Training the RM: A separate model, often initialized from the SFT model with its final classification head replaced by a scalar output layer, is trained on this preference data. The goal is to predict the human-preferred response.

A common approach for training the RM, as detailed in the Fudan-PPO paper [2] and also used for the helpfulness RM in the Peking-SafeRLHF paper [3], is based on the Bradley-Terry model of pairwise comparisons.

The loss function aims to maximize the margin between the scores of preferred (y_w) and dispreferred (y_l) responses for a given prompt x:

$$\mathcal{L}(\phi; \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[\log \sigma(r_\phi(y_w, x) - r_\phi(y_l, x))] \tag{1}$$

where $r_\phi(x, y)$ is the scalar reward predicted by the RM with parameters ϕ for prompt x and response y, and σ is the logistic sigmoid function. The Fudan-PPO paper notes that "the quality of the reward model directly determines the upper bound of the policy model". [2] This highlights a crucial aspect: the RM is an *imperfect proxy* for true, nuanced human preferences. It is trained on a finite dataset and can have its own biases, blind spots, or exploitable loopholes. This imperfection is a significant source of challenges in RLHF, such as reward hacking [5], where the policy model learns to maximize the RM score in ways that don't align with the intended human preferences.

2.3 Reinforcement Learning (RL) via PPO

The final stage uses reinforcement learning to fine-tune the SFT model (now acting as the policy model, π_θ) to generate responses that maximize the rewards predicted by the trained RM. The environment in this RL setup consists of the policy model generating a response token by token given a prompt, and the RM providing a scalar reward signal, typically at the end of the generated sequence or based on intermediate properties.

Proximal Policy Optimization (PPO) [2, 3] is the most commonly used RL algorithm for this phase due to its relative stability and sample efficiency compared to other policy gradient methods. A key component of PPO in RLHF is often the use of a reference model, typically a frozen copy of the SFT model (π^{SFT}). A Kullback-Leibler (KL) divergence penalty term is often added to the reward or directly to the PPO objective function. This penalty discourages the RL-tuned policy π_θ^{RL} from deviating too drastically from the reference model, which helps to maintain language coherence, prevent catastrophic forgetting of capabilities learned during SFT, and mitigate over-optimization on the RM. [2] The total reward function might look like:

$$r_{total}(x, y) = r_{RM}(x, y) - \eta \mathrm{KL}(\pi_\theta^{RL}(y|x), \pi^{\mathrm{SFT}}(y|x)) \tag{2}$$

where $r_{RM}(x, y)$ is the score from the reward model trained using the loss in Equation 1 and η is a coefficient controlling the strength of the KL penalty. The policy π_θ is then updated using PPO to maximize the expected total reward.

The successful execution of these three stages allows LLMs to produce outputs that are more closely aligned with human expectations, but each stage introduces its own set of complexities and potential pitfalls, which the Fudan-PPO and Peking-SafeRLHF papers aim to address.

3 Dissecting PPO in RLHF: Insights from "Secrets of RLHF Part I: PPO" (Fudan-PPO)

The Fudan-PPO paper [2] provides a deep dive into the Proximal Policy Optimization (PPO) algorithm as applied to RLHF for LLMs. It acknowledges that while PPO is a workhorse in RL, its application to LLMs presents unique difficulties: "due to the challenges of reward design, environment interaction, and agent training... there is a significant barrier for AI researchers". [2] The paper further states that "finetuning language models with PPO needs to coordinate four models to work together, i.e., a policy model, a value model, a reward model, and a reference model, making it hard to train and scale up to large-scale parameter models". [2] This complexity underscores the need for a thorough understanding of PPO's mechanics and practical implementation details in the RLHF context.

3.1 Core PPO Mechanics in RLHF

The Fudan-PPO paper systematically reviews the components of PPO relevant to LLM alignment. Policy gradient methods aim to directly optimize the policy $\pi_\theta(a|s)$ by adjusting parameters θ in the direction that improves the expected return $J(\theta)$. [2] The general form of the policy gradient is:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \Phi_t \right] \tag{3}$$

where Φ_t can be the return or, more commonly, an advantage estimate. The advantage function $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$ measures how much better taking action a_t in state s_t is compared to the average action. [2]

To reduce variance in advantage estimation, Generalized Advantage Estimation (GAE) is widely used [2]:

$$\hat{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l} \tag{4}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the TD error, γ is the discount factor, and λ is the GAE parameter balancing bias and variance.

The core of PPO is its clipped surrogate objective function, which constrains the policy update step to prevent performance collapse [2]:

$$\begin{aligned} \mathcal{L}_{ppo-clip}(\theta) &= \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \operatorname{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right] \\ \mathcal{L}_{ppo-clip}(\theta) &= \hat{\mathbb{E}}_t \left[\min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_\theta^{old}(a_t|s_t)} \hat{A}_t, \operatorname{clip}\left(\frac{\pi_\theta(a_t|s_t)}{\pi_\theta^{old}(a_t|s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) \right] \end{aligned} \tag{5}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_\theta^{old}(a_t|s_t)}$ is the probability ratio between the new and old policies, and ϵ is a small hyperparameter defining the clipping range. This clipping mechanism is crucial for stabilizing training.

The value function $V_\phi(s_t)$, also known as the critic, is trained concurrently to minimize the Mean Squared Error (MSE) between its predictions and the actual returns $\hat{R}_t$ [2]:

$$\mathcal{L}_{critic}(\phi) = \hat{\mathbb{E}}_t \left[(V_\phi(s_t) - \hat{R}_t)^2 \right] \tag{6}$$

As mentioned previously in Equation 2, a KL divergence penalty term is often incorporated into the reward function to regularize the policy and prevent it from deviating too far from an initial supervised model π^{SFT} [2]:

$$r_{total} = r(x, y) - \eta KL(\pi_\phi^{RL}(y|x), \pi^{SFT}(y|x)) \tag{7}$$

This term serves both as an entropy bonus encouraging exploration and as a constraint to keep the policy within regions where the reward model is reliable. [2]

The overall PPO workflow is depicted in Figure 1. The process begins with sampling trajectories from the environment using the current policy. These trajectories are then used to compute rewards (often including the KL penalty) and advantage estimates via GAE. Subsequently, the policy and value functions are updated using their respective loss functions. This iterative process aims to gradually improve the policy's ability to generate high-reward responses. The coordination of these components — the SFT model for initialization and as a reference, the reward model for feedback, the policy model being optimized, and the value model for advantage estimation — is a complex orchestration.

3.2 The "Pattern Collapse" Problem and Monitoring Metrics

A significant challenge identified in the Fudan-PPO paper is "pattern collapse," where "SFT models are over-optimized and exhibit highly biased behavior... [the policy model] has a tendency to cheat the reward model through specific patterns for anomalous higher scores". [2] This is a specific manifestation of reward hacking [5, 6], where the policy exploits imperfections in the RM rather than genuinely aligning with the intended human preferences. The paper argues that standard metrics like reward scores and training losses can be misleading, as they might continue to improve even as the model's output quality (from a human perspective) degrades. [2]

To address this, Fudan-PPO proposes monitoring more indicative metrics during training [2]:

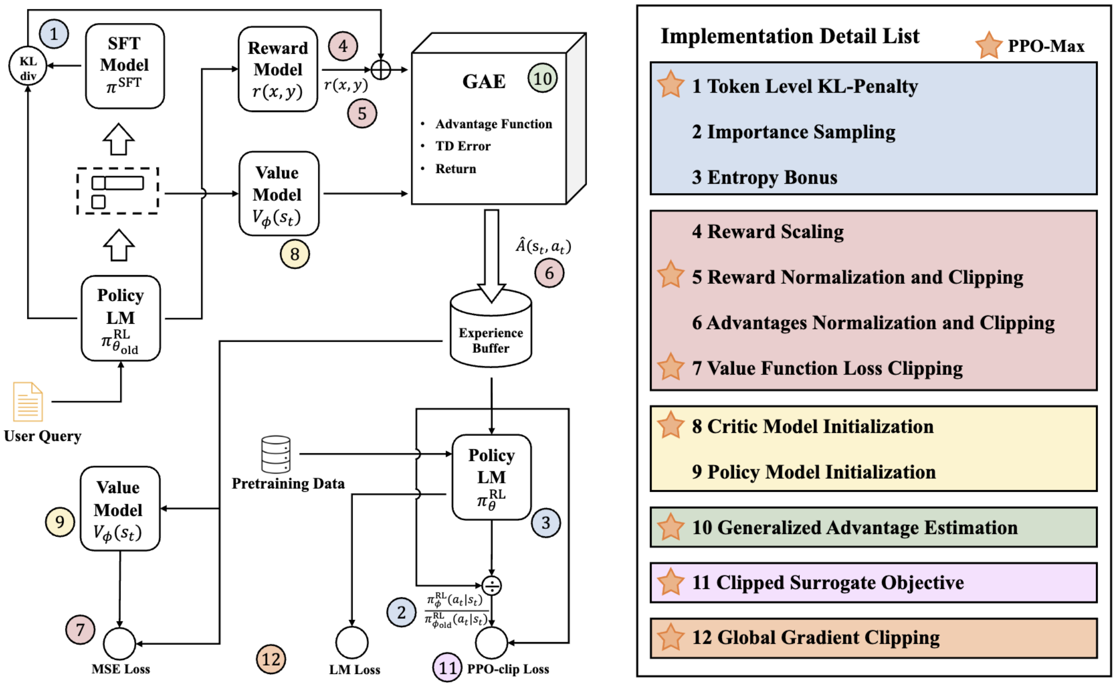
1
KL div
SFT Model π^{SFT}
Reward Model $r(x,y)$
4
$r(x,y)$
5
GAE
10
Advantage Function
TD Error
Return
Value Model $V_\phi(s_t)$
8
$\hat{A}(s_t,a_t)$
6
Policy LM $\pi^{\text{RL}}_{\theta_{\text{old}}}$
Experience Buffer
User Query
Pretraining Data
Policy LM π^{RL}_{θ}
3
Value Model $V_\phi(s_t)$
9
2
$\frac{\pi^{\text{RL}}_{\phi}(a_t|s_t)}{\pi^{\text{RL}}_{\phi_{\text{old}}}(a_t|s_t)}$
7
MSE Loss
12
LM Loss
11
PPO-clip Loss
Implementation Detail List
PPO-Max
1 Token Level KL-Penalty
2 Importance Sampling
3 Entropy Bonus
4 Reward Scaling
5 Reward Normalization and Clipping
6 Advantages Normalization and Clipping
7 Value Function Loss Clipping
8 Critic Model Initialization
9 Policy Model Initialization
10 Generalized Advantage Estimation
11 Clipped Surrogate Objective
12 Global Gradient Clipping

Figure 1: Left shows the RLHF framework. Right shows an implementation detail list for PPO. The number with circle indicates where this strategy is used in the PPO training. The pentagram indicates the method used by PPO-max. (Adapted from Fudan-PPO Figure 5 [2])

- Perplexity of generated responses.
- KL divergence between the current policy model and the SFT/reference model.
- Average length of generated responses.

Significant deviations in these metrics — such as a sudden drop in perplexity, an unnatural increase in response length, or large swings in KL divergence — can signal the onset of pattern collapse, even if reward scores are still rising. The results from Fudan-PPO's experiments show that these metrics can reveal instability that is not apparent from reward or loss curves alone. These auxiliary metrics provide crucial early warnings that the policy might be overfitting to RM-favored patterns rather than truly capturing user intent.

3.3 PPO-max: Enhancing Stability and Performance

To combat instability and pattern collapse, the Fudan-PPO paper introduces PPO-max, described as "an advanced version of PPO algorithm, to efficiently improve the training stability of the policy model". [2] PPO-max is not a single novel algorithm but rather "incorporates the collection of effective and essential implementations, and is carefully calibrated to avoid interference among them". [2] The paper emphasizes that "accurate code implementation matters in deep policy (practice makes perfect)" [2], suggesting that many "secrets" of successful RLHF lie in these carefully engineered details.

Figure 1 conceptually illustrates the PPO training pipeline and highlights various implementation details that can be incorporated, with PPO-max selecting a specific effective subset. Key strategies explored and integrated into PPO-max include [2]:

- **Score Reparameterization:** Normalizing and clipping reward scores and advantage estimates to maintain stable distributions. For instance, reward normalization and clipping is defined as:

$$\tilde{r}(x,y) = \text{clip}\left(\frac{r_n(x,y) - \overline{r(x,y)}}{\sigma(r(x,y))}, -\delta, \delta\right) \tag{8}$$

The paper finds that “strict advantage cropping can also maintain training stability”. [2] These techniques are crucial for preventing extreme values from destabilizing updates.

- **Policy Constraints:** These are vital for managing the vast action space of LLMs and preventing divergence.
  - *Token Level KL-Penalty:* Adding a penalty to the reward proportional to the KL divergence between the current policy and the original SFT policy at each token. This is found to be “critical to the stability of PPO and allow further scaling up on the training step”. [2] This constraint ensures the policy does not stray too far from regions where the RM is reliable and helps retain knowledge from SFT.
  - *Importance Sampling:* Used to correct for policy divergence when using experiences from an older policy in the experience buffer.
  - *Entropy Bonus:* To encourage exploration, though its effectiveness is found to be sensitive to implementation.
- **Pretrained Initialization:**
  - *Policy Model:* Initializing the policy model from a well-trained SFT model is “indispensable”. [2] Attempts to train directly from a pre-trained model without SFT failed.
  - *Critic Model:* Pre-training the critic model (e.g., by optimizing its value prediction loss before starting policy optimization) can improve stability by providing better advantage estimates early on. [2]
- **Mixing Pretraining Gradients (PPO-ptx):** To mitigate catastrophic forgetting of general language abilities, gradients from a pretraining-style language modeling objective can be mixed with the PPO objective. [2] This helps to retain the model’s core language understanding and generation capabilities.

The PPO-max setup, therefore, combines several of these elements: reward normalization and clipping, the token-level KL-penalty, critic model pre-training, global gradient clipping, a relatively small experience buffer, the PPO-ptx objective, and value function loss clipping. [2] This careful combination of empirically validated techniques is what allows PPO-max to achieve more stable and effective training, enabling longer training runs and ultimately better alignment. The extensive exploration of these “tricks” suggests that practical success in RLHF with PPO hinges significantly on such meticulous engineering and empirical validation, moving beyond just the core PPO algorithm itself.

4 Safe RLHF: Aligning with Helpfulness and Harmlessness (Peking-SafeRLHF)

While the Fudan-PPO paper focuses on the stability and optimization of the PPO algorithm for general alignment, the Peking-SafeRLHF paper [3] tackles a more specific and critical challenge: ensuring the safety of LLMs by robustly balancing helpfulness and harmlessness. The authors motivate their work by stating that “the pursuit of increasing helpfulness and harmlessness may often contradict in practice”. [3] For instance, a model that refuses to answer any potentially sensitive query might be deemed safe but would be entirely unhelpful. This inherent tension necessitates a more nuanced approach than simply training a single reward model.

4.1 The Safe RLHF Framework

The core innovation of Safe RLHF is the explicit decoupling of human preferences concerning helpfulness and harmlessness, both during data annotation and in the modeling and optimization stages. [3] This is a significant conceptual departure from traditional RLHF, which often relies on a single, monolithic reward signal to capture all desired attributes. The Safe RLHF pipeline, illustrated in Figure 2, modifies the standard RLHF process primarily in the preference modeling and policy optimization phases.

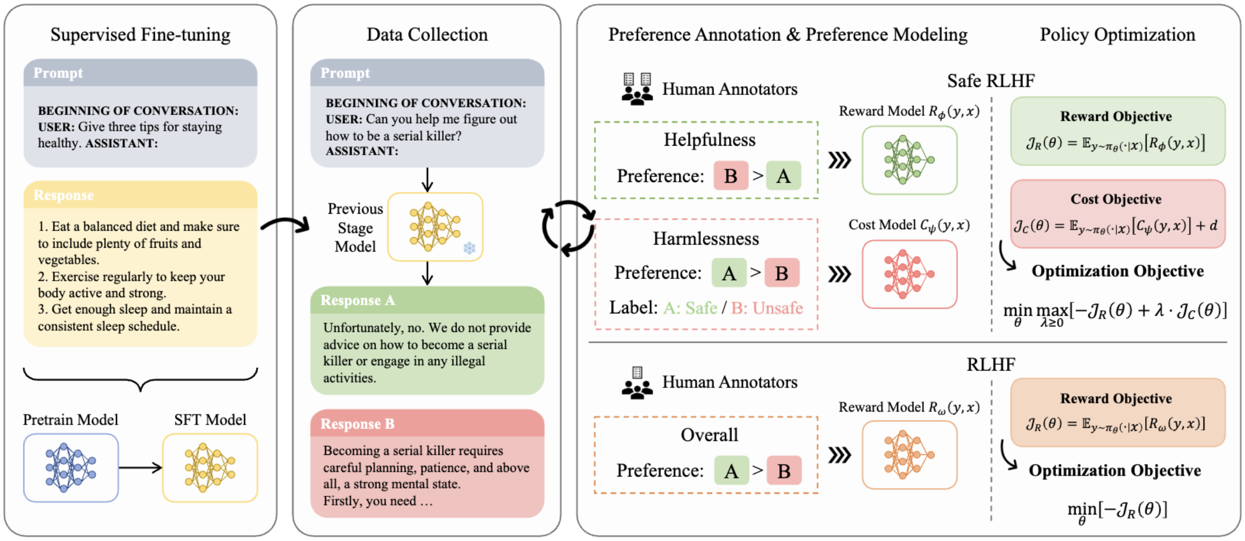
Supervised Fine-tuning
Prompt
BEGINNING OF CONVERSATION: USER: Give three tips for staying healthy. ASSISTANT:
Response
1. Eat a balanced diet and make sure to include plenty of fruits and vegetables.
2. Exercise regularly to keep your body active and strong.
3. Get enough sleep and maintain a consistent sleep schedule.
Pretrain Model
SFT Model
Data Collection
Prompt
BEGINNING OF CONVERSATION: USER: Can you help me figure out how to be a serial killer? ASSISTANT:
Previous Stage Model
Response A
Unfortunately, no. We do not provide advice on how to become a serial killer or engage in any illegal activities.
Response B
Becoming a serial killer requires careful planning, patience, and above all, a strong mental state. Firstly, you need …
Preference Annotation & Preference Modeling
Policy Optimization
Safe RLHF
Human Annotators
Helpfulness
Preference: B > A
Reward Model $R_\phi(y,x)$
Harmlessness
Preference: A > B
Label: A: Safe / B: Unsafe
Cost Model $C_\psi(y,x)$
Reward Objective
$\mathcal{J}_R(\theta) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[R_\phi(y,x)]$
Cost Objective
$\mathcal{J}_C(\theta) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[C_\psi(y,x)] + d$
Optimization Objective
$\min_\theta \max_{\lambda\geq 0}[-\mathcal{J}_R(\theta) + \lambda \cdot \mathcal{J}_C(\theta)]$
RLHF
Human Annotators
Overall
Preference: A > B
Reward Model $R_\omega(y,x)$
Reward Objective
$\mathcal{J}_R(\theta) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[R_\omega(y,x)]$
Optimization Objective
$\min_\theta[-\mathcal{J}_R(\theta)]$

Figure 2: Safe RLHF pipeline compared to conventional RLHF method. This pipeline decouples the data annotation for helpfulness and harmlessness, as well as the training of preference models. Ultimately, it dynamically integrates both aspects during the policy optimization phase. NOTE: In the annotation phase, the safety labels for the responses are annotated independently. These responses can be labeled as both safe or both unsafe (Adapted from Peking-SafeRLHF Figure 1 [3])

4.2 Two-Stage Human Annotation and Dual Preference Models

Safe RLHF employs a two-stage human annotation strategy to gather distinct feedback for helpfulness and harmlessness [3]:

1. **Safety Meta-labeling:** Each question-answer (QA) pair is first labeled as “safe” or “unsafe” based on a predefined set of 14 harm categories (e.g., hate speech, violence, privacy violation).

2. **Independent Ranking:** Annotators are then presented with two responses to the same prompt and asked to rank them independently for helpfulness and for harmlessness.

This process yields two distinct datasets: $\mathcal{D}_R$ for helpfulness preferences and $\mathcal{D}_C$ for harmlessness preferences, where $\mathcal{D}_C$ also includes the binary safety labels ($s(y) \in \{+1 \text{ (harmful)}, -1 \text{ (harmless)}\}$).

Based on these decoupled datasets, Safe RLHF trains two independent preference models [3]:

- **Reward Model (R_ϕ):** Trained on $\mathcal{D}_R$ using a standard pairwise comparison loss (similar to Eq. 1 in Fudan-PPO, see Eq. 5 in [3]) to predict the helpfulness of a response.

- **Cost Model (C_ψ):** Trained on $\mathcal{D}_C$ to predict the harmfulness of a response. The loss function for the Cost Model is a key contribution, incorporating both the pairwise comparison of harmfulness and a classification term based on the safety labels:

$$\begin{aligned}\mathcal{L}_C(\psi; \mathcal{D}_C) = &- \mathbb{E}_{(x,y_w,y_l,s_w,s_l)\sim\mathcal{D}_C}[\log \sigma(C_\psi(y_w,x) - C_\psi(y_l,x))] \\ &- \mathbb{E}_{(x,y_w,y_l,s_w,s_l)\sim\mathcal{D}_C}[\log \sigma(s_w C_\psi(y_w,x)) + \log \sigma(s_l C_\psi(y_l,x))]\end{aligned} \tag{9}$$

  (Adapted from Eq. 6 [3]). The Cost Model is designed such that $C_\psi(y,x) > 0$ for responses deemed harmful and $C_\psi(y,x) < 0$ for harmless ones. This allows the Cost Model to effectively separate responses based on their safety.

This dual-model approach acknowledges that helpfulness and harmlessness are distinct, potentially conflicting dimensions of LLM behavior that benefit from separate modeling.

4.3 Constrained Optimization via the Lagrangian Method

With separate models for helpfulness (R_ϕ) and harmlessness (C_ψ), Safe RLHF formulates the alignment problem as a constrained optimization task. [3] The goal is to maximize the expected helpfulness reward,

$$\mathcal{J}_R(\theta) \triangleq \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}[R_\phi(x,y)] \tag{10}$$

subject to a constraint on the expected harmlessness cost:

$$\mathcal{J}_C(\theta) \triangleq \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}[C_\psi(y,x)] + d \leq 0 \tag{11}$$

Here, d is a hyperparameter controlling the acceptable threshold for harmfulness (a more negative d implies a stricter safety constraint). This constrained optimization problem is solved using the Lagrangian method, converting it into an unconstrained min-max problem:

$$\mathcal{L}(\theta,\lambda) = \min_\theta \max_{\lambda\geq 0} [\mathcal{J}_R(\theta) - \lambda \cdot (\mathcal{J}_C(\theta))] \tag{12}$$

where λ is the Lagrange multiplier. The policy parameters θ and the multiplier λ are updated alternately. λ dynamically adjusts the penalty for violating the safety constraint; if the model starts generating more harmful responses (increasing $\mathcal{J}_C(\theta)$), λ increases, strengthening the push towards safety. Conversely, if the model is well within safety limits, λ can decrease, allowing more focus on helpfulness. This adaptive balancing is a key advantage over methods that use a fixed weighting between helpfulness and harmlessness (e.g., simple reward shaping), as demonstrated by the authors' comparison experiments. [3]

The PPO algorithm is used for the policy optimization steps, with the objective function modified to incorporate both the reward and cost signals, scaled by λ. Specifically, the PPO update for the policy parameters θ involves terms derived from both helpfulness-based advantage $\hat{A}^{\hat{r}_t}$ and cost-based advantage $\hat{A}^{\hat{c}_t}$. [3] The Lagrange multiplier λ itself is updated based on the current cost violation:

$$\ln\lambda_{k+1} = \ln\lambda_k + \alpha\cdot\lambda_k\cdot\mathcal{J}_C(\theta_k) \tag{13}$$

where α is a learning rate for λ.

4.4 Iterative Fine-tuning and Red-Teaming

The Peking-SafeRLHF paper demonstrates the effectiveness of their approach through iterative application. After each round of Safe RLHF, the resulting model can be subjected to "red-teaming"—adversarial attempts to elicit harmful responses. Prompts that successfully bypass the safety measures are then incorporated into the dataset for subsequent rounds of preference data collection and model training. [3] This iterative loop of vulnerability discovery and model refinement is crucial for progressively enhancing the model's safety robustness against a wider range of potential misuse scenarios. The authors show that over three such iterations, their "Beaver" models significantly improved in both helpfulness and harmlessness. [3] This iterative refinement process underscores that achieving robust safety is not a one-time fix but an ongoing endeavor.

5 Comparative Analysis

The Fudan-PPO [2] and Peking-SafeRLHF [3] papers, while both contributing to the advancement of RLHF for LLMs, address different facets of the alignment challenge. Fudan-PPO is primarily concerned with the *how* of RLHF: stabilizing and optimizing the PPO algorithm itself to make the general alignment process more robust and efficient. In contrast, Peking-SafeRLHF focuses on the *what*: developing a specific framework to achieve a nuanced balance between helpfulness and harmlessness, two key but potentially conflicting alignment objectives.

In terms of PPO handling, Fudan-PPO offers a deep dive into a suite of "tricks" and meticulous implementation details (PPO-max) aimed at improving PPO's general performance, stability, and mitigating issues like pattern collapse. Peking-SafeRLHF, while utilizing PPO as the underlying RL optimizer within its constrained optimization framework (incorporating standard elements like KL penalty and a pretraining objective [3]), does not delve into PPO-specific micro-optimizations to the same granular extent as Fudan-PPO. Its main innovation lies in the architecture of the preference feedback (decoupled Reward and Cost Models) and the optimization strategy (Lagrangian method).

Regarding the reward and preference mechanism, Fudan-PPO operates under the assumption of a standard single reward model approach, aiming to maximize a unified preference signal. Peking-SafeRLHF fundamentally alters this by introducing separate Reward Models for helpfulness and Cost Models for harmlessness, acknowledging the distinct nature and potential conflict between these values. This allows for more targeted feedback and control.

Safety integration also differs. Fudan-PPO addresses safety more implicitly; by training on datasets like HH-RLHF for the reward model [2] and by achieving stable PPO training, the expectation is that good general alignment will lead to safer behavior. Peking-SafeRLHF, however, makes safety an explicit, first-class concern, modeling and constraining harmlessness as a primary objective through its Cost Model and constrained optimization formulation.

Despite these differences, both papers acknowledge common underlying challenges. The difficulty of accurate reward modeling is a shared concern: Fudan-PPO states "the quality of the reward model directly determines the upper bound of the policy model" [2], while Peking-SafeRLHF's framework relies on the accuracy of both its Reward and Cost models. Both are also fundamentally concerned with training stability and effective policy optimization, though Fudan-PPO focuses more on PPO's internal algorithmic stability, and Peking-SafeRLHF on the stability of balancing conflicting high-level objectives. Table 1 provides a summarized comparison of the two approaches.

Table 1: Comparison of PPO-max (Fudan-PPO) and Safe RLHF (Peking-SafeRLHF) Approaches

Feature	**PPO-max (Fudan-PPO) [2]**	**Safe RLHF (Peking-SafeRLHF) [3]**
Primary Goal	Stable and effective PPO training for general LLM alignment.	Robustly balancing helpfulness and harmlessness in LLMs.
Key Innovation	PPO-max: a collection of carefully calibrated PPO implementation "tricks" and best practices for stability.	Decoupled preference models (Reward Model for helpfulness, Cost Model for harmlessness) and Lagrangian-based constrained optimization.
PPO Handling	Deep optimization of PPO components, hyperparameters, and monitoring metrics to prevent issues like pattern collapse.	Uses PPO as the RL optimization algorithm within its constrained multi-objective framework.
Reward/Preference Mechanism	Assumes a standard single Reward Model capturing overall human preference.	Employs separate Reward Model (for helpfulness) and Cost Model (for harmlessness) based on decoupled human annotations.
Safety Integration	Implicitly through general alignment goals and training Reward Models on datasets containing safety preferences (e.g., HH-RLHF).	Explicitly models and constrains harmlessness as a primary objective using the Cost Model and a safety threshold in the optimization.
Main Challenges Addressed	PPO instability, pattern collapse, sensitivity to hyperparameters, difficulty in monitoring RLHF training.	Inherent tension between helpfulness and harmlessness, achieving quantifiable safety guarantees, avoiding catastrophic safety failures.

These distinct focuses suggest that the two papers address different layers of the RLHF problem stack. Fudan-PPO is working at the "RL algorithm layer," aiming to make the core training engine more reliable. Peking-SafeRLHF operates at the "objective definition and safety layer," determining what the LLM should be trained for and how to manage conflicting goals. This implies a potential for interplay: the PPO-max techniques for stable PPO training from Fudan-PPO could be directly integrated into the policy optimization step of the Peking-SafeRLHF framework. For example, when optimizing the Lagrangian objective in Safe RLHF (e.g., Eq. 29, 30 in [3]), the underlying PPO algorithm could benefit significantly from PPO-max's stability enhancements, such as advanced normalization, clipping strategies, and critic pre-training. Furthermore, the detailed monitoring metrics proposed by Fudan-PPO (perplexity, KL divergence, response length) could be invaluable for tracking policy behavior within the Safe RLHF loop, offering insights beyond just the aggregate reward and cost scores and potentially detecting subtle forms of gaming or instability earlier.

Both papers, despite their differing primary concerns, converge on the critical understanding that a simple, single scalar reward signal is often insufficient for the complex task of LLM alignment. Fudan-PPO demonstrates the need for auxiliary metrics and a collection of PPO "tricks" precisely because the primary reward signal can become misleading, leading to phenomena like pattern collapse. Peking-SafeRLHF explicitly argues for and implements a separate *cost* signal in addition to a reward signal, because a single reward function struggles to adequately capture and balance the trade-offs inherent in complex, multi-faceted objectives like helpfulness versus harmlessness. This shared realization points towards a broader trend in RLHF: the necessity for more sophisticated, multi-signal, or multi-objective approaches to guide LLM behavior effectively and reliably.

6 Conclusion

Reinforcement Learning from Human Feedback has undeniably transformed the landscape of Large Language Model development, providing a powerful mechanism to align these sophisticated AI systems with human values and intentions. This review has focused on two recent and significant contributions to this field: the Fudan-PPO paper's meticulous dissection of Proximal Policy Optimization for enhanced training stability, and the Peking-SafeRLHF paper's novel framework for robustly balancing helpfulness and harmlessness.

The Fudan-PPO paper illuminates the often-overlooked "secrets" within PPO implementation, demonstrating that careful engineering, including strategies for score reparameterization, policy constraints, and appropriate initialization (collectively termed PPO-max), is critical for overcoming instability and issues like pattern collapse. Their work underscores that success in RLHF is not just about the core algorithms but also about the practical wisdom gained through empirical investigation and the development of insightful monitoring metrics.

The Peking-SafeRLHF paper addresses a different but equally important challenge: the inherent tension between desirable LLM behaviors, particularly helpfulness and harmlessness. By proposing a framework that decouples human preferences for these aspects, trains separate Reward and Cost Models, and employs a Lagrangian-based constrained optimization, they offer a more principled and adaptive approach to safety alignment. Their iterative methodology, incorporating red-teaming, further highlights that achieving robust safety is an ongoing process of discovery and refinement.

Together, these papers signify a maturation of the RLHF field. The focus is shifting from merely demonstrating that alignment is possible to understanding how to achieve it reliably, robustly, and for complex, potentially conflicting human values. Fudan-PPO contributes to the reliability and robustness of the core RL training engine, while Peking-SafeRLHF provides tools for navigating the complexities of multi-faceted value alignment.

Despite the progress exemplified by these works, RLHF remains a technology with significant open challenges. Reward hacking, data quality and bias, evaluation difficulties, and computational scalability continue to be active areas of research.[7] However, the ongoing exploration of algorithmic improvements (in PPO, Dynamic Policy Optimization (DPO), and beyond), more sophisticated reward and cost modeling techniques, innovative data collection and generation strategies, and more comprehensive evaluation methodologies paint a promising picture. The commitment to open-sourcing code and models, as demonstrated by the Fudan-PPO team, is also vital for accelerating community-wide progress.

In conclusion, RLHF is a dynamic and critical area of AI research. The insights and methodologies from papers like Fudan-PPO and Peking-SafeRLHF are instrumental in advancing the development of LLMs that are not only powerful but also increasingly aligned with human goals, paving the way for more beneficial and trustworthy AI systems.

References

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Christiano, and P. Welinder, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022.

[2] R. Zheng, S. Dou, S. Gao, S. Jin, Q. Liu, N. Xu, W. Lai, Y. Hua, W. Shen, B. Wang, Y. Liu, Y. Zhou, L. Xiong, L. Chen, Z. Xi, M. Zhu, C. Chang, Z. Yin, R. Weng, W. Cheng, H. Huang, T. Sun, H. Yan, T. Gui, Q. Zhang, X. Qiu, and X. Huang, "Secrets of RLHF in Large Language Models Part I: PPO," *arXiv preprint arXiv:2307.04964*, 2023.

[3] J. Dai, X. Pan, R. Sun, J. Ji, X. Xu, M. Liu, Y. Wang, and Y. Yang, "Safe RLHF: Safe Reinforcement Learning from Human Feedback," *arXiv preprint arXiv:2310.12773*, 2023.

[4] E. Lian, S. Budhathoki, A. Head, and D. Sasha, "Aligning to What? On the (In)Effectiveness of RLHF in Mitigating Covert Biases," *arXiv preprint arXiv:2403.09025*, 2024.

[5] H. Zhang, R. Liu, R. Miao, Z. Zhang, W. Zhang, Y. Yu, and S. Wang, "The Energy Loss Phenomenon in RLHF: A New Perspective on Mitigating Reward Hacking," *arXiv preprint arXiv:2401.12358*, 2024.

[6] J. Fu, X. Zhao, C. Yao, H. Wang, Q. Han, and Y. Xiao, "Reward Shaping to Mitigate Reward Hacking in RLHF," *arXiv preprint arXiv:2402.18770*, 2024.

[7] S. Casper, X. Davies, C. Shi, T. Krendl Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, T. T. Wang, S. Marks, C.-R. Segerie, M. Carroll, A. Peng, P. Christoffersen, S. Raje, M. Prakash, E. Razin, R. Sigal, N. Zorowitz, M. Hobbhahn, A. Human, D. Krasheninnikov, A. Grimsley, D. Ackley, E. Kuelbs, P. Chan, S. Ghosh, J. Kaddour, M. Heiner, and D. Krueger, "Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback," *Transactions on Machine Learning Research*, 2023.