LAMPO: Language and Preference Conditioned Reinforcement Learning for Controllable Speed Safety Tradeoffs

1st Aryan Ramachandra
Dept. of Computer Science and Engineering
University of California, Riverside
Riverside, USA
arama081@ucr.edu

2nd Demetreous Stillman
Dept. of Computer Science and Engineering
University of California, Riverside
Riverside, USA
dstill005@ucr.edu

***Abstract*—Standard deep reinforcement learning agents typically learn a single fixed behavior that reflects a particular weighting of task objectives, such as speed versus safety. Real world tasks instead require continuous control over these tradeoffs and often need to respect human preferences that are only partially captured by the environment reward. This paper presents LAMPO, a single policy that is conditioned on a scalar tradeoff parameter and a natural language style prompt. LAMPO is first trained with Proximal Policy Optimization (PPO) on three Gymnasium environments and then refined with reinforcement learning from human feedback (RLHF) using pairwise trajectory comparisons. We show that LAMPO covers a broad spectrum of behaviors across CartPole v1, LunarLander v3, and MountainCar v0, and that RLHF reshapes the speed safety Pareto frontier according to human judgments. RLHF yields particularly strong gains on LunarLander, where combined human aligned reward improves by more than 30 percent while environment return increases by more than 280 percent at evaluation settings.**

***Index Terms*—reinforcement learning, human feedback, multi objective optimization, natural language conditioning, safety**

I. Introduction

Deep reinforcement learning (RL) has achieved strong performance on many control tasks, yet most agents optimize a single scalar reward and therefore learn only one operating point on the speed safety spectrum. In safety critical settings, stakeholders may want to dynamically choose between aggressive, high speed behavior and conservative, risk averse behavior depending on context. Training and maintaining separate agents for each setting is inefficient and makes it difficult to interpolate between behaviors.

Human preferences further complicate this picture. Environment rewards often encode coarse task success signals but fail to capture stylistic aspects such as smoothness, comfort, or perceived risk. For example, a landing that technically maximizes LunarLander score may look dangerously hard to a human observer. Reinforcement learning from human feedback (RLHF) addresses this gap by learning a reward model from human trajectory comparisons and fine tuning the policy to align with human choices.

This work investigates how to combine multi objective RL with RLHF in a single agent. We introduce LAMPO (Language and Preference Conditioned RL agent), which conditions its policy on both a scalar tradeoff parameter and a natural language prompt. LAMPO aims to represent an entire continuum of behaviors within one policy while remaining adaptable to human feedback.

A. Contributions

The main contributions of this project are:

- A unified policy architecture that conditions on state, a continuous tradeoff parameter, and text embeddings to represent a spectrum of behaviors.
- A multi objective reward design that explicitly separates speed and safety and combines them through a controllable mixing coefficient.
- A human preference dataset collected across CartPole v1, LunarLander v3, and MountainCar v0, together with analyses of environment specific and annotator specific biases.
- An RLHF pipeline that trains a reward model from pairwise comparisons and fine tunes LAMPO, along with an evaluation of its impact on Pareto frontiers and environment returns.

II. Background and Related Work

A. Reinforcement Learning and PPO

We consider standard episodic RL with states s, actions a, transition dynamics, and scalar rewards. Policy gradient methods optimize a parameterized policy $\pi_\theta(a \mid s)$ by following gradients of expected return. Proximal Policy Optimization (PPO) stabilizes updates by clipping policy ratio terms in the objective [1]. PPO has become a widely used baseline in continuous control and in RLHF pipelines.

B. Multi Objective RL

Many tasks naturally decompose into multiple reward components such as task completion, energy usage, and safety.

Multi objective RL introduces a vector reward and seeks policies that trade off these objectives according to a preference vector. One approach is to condition the policy on a weight vector and train it across a distribution of tradeoffs so that it can generalize at test time. Our work adopts this approach with a one dimensional tradeoff parameter λ that interpolates between speed and safety.

C. Reinforcement Learning from Human Feedback

RLHF typically proceeds in two stages [2]. First, human annotators provide pairwise preferences over short trajectory segments. A reward model is trained to predict these preferences. Second, the policy is fine tuned with PPO using a combined reward consisting of environment reward and the learned preference reward, often with a KL penalty to keep the updated policy close to a reference policy. LAMPO follows this template but operates on policies that are already conditioned on a tradeoff parameter and natural language prompts.

III. LAMPO Method

A. Policy Architecture

LAMPO extends a standard PPO actor critic network so that the policy depends not only on the environment state s but also on a scalar tradeoff parameter $\lambda \in [0, 1]$ and a text embedding z derived from a natural language prompt. The inputs (s, λ, z) are concatenated and passed through shared fully connected layers, followed by separate heads for the policy logits and value estimate. This design encourages the network to learn how the same state should be acted upon differently as λ and z vary.

We use three base prompts to capture distinct behavioral styles: *be cautious*, *balanced*, and *go fast*. Each prompt is mapped to a fixed embedding vector using a text encoder. During training, prompts are sampled from this set and occasionally mixed or rephrased, encouraging some generalization to unseen natural language instructions.

B. Multi Objective Reward Design

We decompose the environment reward into a speed component r_{speed} and a safety component r_{safety}. The exact definitions are environment specific but follow the same principles.

For CartPole, the speed reward encourages progress and quick episode completion while the safety reward penalizes large pole angles and cart positions away from the center. For LunarLander, the speed reward rewards fast descents and touchdown, while the safety reward penalizes hard landings, excessive tilt, and crashes. For MountainCar, the speed reward captures momentum and rapid hill climbing, whereas the safety reward discourages unnecessary oscillations and large accelerations.

The combined reward for a given tradeoff parameter λ is

$$R_\lambda = \lambda\, r_{\text{speed}} + (1 - \lambda)\, r_{\text{safety}}. \tag{1}$$

Small values of λ yield conservative behavior that prioritizes safety, while larger values favor speed.

C. Training Pipeline

The pre RLHF training pipeline operates as follows. At the start of each episode we sample $\lambda \sim \mathcal{U}(0, 1)$ and a text prompt embedding z. The LAMPO policy then interacts with the environment for one episode conditioned on (λ, z). At each step we compute both r_{speed} and r_{safety} and combine them using (1). Trajectories from many episodes with diverse λ values and prompts are collected into a replay buffer and used to perform PPO updates.

Evaluation uses three representative tradeoff settings: $\lambda = 0$ (fully safety oriented), $\lambda = 0.5$ (balanced), and $\lambda = 1.0$ (fully speed oriented). This allows us to visualize a Pareto frontier in safety speed space and to compute combined reward as a function of λ.

IV. Experimental Setup

LAMPO is trained and evaluated on three Gymnasium control environments: CartPole v1, LunarLander v3, and MountainCar v0. These environments provide increasing difficulty and different inherent speed safety tensions.

Training hyperparameters are shared across tasks where possible. Each environment is trained for several million time steps with mini batch PPO updates. Episodes are capped at 500 steps during evaluation. We use a fixed set of prompts across environments and treat the prompt space as stylistic control rather than a separate task.

Figure 1 illustrates an example Pareto frontier of safety versus speed reward for CartPole, together with combined reward as a function of λ. Similar plots for LunarLander and MountainCar are given in Figures 2 and 3.

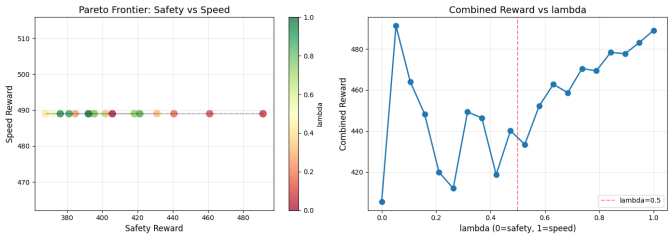

Fig. 1. CartPole v1: speed safety Pareto frontier and combined reward versus λ before RLHF.

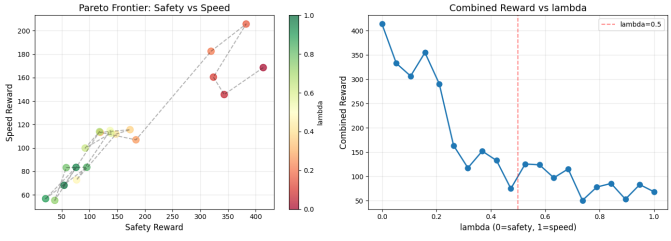

Fig. 2. LunarLander v3: speed safety Pareto frontier and combined reward versus λ before RLHF.

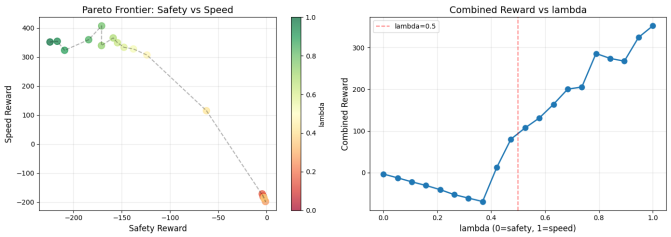

Fig. 3. MountainCar v0: speed safety Pareto frontier and combined reward versus λ before RLHF.

V. Pre RLHF Results

A. CartPole v1

CartPole serves as a sanity check for the LAMPO architecture. During training the total reward converges to approximately 372.6, with average speed reward of 482.8 and safety reward of 334.2. At evaluation, all three key tradeoff settings solve the task reliably: $\lambda = 0$ yields average return 488.1, $\lambda = 0.5$ yields 447.9, and $\lambda = 1.0$ yields 489.0. The Pareto frontier is nearly flat, indicating that CartPole admits policies that are both fast and safe over a broad range of λ values.

B. LunarLander v3

LunarLander exhibits a more pronounced tradeoff between speed and safety. The training snapshot shows total reward around 230.5, with speed reward 173.2 and safety reward 239.8. At evaluation, the best performance occurs at $\lambda = 0$, which achieves an average return of 476.2. Intermediate and high λ values lead to lower returns (126.7 at $\lambda = 0.5$ and 84.3 at $\lambda = 1.0$), reflecting the difficulty of landing quickly without sacrificing stability. The Pareto frontier in Figure 2 traces a clear curve between very safe but slow landings and riskier high speed trajectories.

C. MountainCar v0

MountainCar has sparse rewards and typically requires building momentum through aggressive back and forth oscillations. At low λ values the agent fails to climb the hill reliably. As λ increases toward 1.0, the policy learns to pump energy into the system and eventually reaches the goal, achieving returns near 468 at the most aggressive settings. The frontier in Figure 3 therefore lies in a region where safety scores are relatively low but necessary for task success.

VI. Human Preference Dataset and Analysis

To align LAMPO with human expectations, we collect pairwise preferences over trajectory segments. Each sample contains two trajectories A and B with associated λ values, prompts, episode lengths, and returns. Annotators choose which trajectory they prefer, and optionally which one appears safer or faster.

Across all environments, annotators prefer trajectory B 45.5 percent of the time, trajectory A 36.4 percent of the time, and mark ties in 18.2 percent of comparisons. Environment specific analyses reveal distinct patterns. Figures 4–6 summarize preference distributions, return differences, and preference by λ range.

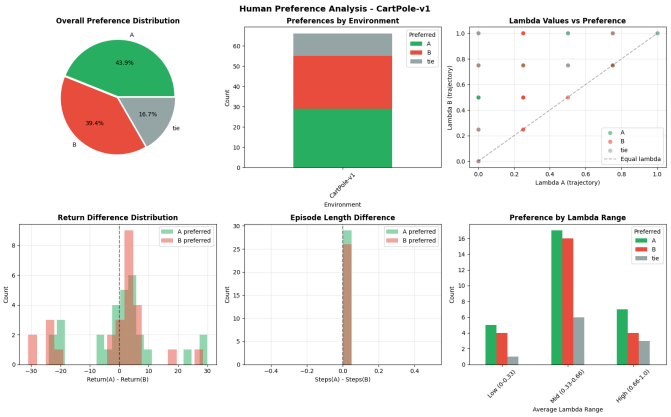

Fig. 4. Human preference analysis for CartPole v1.

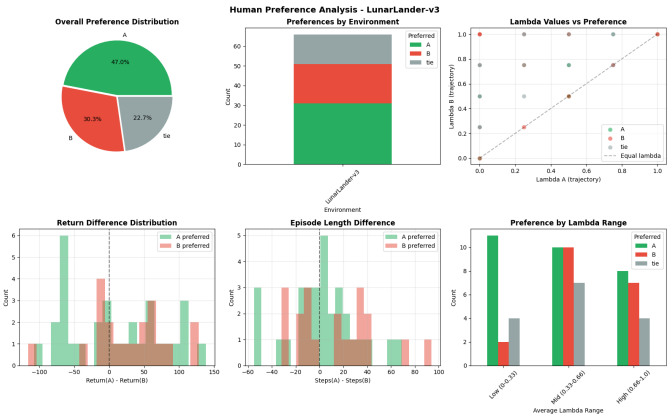

Fig. 5. Human preference analysis for LunarLander v3.

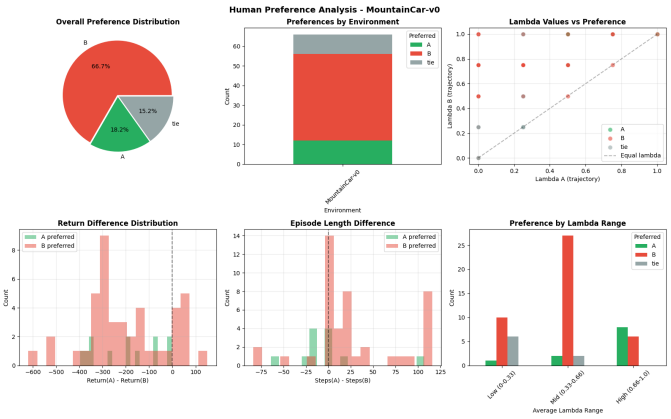

Fig. 6. Human preference analysis for MountainCar v0.

Figure 7 provides a complementary summary across all environments and annotators. The top row shows how often the preferred trajectory is safer or faster overall and how this tradeoff varies by environment. The bottom row breaks

preferences down by annotator and by the lambda values of preferred safer and preferred faster trajectories, highlighting that lower lambdas are more likely to be chosen when safety dominates while higher lambdas are favored when speed is more important.

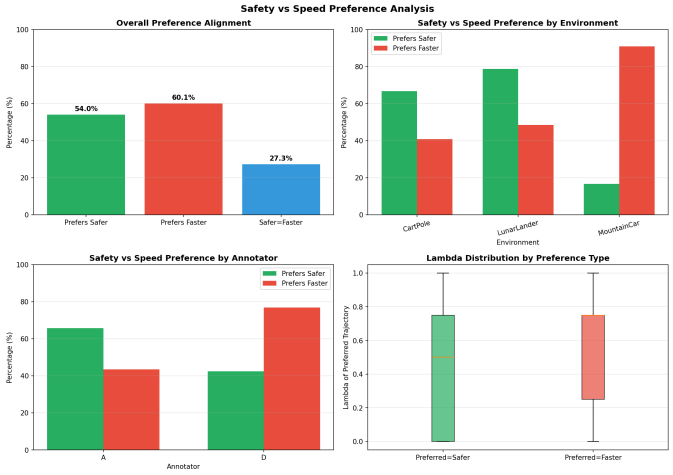

Fig. 7. Safety versus speed preference analysis across environments and annotators.

For CartPole, most trajectories succeed, so preferences are largely driven by stylistic differences such as oscillation smoothness and how close the pole comes to failure. LunarLander preferences favor smooth and controlled descents with gentle touchdowns and penalize crashes or visibly unstable landings. MountainCar preferences are more diverse; while high reward trajectories are necessarily aggressive, some annotators dislike extreme oscillations and favor slightly slower but more controlled climbs.

A. Annotator Behavior

Two primary annotators, A and D, differ in their attitudes toward risk. Annotator A tends to prefer safer trajectories and records more ties, whereas annotator D more often favors faster and more aggressive behaviors. Overall, the preferred trajectory is safer in only 54 percent of comparisons and faster in 60.1 percent, and both properties align in only about 27.3 percent of cases. Figure 8 visualizes the distribution of preferences and associated λ values.

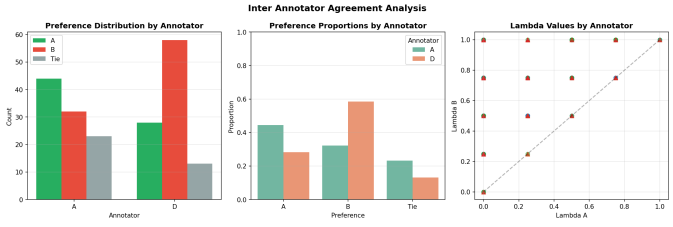

Fig. 8. Inter annotator agreement and preference patterns across λ values.

These differences motivate the use of a learned reward model rather than hard coding a fixed combination of environment rewards.

VII. Reward Model and RLHF Pipeline

A. Reward Model Training and Evaluation

We train a reward model r_ϕ to predict the probability that trajectory A is preferred over B given features derived from both trajectories. The model is optimized with a cross entropy loss on the preference labels. At evaluation time, the scalar reward assigned to a trajectory is the log odds of being preferred.

Figure 9 summarizes the reward model. The prediction histogram shows a broad distribution of preference probabilities with a decision boundary at 0.5. The calibration plot reveals that predicted probabilities track empirical preference frequencies reasonably well. The confusion matrix indicates that the model correctly classifies a majority of pairs, and per environment accuracy is substantially above the random 50 percent baseline, reaching more than 80 percent on MountainCar.

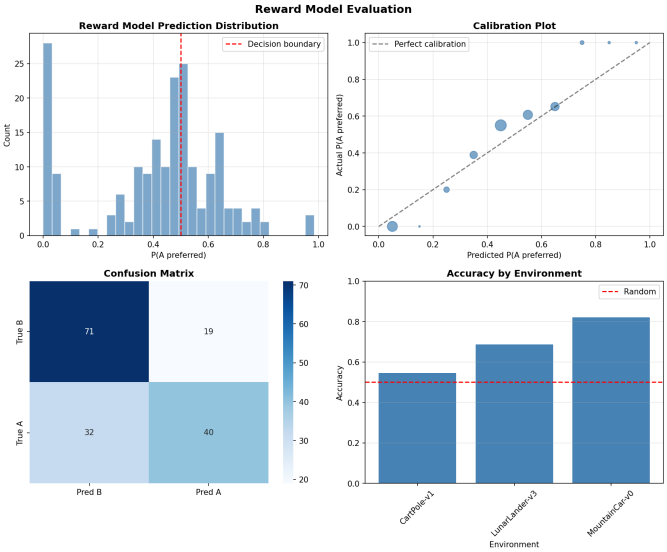

Fig. 9. Reward model evaluation: prediction distribution, calibration, confusion matrix, and accuracy by environment.

B. RLHF Fine Tuning

After pretraining LAMPO purely on environment rewards, we fine tune the policy with PPO using a combined reward

$$R_{\text{RLHF}} = R_\lambda + \alpha\, r_\phi + \beta\, R_{\text{KL}}, \tag{2}$$

where R_λ is the original combined reward from (1), r_ϕ is the learned human preference reward, and R_{KL} is a KL penalty that keeps the updated policy close to the pre RLHF policy. The weights α and β control the strength of alignment and regularization.

Fine tuning is performed for several hundred episodes per environment. We then re evaluate the agent at the same three representative λ values and compare speed reward, safety reward, combined reward, and environment reward before and after RLHF.

VIII. Post RLHF Results

A. CartPole v1

CartPole is already essentially solved before RLHF, leaving limited room for improvement. After RLHF, the speed reward at $\lambda = 1.0$ remains unchanged, while the safety reward at $\lambda = 0$ decreases by about 105.9 percent due to rescaling by the learned reward. Nevertheless, combined reward averaged over λ increases by 20.5 percent and environment reward increases by 8.7 percent. Qualitatively, trajectories remain successful but display slightly different balancing styles. Because nearly all trajectories succeed, human labels are noisy and RLHF mainly produces small stylistic adjustments.

B. LunarLander v3

LunarLander benefits most from RLHF. Across evaluation settings, speed reward increases by 31.6 percent while safety reward decreases by 24.0 percent, but combined human aligned reward improves by 31.7 percent. Most notably, average environment reward increases by approximately 281.1 percent. The updated Pareto frontier shifts upward and to the right, indicating that RLHF discovers policies that both land more reliably and achieve higher scores. Human preference labels strongly favor smooth, well controlled landings with fewer crashes, and the reward model successfully shapes the policy toward these strategies.

C. MountainCar v0

For MountainCar, RLHF produces safer and smoother behavior but slightly reduces task performance. Speed reward decreases by 31.7 percent while safety reward increases by 97.3 percent. Combined reward gains 33.6 percent on average, but environment reward drops by 7.6 percent. The Pareto frontier becomes more compact, reflecting a narrowing of behaviors toward trajectories that look less extreme to humans. High scoring solutions require aggressive oscillations that some annotators perceive as risky, so the preference reward and environment reward do not perfectly align.

D. Qualitative Before After Comparison

To further understand RLHF effects, we record short videos for each environment at $\lambda = 0.5$ with the *balanced* prompt, for up to 500 steps. For CartPole, total return changes from 318.6 to 281.5 but both policies maintain the pole for almost the full episode. For LunarLander, total return improves significantly from 76.7 to 211.1, with the post RLHF agent landing more smoothly and crashing less often. For MountainCar, return increases from 79.8 to 140.8, and the post RLHF agent reaches the goal with more controlled oscillations.

IX. Discussion

LAMPO demonstrates that a single conditioned policy can approximate an entire family of behaviors that trade off speed and safety across multiple environments. The scalar tradeoff parameter provides a simple interface for users to adjust behavior at runtime, while natural language prompts add stylistic control.

The human preference analysis highlights that speed and safety preferences vary across environments and annotators. RLHF exposes and partially resolves conflicts between environment reward and human judgments. On LunarLander, alignment with human preferences also improves environment performance, suggesting that environment reward was under specified. On MountainCar, some high reward behaviors violate human notions of safety, and RLHF prioritizes these human preferences even at a slight cost in raw return.

A. Limitations and Future Work

This project has several limitations. First, the preference dataset is modest in size and involves only two annotators, which may not capture the full diversity of human attitudes toward risk. Second, the reward model operates on trajectory level aggregates rather than raw sequences, which may overlook important temporal structure. Third, we use a single scalar tradeoff parameter; real applications may require higher dimensional preference vectors.

Future work includes collecting more diverse human feedback, exploring richer text conditioning, and extending LAMPO to more complex environments. Another direction is to learn a fully parametric Pareto front that allows users to specify arbitrary combinations of criteria beyond speed and safety.

Conclusion

We introduced LAMPO, a language and preference conditioned RL agent that unifies multi objective control and RLHF in a single policy. LAMPO enables continuous adjustment of speed safety tradeoffs through a scalar parameter and natural language prompts and learns from human trajectory comparisons. Experiments on CartPole v1, LunarLander v3, and MountainCar v0 show that LAMPO recovers a broad spectrum of behaviors and that RLHF can substantially reshape Pareto frontiers. On LunarLander in particular, RLHF yields large gains in both human aligned reward and environment return. These results suggest that combining conditioned policies with preference based fine tuning is a promising approach for building controllable and human aligned agents.

Acknowledgment

We sincerely thank Prof. Ioannis Karamouzas for his valuable feedback and guidance, as well as the teaching assistant for their support.

References

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[2] P. Christiano, J. Leike, T. Brown, et al., "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, 2017.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

Supplementary Materials

Google Colab link

Presentation Slides link