The Role of Feedback Variety in Reinforcement Learning from Human Feedback

Ivan Makarov[1]
Supervisor(s): Luciano Cavalcante Siebert[1]**, Antonio Mone**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Ivan Makarov
Final project course: CSE3000 Research Project
Thesis committee: Luciano Cavalcante Siebert, Antonio Mone, Wendelin Böhmer

Abstract

Reinforcement Learning from Human Feedback (RLHF) offers a powerful approach to training agents in environments where defining an explicit reward function is challenging by learning from human feedback provided in various forms. This research evaluates three common feedback types within RLHF: Scalar Feedback, Binary Comparison Feedback, and Binary Comparison with a preference strength margin. Synthetic feedback is used to replace real human feedback to address cost and time constraints. Simplified RLHF setups using Q-learning are initially implemented in a grid environment to ensure the robustness of the methods. Subsequent experiments are conducted in more complex environments using the Imitation library and PPO from Stable Baselines3. Our findings demonstrate the efficacy of various feedback types, highlighting the trade-offs between ease of use for human feedback providers and the amount of information conveyed. This comparative analysis provides insights into optimizing RLHF systems for improved agent performance. Full code is available online in the supplementary material https://github.com/navimakarov/rlhf-feedback-variety.

1 Introduction

In reinforcement learning (RL), an agent is directed through an environment where it learns to make decisions based on a process of trial and error. The effectiveness of these decisions is evaluated through a reward function. Properly defining a reward function is crucial for the agent to learn desired behaviours. Creating the reward functions manually can be quite challenging, especially since many tasks involve goals that are difficult to define precisely. For example, getting a robot to perform a backflip is easy to evaluate but challenging to specify a reward function for [1].

In response to this problem, reinforcement learning from human feedback (RLHF) [1] has been proposed as a powerful tool to train agents when the reward function is hard to specify or human judgment can improve training efficiency. In RLHF, human feedback on the agent's actions is used to train a reward model to translate human preference into learned reward signals. While a human could, in theory, directly assign rewards to each of the agent's actions, taking on the role of the reward function, this may be effective only in environments with a limited number of states and actions (e.g., simple grid environments). However, this approach becomes impractical in more complex environments due to the substantial effort required to consistently assign rewards and the absence of generalization. Consequently, it is more feasible to employ a trainable reward model. RLHF has a wide range of applications, including the refinement of large language models (LLMs) based on human preferences [2], [3], continuous control systems [1], and games [4].

In RLHF, various types of human feedback are utilized to convey information about preferences. Feedback in RLHF can vary in granularity, such as state preference, action preference, or multiple state-actions preference (trajectory) [5]. It can be given on one instance or many, comparing multiple trajectories. Feedback may be explicit or implicit and can be in different forms, such as a binary preference, a scalar, or natural language [6]. For example, RLHF on LLM chatbots is sometimes performed using conversation pairs, with feedback given in the form of preferences indicating which generated text is more preferred. As outlined in [7], "RLHF suffers from a tradeoff between the richness and efficiency of feedback types."

Building on the existing literature, a notable gap is that no studies have specifically focused on the comparative evaluation of different types of feedback in RLHF. Surveys such as [6] provide a broad overview of RLHF components but do not focus specifically on feedback

types comparison. Research focusing on specific types of feedback [1], [8] investigates individual feedback mechanisms but lacks a comprehensive review. Research on RLHF systems, such as those by [9] and [10], primarily focuses on designing and developing the platforms, including various components like annotation systems, usability design, dataset collection, and testing with actual human experts. While these platforms incorporate different feedback types, they do not prioritize comparing the effectiveness of these feedback types. In [10], comparing different feedback performances is included as a potential future research direction.

Given the variety of feedback types and their respective tradeoffs, it is essential to thoroughly explore both their methodologies and comparative advantages. This research aims to investigate how different feedback types can be integrated into RLHF systems and provide an extensive comparison between these feedback types to identify tradeoffs and guide practitioners in designing better RLHF systems. By comparing the respective advantages and disadvantages of different feedback types in multiple environments, we aim to offer valuable insights for improving RLHF implementations and selecting the most appropriate feedback type for specific scenarios.

2 Background

2.1 Foundations of RLHF

As formulated in [6], in the RLHF setting, the learning agent must solve a task without a predefined reward function. To do this, the agent usually learns an approximation of the reward function from human feedback alongside an RL policy. Consequently, a typical RLHF algorithm repeats two phases: (1) reward learning and (2) RL training. The reward learning phase can be further divided into two steps: (i) generating queries for the oracle and (ii) training a reward function approximator based on the oracle's responses. The RL training phase is more conventional and typically involves running an RL algorithm using the currently trained reward function approximator. We outline a generic RLHF Algorithm seen in the literature [1], [6], [7] and implemented in [11]:

Algorithm 1 Generic RLHF Algorithm

Initialize reinforcement learning (RL) model π_θ and reward model $\hat{r}_\theta$
for $i = 1, \ldots, N$ **do**
    Rollout trajectories using the current RL model π_θ
    Generate a query from the collected trajectories
    Collect human feedback for the generated query
    Update the reward model $\hat{r}_\theta$ by minimizing the loss function on the queries and human feedback
    Train the RL model π_θ using the updated reward model $\hat{r}_\theta$
end for
Train a new RL model using the final learned reward model $\hat{r}_\theta$

As the reward model learns from human feedback, it is crucial to first understand the various types of human feedback in the RLHF setting, how these different types are integrated into RLHF systems, and the loss functions used to train reward models.

2.2 Dimensions of Human Feedback

As summarized in [6], feedback types vary across multiple dimensions, including the method of feedback delivery (arity, involvement), the form of the query instance (granularity, abstraction), and features of human interaction (intent, explicitness). In this subsection, we outline the structuring of human feedback, inspired by the taxonomies proposed in [6], [9], which helps to understand and categorize the different ways feedback can be provided and processed.

Arity This attribute indicates if an instance is evaluated in isolation (unary) or relative to others (binary, n-ary). Unary feedback allows for detailed descriptions, but giving consistent feedback is challenging for humans. Non-unary feedback is less demanding but requires comparable instances. N-ary feedback, like rankings, offers more information than binary feedback but increases the cognitive load on the labeller.

Involvement The labeller may either passively observe an instance, actively generate it, or coactively participate in its generation (co-generation). Passive involvement poses the smallest challenge to the labellers as it does not require the ability to demonstrate the task.

Granularity There are three different types of feedback granularity commonly found in the literature [5]: action, state, and trajectory feedback. Each type presents unique challenges for both the expert providing the feedback and the algorithm processing it.

- Action Feedback: Involves giving feedback on actions in the same state, indicating which action should be preferred. This is demanding for the expert, who needs to understand the long-term outcomes, but it is computationally simpler for the algorithm.
- State Feedback: Compares states, suggesting that one state is preferable over the other based on available actions. This offers more information than action feedback but still requires the expert to estimate future outcomes, making it less demanding but still complex.
- Trajectory Feedback: This is the most informative and least demanding for the expert, as it involves evaluating the overall outcomes of full sequences of states and actions. This is the most widely used form of feedback but presents a challenge for the algorithm in determining which parts of the trajectory are responsible for the preference, especially when starting from different initial states.

Abstraction This describes whether feedback is given directly on raw instances (e.g. trajectories) or on abstract features of the instances.

Intent Human feedback can be evaluative, instructive, or descriptive, aiming to teach a reward function (pedagogical), or literal, a byproduct of direct reward optimization. Evaluative, instructive, and literal feedback typically address specific queries, whereas descriptive feedback can provide a broader overview of the task, such as through a partial reward function.

Explicitness Humans may communicate explicitly for the purposes of feedback or implicitly as a side-effect of actions directed at other purposes.

2.3 Feedback Types

A variety of feedback types can be found in the literature. We focus on the most common classes of feedback for RLHF as detailed in [9], [7], [6], [10].

2.3.1 Evaluative Feedback

As outlined in [6], the key characteristics of evaluative feedback are that the human passively observes the behaviour (involvement), provides feedback on a single instance (arity), and does so explicitly (explicitness) with an evaluative intent. This feedback can be given at any level of granularity and abstraction. A teacher assigns a scalar value y to a segment σ^i, and as feedback is collected, it is stored as tuples (σ^i, y) in the reward model data set D. We then apply standard regression and update $\hat{r}_\theta$ by minimizing the mean squared error:

$$L^{\text{MSE}}(\theta, D) = \frac{1}{|D|} \sum_{(\sigma^i, y) \in D} \left(y - \hat{r}_\theta(\sigma^i)\right)^2 \tag{1}$$

This type of feedback is relatively easy to utilize for reward models because it can serve as a prediction target [9].

2.3.2 Comparative Feedback

Comparative Feedback is the most common type of feedback used with RLHF, as stated in [7]. As outlined in [6], the key characteristics of comparisons are that the human passively observes the behaviour (involvement), provides relative feedback on multiple instances (arity), and does so explicitly (explicitness) with an evaluative intent. This feedback is most commonly given on a segment (granularity) and is often requested at an instance level (abstraction).

The most common setting relies on pairwise comparisons of trajectory segments [1]. To define a preference predictor using the reward function $\hat{r}_\theta$ the Bradley-Terry model [12] is used in [1].

$$P_\theta(\sigma^1 > \sigma^0) = \frac{\exp\left(\sum_t \hat{r}_\theta(s_t^1, a_t^1)\right)}{\sum_{i \in \{0,1\}} \exp\left(\sum_t \hat{r}_\theta(s_t^i, a_t^i)\right)} \tag{2}$$

Some papers, such as [3], use an alternative notation of a preference predictor that is mathematically equivalent, using a sigmoid function:

$$P_\theta(\sigma^1 > \sigma^0) = \text{sigmoid}\left(\sum_t \hat{r}_\theta(s_t^1, a_t^1) - \sum_t \hat{r}_\theta(s_t^0, a_t^0)\right) \tag{3}$$

We update $\hat{r}_\theta$ by minimizing the standard binary cross-entropy objective:

$$L^{\text{BCE}}(\theta, D) = -\frac{1}{|D|} \sum_{(\sigma^0, \sigma^1, y) \in D} \left((1-y) \log P_\theta(\sigma^0 > \sigma^1) + y \log P_\theta(\sigma^1 > \sigma^0)\right) \tag{4}$$

Another method that exists is the ranking of multiple targets, as described in [13].

Comparative feedback is widely used because it is often easier for humans to give comparative judgment compared to absolute scores [5]. However, the above-mentioned comparative feedback methods do not offer precise information on the intensity of preferences [7]. To bridge this gap, an adjustment to pairwise comparisons feedback was made in [2] for the training of LLama2. The idea is to include a margin to utilize additional information on the

intensity of preferences to explicitly train the reward model to assign more distinct scores to segments with greater differences. The loss function is not modified, but the preference predictor is updated to include the margin:

$$P_\theta(\sigma^1 > \sigma^0) = \text{sigmoid}\left(\sum_t \hat{r}_\theta(s_t^1, a_t^1) - \sum_t \hat{r}_\theta(s_t^0, a_t^0) - m(r)\right) \tag{5}$$

where the margin $m(r)$ is a discrete function of the preference rating. A larger margin is used for pairs with higher preference intensity and a smaller margin for those with smaller preference intensity.

2.3.3 Corrective Feedback

As detailed in [6], corrective feedback is a form of feedback in which a human refines the agent's behaviour, either by intervening while the agent acts or by providing a corrected behaviour after the agent has acted. To improve an episode, it is usually necessary to observe the entire episode (granularity) at the instance level (abstraction). In this type of feedback, the human both observes and demonstrates behaviour, resulting in co-generative involvement. Comparative feedback usually involves improving a single trajectory (unary arity).

3 Methodology

3.1 Feedback Types

Among the common classes of feedback types specified in Section 2, we focused on Evaluative Feedback and Comparative Feedback. These classes are prevalent in the literature [7], [6], [9] and require passive involvement from the teacher, unlike Corrective Feedback, which necessitates making corrections to given trajectories. Additionally, it is relatively simple to gather synthetic feedback for these classes compared to natural language feedback, which is more nuanced and cannot be easily generated. Within these two feedback classes, we select the following feedback types:

Scalar Feedback is considered to be the most direct type of feedback [6]. In this type of feedback, the human teacher assigns numerical ratings to segments of trajectories. The precision of Scalar Feedback comes at the cost of being challenging for humans to indicate rewards accurately. Human annotators often find it challenging to quantify the success of an example, and this task demands more cognitive effort compared to simply comparing examples [7].

Preference Feedback, as introduced in [1], is the most popular feedback type for RLHF [7]. This type of feedback is much easier for humans than Scalar Feedback, as it only requires indicating which trajectory is preferred. However, this simplicity comes at the cost of reduced information conveyed, as the only information provided is the preference between trajectories [7].

Marginal Preference Feedback takes an intermediate position between Scalar Feedback and Preference Feedback. It conveys more information than binary trajectory preference by quantifying the extent of preference for one trajectory over another. Although this type of feedback requires more effort from humans than Preference Feedback, it is less demanding than providing precise Scalar Feedback.

In this research, we opted for trajectories as our feedback granularity, as this method is the most widely used and conveys the most information [5].

3.2 Synthetic Feedback

In this study, we use synthetic feedback instead of real human feedback for our RLHF models. This approach is adopted because obtaining real human feedback is more expensive and time-consuming. To achieve this, we collect real rewards from the environment and use them as scalar values for evaluative feedback or as a means to compare trajectory segments for comparative feedback. We also simulate the inaccuracy of user-provided rewards for evaluative feedback by applying a random error within a specified range to mimic human mistakes. For marginal preference feedback, we use a margin to speed up learning for trajectories with significant differences in true rewards, as shown in Section 2.3.2. Trajectories with minimal differences are assigned a margin of 0, while those with large differences receive the maximum margin, with intermediate values for others. This quantifies the significance of differences between trajectories (e.g., slightly better, significantly better). As shown in [2], the margin value is a hyperparameter. For synthetic feedback, we define intervals to specify the significance of differences. For example, differences less than 10 may be insignificant, 10 to 20 slightly important, 20 to 30 important, and over 30 significantly important. Margins are assigned based on these intervals to simulate human feedback. If it were real feedback, human evaluators would directly quantify the significance of the differences. Therefore, we consider intervals as heuristics and choose them based on the information about the ground-truth reward function for a given environment.

3.3 Implementation of RLHF

The Imitation library [11] provides a framework for training reward models using RLHF with synthetic feedback by following the standard RLHF procedure described in Section 2.1. It implements only one type of feedback: Preference Feedback (pairwise comparisons of trajectory segments). Instead of human feedback, it uses synthetic feedback, which is obtained by directly comparing ground-truth rewards from the environments for the pair of trajectories. The trajectory with a higher reward is preferred. The preferred trajectory receives a score of 1, and the other trajectory receives a score of 0. In case the trajectories get equal ground-truth rewards, they both receive 0.5 as scores.

As shown in Algorithm 1, the loss function alongside functions for query creation from the trajectories are the backbone of different feedback types. To add support for a new feedback type, we need to make modifications to these functions. These changes correspond to steps 4-6 of Algorithm 1. We extended the synthetic feedback gatherer by adding a field with a numeric indicator of preference strength for Marginal Preference Feedback. We also added an option for Scalar Feedback, which consists of a single trajectory with a ground-truth reward. For this type of feedback, we implement a feature to add noise by introducing random deviations from the ground-truth reward, thereby simulating less accurate responses. Noise is disabled by default but is used in one of our experiments, further detailed in Section 4.2. As the trajectories with synthetic feedback are collected and fed into the reward model, we also define loss functions corresponding to each feedback type as detailed in Section 2.3.

Our adjustments can be summarized by the following steps:

1. We create an appropriate query for the selected feedback type from rolled-out trajectories. This consists of a single trajectory for Evaluative Feedback and a pair of

trajectories for Comparison Feedback.

2. We define rules to give synthetic feedback on queries (emulating human feedback) based on trajectories and their ground-truth rewards. For Comparison Feedback, we prefer a trajectory with a higher true reward. For Evaluative Feedback, we use the true reward as our scalar.

3. We implement an appropriate loss function for each feedback type.

A simplified RLHF setup is implemented to validate our feedback types and implementation methods as proof of concept. We reuse parts of the Imitation library with Q-learning to avoid the instability associated with deep learning models used with it. After completing this feasibility evaluation, experiments are conducted with the Imitation library in more complex environments. We use Proximal Policy Optimization (PPO) [14] from stable baselines3 [15] as our deep learning model alongside the Imitation library. We chose PPO because it is widely used for RLHF tasks [6] and achieves expert-level performance in selected environments with a known reward function (standard Reinforcement Learning).

4 Experimental Setup

4.1 Environments

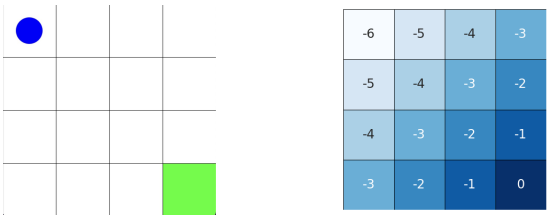
In the literature, MuJoCo [16] and the Arcade Learning Environment [17], interfaced through the Gymnasium [18] are often used to evaluate RLHF methods [1], [4], [8], [10]. These environments are complex and require extensive training to reach an expert level. Therefore, sometimes simpler environments are also used. An example of such an environment is a simple cartpole task, "Pendulum", which is used alongside the above-mentioned complex environments for evaluation of RLHF with comparative feedback in [1]. Simple grid environments are less frequently used with RLHF, however, they can also be included as shown in [10].

In the scope of this paper, three environments were selected for evaluation. We chose Pendulum-v1 and seals/CartPole-v0 from Gymnasium's [18] "Classic Control Environments," as such tasks have been used in the literature [1] and serve to demonstrate and evaluate RLHF methods effectively. Pendulum does not require any additional adjustments to work with the Imitation library. CartPole has early episode termination, which is discouraged in RLHF, as it might leak information about rewards through side channels (early termination). Therefore, seals library [19] is used to make CartPole suitable for RLHF.

While "Classic Control Environments" are suitable for showcasing differences between RLHF methods, they suffer from instability due to the use of deep reinforcement learning in model training. Therefore, the first environment we use to evaluate our RLHF models is a simple grid environment to demonstrate the effectiveness of our methods and show that models can achieve expert-level performance with selected RLHF feedback types.

Simple Grid Environment is inspired by the MiniGrid-Empty environment [20]. The goal of this environment is to demonstrate the convergence of different RLHF feedback types at the expert level. It is simpler to identify implementation mistakes using basic reinforcement learning methods such as Q-learning rather than deep reinforcement learning. This is a 4x4 grid environment with 16 observation states and 4 possible actions in each state: left, right, up, and down. The agent starts in the top left corner, and the goal is located in the bottom right corner. In each state, the reward is the negative Manhattan distance to the goal. The

environment is depicted in Figure 1a, and the rewards are visualized in Figure 1b. In Figure 1a, the blue ball represents the agent in the starting state, and the green tile represents the goal state. As RLHF requires a fixed episode length to prevent the agent from learning through side channels, the episode length is set to 10 steps with no early termination. This is achieved by transitioning the agent into an absorbing state upon reaching a reward state. In the absorbing state, the reward is fixed at 0, and actions are disregarded, meaning the state does not change.



(a) Environment (b) Rewards

Figure 1: Simple Grid Environment

4.2 Evaluations

We evaluate all feedback types on the Simple Grid Environment first. For each feedback type, we used 200 queries and trained our Q-Learning model for 25 steps after each query. For Evaluative Feedback, we roll out one trajectory, gather its true reward and feed it to the reward network. We roll out a pair of trajectories for Preference Feedback, decide which trajectory is preferred based on their ground-truth rewards, and feed this pair comparison to the reward network. For Marginal Preference Feedback, we do the same with a margin of 10 and reward difference significance intervals of [10, 20, 30, 40, 50] chosen heuristically based on the environment rewards. As our Q-Learning balances exploration with exploitation, by sometimes selecting random action, we average our results over 5 runs with different seeds for consistency and reproducibility. We also include a randomly initialized network, which is not trained on any feedback, to contrast with the expert performance we are achieving with our feedback methods.

For "Classic-Control Environments", we use the Imitation library, which handles trajectory gathering with true rewards for us. We run 5 evaluations with different seeds for each environment and feedback type and average the results. We constantly measure mean rewards throughout the training process, which we later use to analyse the training progress. We also include error bars in our plots, which consist of standard error over 5 runs. Standard error was preferred over standard deviation, as we want to show how far from the mean the results are. We also report the mean reward and standard error of the models after training is finished and include them in tables to compare different feedback types.

For Marginal Preference Feedback on Pendulum, we use an interval of [200, 400, 600, 800, 1000] and a margin of 100. For CartPole, we use an interval of [20, 40, 60, 80, 100] and a margin of 100. We believe that these intervals and margins are good heuristics for the given environments according to their reward ranges.

For Pendulum environment, we also want to show that if human gives approximately accurate evaluations, the model is still capable of achieving good performance. Given that

the true rewards of Pendulum were observed in the range of [-1200, -180], we consider it reasonable to model a human-simulated scalar by the following formula:

$$\text{rew}_{\text{modified}} = min(\left\lfloor \frac{\text{rew}_{\text{true}}}{100} \right\rfloor \pm 3, 0) \tag{6}$$

5 Results

In this section, we present the comparative results of using RLHF with selected feedback types across three different environments.

5.1 Evaluation 0: Proof of Concept Simple Grid Environment

Figure 2 shows that all feedback types - Evaluative, Binary Preference, and Marginal Binary Preference - reach an expert-level reward, indicating the maximum possible reward achievable in this environment. In contrast, the use of a randomly initialized neural network for estimating human preferences, without any feedback, fails to perform the task, as demonstrated by the consistently low reward. This environment demonstrates the efficacy of the feedback methods, however, it does not highlight their differences, which become more apparent in the more complex environments discussed further.

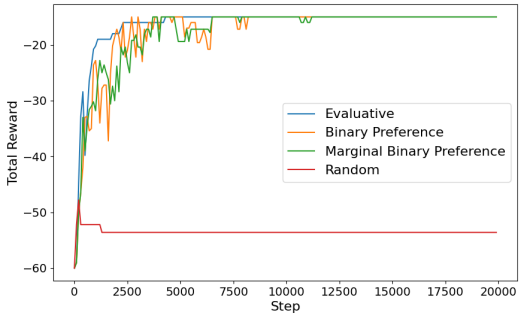
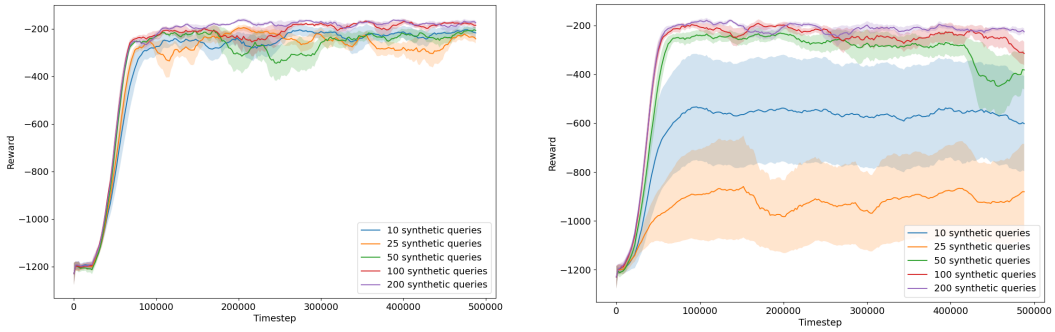


Figure 2: RLHF Grid Environment Training

5.2 Evaluation 1: Pendulum

We use Evaluative Feedback with a ground-truth scalar to demonstrate how quickly our reward model can learn when the evaluator provides very precise scalar feedback, as depicted in Figure 3a. Even with a small number of queries, the reward model learns efficiently, and the agent's performance converges to an expert level with a very small variance. Given the complexity of the task of providing precise scalar evaluations, Figure 3b shows the results when the human evaluator simplifies the task and provides less accurate scalars, as described in Section 4. This significantly degrades performance in experiments with a small number of queries, but the agent still achieves an expert-level performance with 100 and 200 queries. This approach also leads to high variance for runs with a small amount of queries. As outlined in Table 1, the ground truth scalar consistently outperforms the human-simulated scalar on this task, with even a small number of queries being sufficient to achieve favourable results.

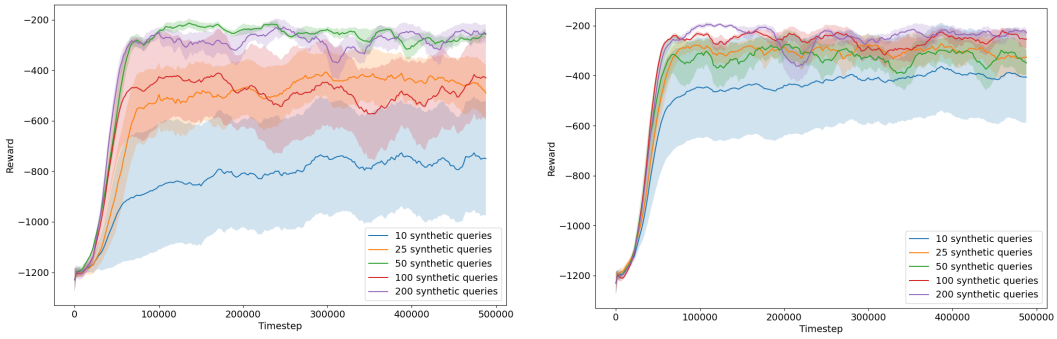


(a) Ground-Truth Scalar

(b) Human-Simulated Scalar

Figure 3: Pendulum Evaluative Feedback

We confirm that Preference Feedback while being easier for humans to give, conveys less information and requires more queries to achieve good performance, as shown in Figure 4a. We found that a higher number of queries does not necessarily lead to better performance. As shown in Table 1, 50 queries yield the best result, 200 queries are second best, and 100 queries are even worse than 25 queries. We also report high overall variance, as was the case with human-simulated scalar feedback. However, while in Human-Simulated Scalar feedback, more queries led to lower variance, this is not the case for Preference Feedback. This might be due to many reasons, such as possible overfitting with more queries, quality of gathered trajectories or instability of deep reinforcement learning models. This behaviour is not unusual and was also found in [1].



(a) Preference Feedback

(b) Marginal Preference Feedback

Figure 4: Pendulum Comparative Feedback

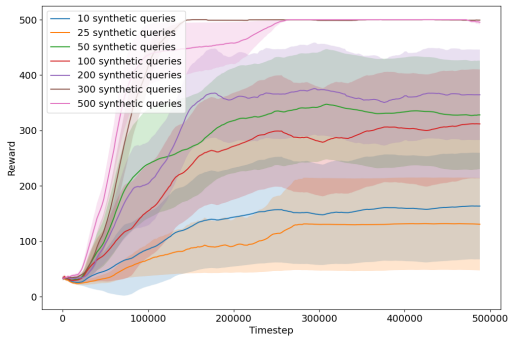
We observe in Figure 4b that Marginal Preference Feedback leads to better performance and lower variance. Moreover, this method outperforms Human-Simulated Scalar on queries 10, 25 and 50 but underperforms on queries 100 and 200.

Queries	Ground-Truth Scalar	Human-Simulated Scalar	Preference	Marginal Preference
10	-288.100 ± 36.223	-845.035 ± 231.174	-1030.344 ± 205.156	-506.565 ± 238.392
25	-289.180 ± 27.869	-816.269 ± 216.278	-369.323 ± 103.125	-252.978 ± 43.340
50	-252.654 ± 40.896	-349.816 ± 84.352	-239.005 ± 37.527	-334.968 ± 40.784
100	-211.503 ± 26.023	-229.678 ± 16.206	-432.061 ± 149.745	-268.170 ± 35.450
200	-182.012 ± 14.294	-210.494 ± 9.569	-308.693 ± 80.716	-269.397 ± 53.279

Table 1: Comparison of Feedback Types (Pendulum)

5.3 Evaluation 2: CartPole

As CartPole has sparse rewards, ground-truth scalar feedback does not converge to near-perfect evaluation scores as quickly as with the Pendulum. However, with 300 and 500 queries, it achieves a perfect score, as shown in Figure 5a and Table 2. Higher variance can also be explained by sparse rewards in the environment. As in the Pendulum evaluation, with more queries, we achieve better performance.

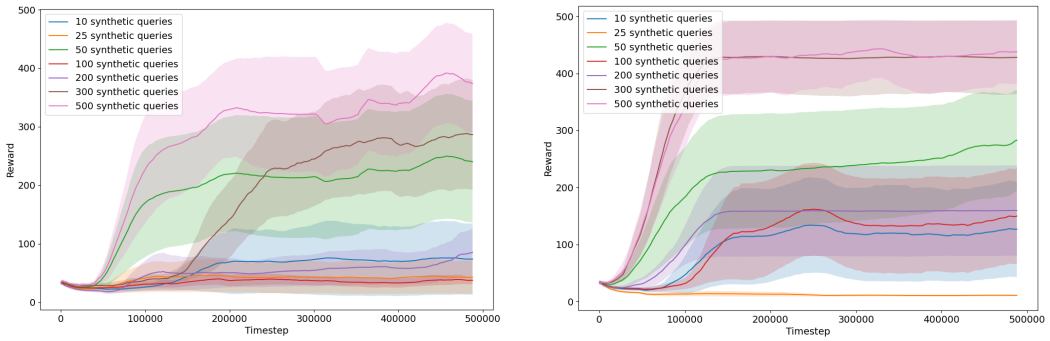


(a) Ground-Truth Scalar

Figure 5: CartPole Evaluative Feedback

With Preference Feedback, we need 500 queries to achieve good performance, as shown in Figure 6a and Table 2. The results, however, are still significantly worse than those of the Ground-Truth Scalar. We also observe increasing queries does not necessarily lead to better performance.

Marginal Preference Feedback outperforms Preference Feedback when the number of queries is higher than 50, improving average reward and reducing variance. This behaviour is not observed on a lower number of queries, likely because it is not enough to learn any meaningful reward estimation.



(a) Preference Feedback

(b) Marginal Preference Feedback

Figure 6: CartPole Comparative Feedback

Queries	Ground-Truth Scalar	Preference	Marginal Preference
10	14.226 ± 2.633	112.738 ± 86.649	106.143 ± 85.204
25	129.764 ± 82.392	49.858 ± 32.009	13.024 ± 2.674
50	179.652 ± 95.518	108.666 ± 87.508	201.362 ± 97.820
100	321.432 ± 96.420	42.052 ± 7.507	210.562 ± 105.655
200	370.910 ± 82.547	144.426 ± 84.794	160.81 ± 79.040
300	500.000 ± 0.000	287.856 ± 91.605	432.054 ± 60.773
500	494.816 ± 4.634	402.04 ± 87.618	450.048 ± 44.678

Table 2: Comparison of Feedback Types (CartPole)

6 Discussion

6.1 Results Analysis

We implemented RLHF using selected feedback types and demonstrated their effectiveness in a proof-of-concept environment, where all methods achieved the maximum possible reward. We compared the performance of RLHF with different feedback types in the CartPole and Pendulum environments. Our research empirically shows the trade-off between the richness and efficiency of feedback types [7].

We achieved fast convergence and low variance using Evaluative Feedback with a ground-truth reward, which conveys the most information among our feedback types but is very challenging for a human to provide. In our research, we used the actual reward, known beforehand, for ground-truth feedback. One might question the need for RLHF if the reward function is already known. This approach, however, can be applied to scalars not derived from a single formula, such as heuristics or approximate evaluations. We demonstrate this with our Pendulum example using human-simulated scalars, where, despite higher variance, we achieve expert level with a sufficient number of queries.

Preference Feedback, being less demanding from humans, requires more queries, gives lower mean rewards, and has higher variance than Evaluative Feedback. Nonetheless, given enough queries, it also achieves good performance. We noticed that the relationship between the number of queries and performance is not straightforward for preference feedback. However, this is already known in the literature [1]. We were able to achieve superior results

compared to Preference Feedback with Marginal Preference Feedback.

While our study used only synthetic feedback, it is important to note that RLHF can help agents perform tasks more aligned with human preferences and potentially achieve superior results, as shown in [1]. Our research did not observe these benefits due to the limitations of synthetic feedback.

6.2 Limitations

One limitation of our study is the time constraint, which prevented us from testing more complex environments, particularly MuJoCo environments that are frequently used for RLHF research. These environments require significantly more training time. For example, RLHF on MuJoCo environments in [1] required 20 million training timesteps, whereas Classic Control Environments only require 500,000 timesteps in our research.

Another limitation is the use of synthetic feedback. Real human feedback would provide a more accurate assessment of the agent's performance and might be able to identify overlooked aspects that synthetic feedback cannot capture. Humans may not be able to fully articulate their internal (user-optimal) reward function through feedback. Consequently, the feedback can be subject to noise or uncertainty due to inherent human irrationality [21]. While Evaluative Feedback is straightforward to specify, the inter-annotator agreement is often low due to the subjective nature of the task [22]. Collecting pairwise feedback can be challenging for nearly similar responses and may lead to significant time being spent by labellers on a single input [23]. Moreover, Preference feedback does not offer any improvement in inter-annotator agreement compared to Evaluative feedback [22]. It is problematic to select representative humans and obtain quality feedback, as some evaluators may have harmful biases and opinions, and individual evaluators can potentially poison data [7].

7 Responsible Research

A major limitation is the use of synthetic feedback rather than real data from human experts. While this approach avoids ethical concerns, it may compromise the representativeness and reliability of the results. Synthetic feedback might not fully capture the inconsistent behaviours and decision-making processes of real human experts, reducing the generalizability and practical applicability of the findings. Additionally, the choice of environments poses another constraint. Increasing the complexity of the environments could reveal behavioural issues that synthetic expert feedback may not address. Therefore, it is important to interpret the findings within the context of the chosen environments and synthetic feedback.

To mitigate confirmation bias, we used multiple environments and averaged results from several runs with different seeds. Synthetic queries also help reduce personal bias. To the best of our knowledge, the findings described in this research paper do not create an opportunity for exploitation by malicious parties.

Our experiments are fully reproducible, with the complete codebase, seeds, hyperparameters, and pre-saved models provided.

8 Conclusions and Future Work

The aim of this empirical exploration was to investigate how different feedback types can be integrated into RLHF and to evaluate their effectiveness. Our findings highlight the

tradeoff between the ease of providing feedback and the amount of information conveyed. Our insights can assist in the development of RLHF systems, offering evidence that different feedback types can be used based on specific application needs.

Our approach provides a detailed methodology building on existing literature for creating RLHF models by utilizing and extending the Imitation library. This can serve as a valuable resource for researchers aiming to implement RLHF in their work. Additionally, our exploration into less common feedback types, such as marginal preference and evaluative scalar feedback, can guide future studies looking to refine these methods.

By outlining these processes and demonstrating their effectiveness in various environments, our research contributes to the broader understanding of RLHF and its practical applications.

Future work could address the current limitations by incorporating real human feedback and running experiments on more complex environments to further validate and expand upon these findings. Additionally, exploring a wider variety of feedback types would be a valuable direction for future research. An especially intriguing avenue is the integration of multiple feedback types in the training of a single agent. This could be achieved by using an ensemble of neural networks, each trained with a specific loss function corresponding to a chosen feedback type, or by training the same reward network with different loss functions in sequence, such as using evaluative feedback for 50% of the training time and comparative feedback for the remaining 50%.

References

[1] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, *Deep reinforcement learning from human preferences*, 2023. arXiv: 1706.03741 [stat.ML].

[2] H. Touvron, L. Martin, K. Stone, *et al.*, *Llama 2: Open foundation and fine-tuned chat models*, 2023. arXiv: 2307.09288 [cs.CL].

[3] L. Ouyang, J. Wu, X. Jiang, *et al.*, *Training language models to follow instructions with human feedback*, 2022. arXiv: 2203.02155 [cs.CL].

[4] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei, "Reward learning from human preferences and demonstrations in atari", *Advances in neural information processing systems*, vol. 31, 2018.

[5] C. Wirth, R. Akrour, G. Neumann, and J. Fürnkranz, "A survey of preference-based reinforcement learning methods", *Journal of Machine Learning Research*, vol. 18, no. 136, pp. 1–46, 2017.

[6] T. Kaufmann, P. Weng, V. Bengs, and E. Hüllermeier, *A survey of reinforcement learning from human feedback*, 2023. arXiv: 2312.14925 [cs.LG].

[7] S. Casper, X. Davies, C. Shi, *et al.*, *Open problems and fundamental limitations of reinforcement learning from human feedback*, 2023. arXiv: 2307.15217 [cs.AI].

[8] D. White, M. Wu, E. Novoseller, V. J. Lawhern, N. Waytowich, and Y. Cao, *Rating-based reinforcement learning*, 2024. arXiv: 2307.16348 [cs.LG].

[9] Y. Metz, D. Lindner, R. Baur, D. Keim, and M. El-Assady, *Rlhf-blender: A configurable interactive interface for learning from diverse human feedback*, 2023. arXiv: 2308.04332 [cs.LG].

[10] Y. Yuan, J. Hao, Y. Ma, *et al.*, *Uni-rlhf: Universal platform and benchmark suite for reinforcement learning with diverse human feedback*, 2024. arXiv: 2402.02423 [cs.LG].

[11] A. Gleave, M. Taufeeque, J. Rocamonde, *et al.*, *Imitation: Clean imitation learning implementations*, arXiv:2211.11972v1 [cs.LG], 2022. arXiv: 2211.11972 [cs.LG].

[12] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons", *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[13] B. Zhu, J. Jiao, and M. I. Jordan, *Principled reinforcement learning with human feedback from pairwise or K-wise comparisons*, 2024. arXiv: 2301.11270 [cs.LG].

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, 2017. arXiv: 1707.06347 [cs.LG].

[15] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations", *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html.

[16] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control", in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109.

[17] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents", *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, Jun. 2013.

[18] M. Towers, J. K. Terry, A. Kwiatkowski, *et al.*, *Gymnasium*, Mar. 2023. DOI: 10.5281/zenodo.8127026. (visited on 07/08/2023).

[19] A. Gleave, P. Freire, S. Wang, and S. Toyer, *seals: Suite of environments for algorithms that learn specifications*, https://github.com/HumanCompatibleAI/seals, 2020.

[20] M. Chevalier-Boisvert, B. Dai, M. Towers, *et al.*, "Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks", *CoRR*, vol. abs/2306.13831, 2023.

[21] G. R. Ghosal, M. Zurek, D. S. Brown, and A. D. Dragan, *The effect of modeling human rationality level on learning rewards from multiple feedback types*, 2023. arXiv: 2208.10687 [cs.LG].

[22] J. Kreutzer, J. Uyheng, and S. Riezler, *Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning*, 2018. arXiv: 1805.10627 [cs.CL].

[23] J. Scheurer, J. A. Campos, T. Korbak, *et al.*, *Training language models with language feedback at scale*, 2024. arXiv: 2303.16755 [cs.CL].