Event-Driven Reinforcement Learning for Pluralistic Alignment

Soyoung Yun[1] **Hayoung Oh**[1]

[1]Sungkyunkwan University

Abstract

Recent AI alignment research increasingly emphasizes accounting for conflicting and diverse human values rather than collapsing them into averaged preferences. However, existing reinforcement learning-based alignment approaches are typically optimized for single scalar rewards, making it difficult to address learning instability arising from heterogeneous preferences. This work proposes an event-driven reinforcement learning framework as a step toward pluralistic alignment by using discrepancy-induced learning dynamics as intervention signals rather than treating them only as noise. Under a multi-reward set $\mathbf{R} = \{R_1, \ldots, R_k\}$, spikes and collapses of episode returns are treated as candidate observable signals of latent optimization conflict, which may arise from heterogeneous reward components but can also reflect partial observability and ordinary RL instability. Under this view, spike events selectively reinforce trajectories with positive advantages, while collapse events mitigate policy fluctuations via KL regularization of the EMA anchor. In five-seed pilot experiments on CartPole-v1 with partial observations, the proposed FULL variant demonstrated superior sample efficiency compared to the control group, with an AUC of 256.85 and a solve rate of 80%, achieving the best performance with 1–4 additional event-driven updates. We further provide a reproducible protocol, theoretical hypotheses, MountainCar-v0 ablations, and an explicit experimental design for future multi-reward conflict analysis.

1 Introduction

As AI systems are deployed in broader social contexts, alignment becomes increasingly difficult to reduce to a single average correct answer. Recent work on pluralistic alignment highlights the need for empirical models that capture diverse human values and perspectives, often discussed under notions such as Overton pluralism, steerable pluralism, and distributional pluralism Sorensen et al. (2024).

From a reinforcement learning perspective, pluralistic alignment naturally leads to a multi-objective setting. Flattening multiple values or stakeholders into a single scalar reward risks obscuring violations of specific values and erasing minority preferences. Recent work advocating multi-objective reinforcement learning (MORL) for alignment suggests that trade-off policies should be modeled explicitly through vector rewards and preference conditioning Vamplew et al. (2024).

Research on personalized and pluralistic reward modeling notes that existing preference learning methods implicitly assume homogeneous preferences, motivating the need for modular and personalized reward representations Chen et al. (2025). Similarly, studies on annotator mismatch show that disagreement is not merely noise but can arise from differences in underlying preference structures, such as task ambiguity, response styles, and rejection tendencies Zhang et al. (2025).

A key challenge in this setting is the high variability of reinforcement learning dynamics. Conclusions drawn from a small number of seeds often suffer from substantial statistical uncertainty, and reliance on point estimates can lead to misinterpretation of performance differences. This issue is further amplified in partially observable environments. In POMDPs, perceptual aliasing can cause distinct underlying states to produce similar observations, increasing the need for memory and recurrent policies. Prior work, including the DRQN family and recurrent natural policy gradient methods, addresses this issue both theoretically and empirically Spaan (2012); Hausknecht and Stone (2015).

In this work, we take a different view on instability in reinforcement learning. Rather than treating spikes and collapses in return trajectories only as noise, we treat them as candidate observable signals of latent optimization conflict. In the present pilot experiments, these events are used primarily as stabilization signals; their direct attribution to pluralistic preference conflict remains an empirical question for future multi-reward evaluation. From this perspective, we propose an event-driven reinforcement learning framework that applies differentiated update rules conditioned on such events.

Our main contributions are summarized as follows:

Pluralistic Alignment Workshop @ ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

- We formulate pluralistic alignment as a POMDP with vector-valued rewards, providing a structured problem setting for event-driven analysis.
- We quantify spike and collapse events in learning dynamics and introduce event-conditional update rules at the equation level.
- We empirically analyze the effect of event-driven updates using partial observation CartPole-v1 experiments, isolating the roles of each component through ablations while treating the current results as pilot-level stabilization evidence rather than direct evidence of preference conflict.
- We further extend the experimental protocol to MountainCar-v0 and specify the additional multi-reward measurements needed to test whether event frequency increases under controlled preference conflict.

2 Problem Setting

We define the partially observable multi-reward decision problem as follows. The environment is given by

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \Omega, T, O, \mathbf{R}, \gamma, \rho_0), \tag{1}$$

POMDP formulation follows standard definitions Spaan (2012).

In each episode e, we sample a preference profile w_e, where w_e is a k-simplex. The scalarized reward is defined as

$$r_t(w_e) = w_e^\top \mathbf{R}(s_t, a_t). \tag{2}$$

Events are defined relative to an exponential moving average (EMA) baseline of episode returns. Let G_e be the cumulative return of episode e, and update the baseline to

$$b_e = \beta b_{e-1} + (1-\beta) G_e. \tag{3}$$

We define collapse and spike intensities as normalized deviations from the EMA baseline, capturing the magnitude of downward and upward changes in episodic returns relative to the current learning state,

$$c_e = \frac{\max(0, b_{e-1} - G_e)}{|b_{e-1}| + 1 + \epsilon}, \quad s_e = \frac{\max(0, G_e - b_{e-1})}{|b_{e-1}| + 1 + \epsilon}. \tag{4}$$

which serve as core signals for triggering event-conditioned updates and adapting the learning dynamics to regime changes.

Event indicators are defined as

$$\text{Collapse}_e = \mathbf{1}[G_e < b_{e-1} \wedge c_e \geq \tau_c], \tag{5}$$

$$\text{Spike}_e = \mathbf{1}[G_e > b_{e-1} \wedge s_e \geq \tau_s]. \tag{6}$$

enabling the detection of significant deviations while filtering out minor fluctuations, thereby avoiding unnecessary reactions to noise.

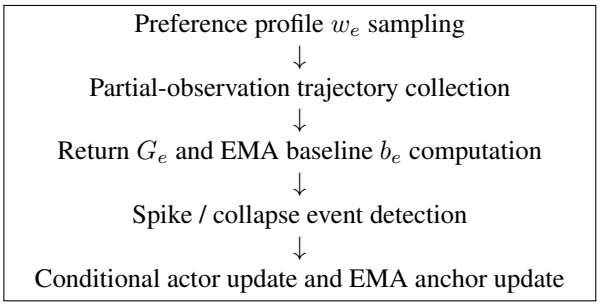

Figure 1: Overview of the event-driven reinforcement learning framework.

The standard reward for CartPole-v1 is +1 per step, and the reward threshold is 475. MountainCar-v0 receives -1 per step until the target point is reached, and the episode terminates when the position reaches 0.5 or the 200-step time limit is reached. The Gymnasium API FAQ specifies the typical reward threshold for MountainCar-v0 as -110 Farama Foundation (2024); Towers et al. (2024). Therefore, the solve criterion is set to $\text{avg}_{100} \geq 475$ for CartPole-v1 and $\text{avg}_{100} \geq -110$ for MountainCar-v0.

Table 1: Notation summary.

Symbol	Description
$\mathcal{M}$	POMDP-based multi-reward environment
$\mathbf{R} = (R_1, \ldots, R_k)$	Reward component set
w_e	Preference profile at episode e
$r_t(w_e)$	Immediate reward scalarized by the preference profile
h_t	Partial-observation history
G_e	Episodic return
b_e	EMA baseline
c_e, s_e	Collapse / spike intensity
τ_c, τ_s	Event detection thresholds
$\bar{\pi}$	EMA anchor policy
m_e	Number of extra actor updates after an event

3 Proposed Method

Policy learning is based on a recurrent actor-critic structure. The implementation uses 1-layer LSTMs for both the actor and critic in the CartPole-v1 partial observation setting, with hidden sizes of 256 for both. In the CartPole-v1 pilot, observations were restricted to $[x_t, \theta_t]$ to remove the velocity component, creating a partial observation setting that requires memory.

The base actor loss is defined as the advantage-weighted

policy gradient and entropy regularization.

$$\mathcal{L}_{\text{base}}(\theta) = -\mathbb{E}_t\big[\log \pi_\theta(a_t \mid h_t, w_e)\hat{A}_t\big] \tag{7}$$

$$- \lambda_H \mathbb{E}_t\big[\mathcal{H}(\pi_\theta(\cdot \mid h_t, w_e))\big]. \tag{8}$$

When collapse occurs, KL constraints are applied to the anchor policy, with the strength of regularization scaled according to the collapse intensity, preventing abrupt policy degradation and excessive drift.

$$\mathcal{L}_{\text{collapse}}(\theta) = \lambda_e^{\text{KL}} \cdot \mathbb{E}_t\Big[D_{\text{KL}}\big(\pi_\theta(\cdot \mid h_t, w_e) \,\|\, \bar{\pi}(\cdot \mid h_t, w_e)\big)\Big] \tag{9}$$

When a spike occurs, a self-imitation term reinforces positive-advantage trajectories, placing greater emphasis on high-return episodes and encouraging efficient exploitation. Unlike the base policy-gradient loss, this term is clipped at zero because it is intended to imitate only trajectories whose estimated advantages are positive; negative-advantage samples are already handled by the base actor update rather than being reinforced by self-imitation.

$$\mathcal{L}_{\text{spike}}(\theta) = \lambda_e^{\text{SIL}} \cdot \Big(- \mathbb{E}_t[\log \pi_\theta(a_t \mid h_t, w_e) \max(\hat{A}_t, 0)]\Big) \tag{10}$$

The total actor loss is defined in an event-gated manner, allowing different update components to be activated selectively depending on the observed learning regime.

$$\mathcal{L}_\pi(\theta) = \mathcal{L}_{\text{base}} + \mathbf{1}[\text{Collapse}_e]\mathcal{L}_{\text{collapse}} + \mathbf{1}[\text{Spike}_e]\mathcal{L}_{\text{spike}} \tag{11}$$

Each of the collapse and spike can trigger an additional actor update step, allowing the optimization process to allocate more updates to critical learning phases and respond more effectively to significant transitions; in the experiments, the number of additional per-event updates was swept. The anchor is maintained as an EMA anchor instead of the "best snapshot". This is closely related to trust-region style stabilization mechanisms Schulman et al. (2015).

Implementation summary. At each episode, the agent samples a preference profile, rolls out a partial-observation trajectory, computes the episodic return and the EMA baseline, detects collapse or spike events using the normalized deviations in Eqs. 3–11, and applies the corresponding gated actor update. When an event is detected, the actor receives at most U_{cap} additional updates, after which the critic, EMA anchor, and EMA return baseline are updated.

4 Computational Complexity

As implemented, actor computation increases by at most 7× compared to the default, even in the worst case. The event determination itself is $O(1)$, and the memory complexity is due to trajectory storage.

For this reason, the actual bottleneck of event-driven methods is not "detection" but the "number of additional actor-steps," and an appropriate range for U must be found through the sweep results.

Table 2: Hyperparameters used in the experiments.

Item	Value
Discount rate γ	0.99
Actor hidden dim	256
Critic hidden dim	256
LSTM layers	1
LSTM dropout	0.1
Actor LR	2×10^{-4}
Critic LR	2×10^{-4}
Gradient clip norm	0.5
Entropy coefficient	0.01
Baseline EMA β	0.90
KL base coefficient	0.5
KL max	5.0
SIL base coefficient	0.5
SIL max	5.0
Event deadzone $\tau_c = \tau_s$	0.04
Anchor EMA α	0.995
Default extra updates	2
Sweep range	$\{0, 1, 2, 4, 6\}$
Number of episodes	5000
CartPole-v1 solve criterion	avg100 ≥ 475
MountainCar-v0 solve criterion	avg100 ≥ -110

4.1 Environment and Observation

CartPole-v1 is a classic control environment with a reward of +1 per step and a threshold of 475. Partial observations are created in this environment by observing only $[x_t, \theta_t]$ and removing the velocity component. MountainCar-v0 has 3 discrete actions, a reward of -1 per step, a goal position of 0.5, and a time limit of 200; for the final version, we use position-only observations or velocity flicker observations for consistent POMDP evaluation. The environment implementation uses the Gymnasium API in user code, but the environment semantics are practically compatible with those of Gymnasium API's CartPole-v1 and MountainCar-v0.

4.2 Network and Optimization

The key hyperparameters found in the user implementation are summarized in Table 2.

4.3 Comparison Group

The headline ablation variants are summarized in Table 3.

Table 3: Ablation variants.

Variant	Description
FULL	KL anchor + spike SIL + event extra updates
NoSpike	Remove spike-based SIL and spike-triggered updates
NoCollapse	Remove collapse KL and collapse-triggered extra updates
NoEventUpdate	Detect events but do not perform extra updates
Vanilla	Remove all KL/SIL/event extra updates

4.4 Reward Variants Design

To simulate pluralistic alignment using only standard Gym, we define three reward components for each environment and configure three or more preference profiles with scalar weights. The MountainCar-v0 component design is directly inspired by the decomposition presented by MO-Gymnasium Alegre et al. (2022). MO-Gymnasium provides a standard API that returns vectorized rewards and examples of MountainCar-v0 series multi-reward.

4.5 Definition and Interpretation of Environmental Compensation Components

The reward components used in our experimental design are summarized in Table 4. The preference profiles are summarized in Table 5. These profiles are intended to support two follow-up evaluations: independent runs for each reward profile and a single preference-conditioned recurrent policy in which w is concatenated to the policy input. The current camera-ready version reports the single-reward pilot results and treats the multi-reward setting as a specified protocol for directly testing the pluralistic-conflict hypothesis.

Table 4: Reward components for the multi-reward protocol.

Env.	Component	Interpretation
CartPole	R_1^{cp}: $+1$ per step	Survival / achievement
CartPole	R_2^{cp}: $-\lvert x_t\rvert/2.4 - \lvert\theta_t\rvert/0.2095$	Safety margin
CartPole	R_3^{cp}: $-\mathbf{1}[a_t \neq a_{t-1}]$	Smooth control
MountainCar	R_1^{mc}: -1 per step	Time minimization
MountainCar	R_2^{mc}: $\lvert v_t\rvert$ or $x_{t+1} - x_t$	Propulsion
MountainCar	R_3^{mc}: $-\mathbf{1}[a_t \in \{0, 2\}]$	Energy saving

Table 5: Preference profiles for the multi-reward protocol.

Env.	Profile	Weight w
CartPole	Achievement-first	$[0.70, 0.25, 0.05]$
CartPole	Safety-first	$[0.45, 0.45, 0.10]$
CartPole	Smooth-control	$[0.40, 0.20, 0.40]$
MountainCar	Fast-goal	$[0.75, 0.20, 0.05]$
MountainCar	Momentum-seeking	$[0.45, 0.45, 0.10]$
MountainCar	Energy-saving	$[0.55, 0.10, 0.35]$

4.6 Metrics and Statistical Testing

The evaluation metrics used in this manuscript are summarized in Table 6.

Table 6: Metrics used in the experiments.

Metric	Definition
AUC mean	Overall average of 5000 episodes return
last100 mean	Average of the last 100 episodes
last100 std	Standard deviation of the last 100 episodes
first solve episode	Episode at which avg_{100} first exceeds the solve threshold
solve rate	Proportion of solved episodes among seeds
collapse count	Number of episodes judged as collapse
spike count	Number of episodes judged as spike
stability_last100	$1 - \mathrm{std}_{100}/(\lvert\mathrm{mean}_{100}\rvert + \epsilon)$
worst-profile return	Minimum value among profile means in pluralistic experiments
profile variance	Average performance variance between profiles

For statistical testing, the Mann–Whitney U two-sided test and Cliff's delta are recommended for seed-level AUC. It is advisable to present the bootstrap 95% CI along with the mean, and if possible, the IQM and performance profile. This aligns with recent RL evaluation recommendations Agarwal et al. (2021).

5 Results

5.1 CartPole-v1 Partial Observation Ablation

Table 7 shows the ablation results. FULL recorded the highest AUC and also had the highest solve rate at 80%. Among solved runs, first solve was much faster than NoSpike, which also showed an 80% solve rate. The main effect of spike processing appears to lie in sample efficiency and accelerating early convergence rather than maximizing final performance.

Conversely, while NoCollapse appears to have the highest stability metric at 0.92, its actual performance is extremely low (average of the last 100 episodes: 9.34). This highlights a core failure mode. Low variance may not indicate good stability, but rather a policy that fails stably. Therefore, stability should be interpreted only as a supplementary metric and must be read in conjunction with AUC, solve rate, and worst-profile return.

NoEventUpd is an interesting comparison group. Although the average for the last 100 episodes is actually the highest, the solve rate is 0%. This indicates that event detection alone is insufficient for consistent convergence.

Table 7: CartPole-v1 partial observation ablation results.

Variant	AUC mean	last100 mean	mean last100 std	collapse count	spike count	stability_last100	solve rate	mean first-solve among solved
FULL	256.85	318.61	85.02	1944.2	2337.6	0.684	4/5	2711.5
NoSpike	215.81	388.30	58.74	2075.2	2146.4	0.808	4/5	3748.75
NoEventUpd	175.49	400.94	152.91	2484.8	2313.2	0.614	0/5	–
NoCollapse	21.24	9.34	0.75	1984.4	1957.2	0.920	0/5	–
Vanilla	68.81	198.96	120.74	2808.2	1924.2	0.261	0/5	–

Table 8: Effect of extra updates on event (FULL variant, complete sweep).

Updates	AUC	Last100	Collapse	Stability	Solve
0	175.49	400.94	2484.8	0.614	0/5
1	247.46	330.85	2054.6	0.675	2/5
2	255.39	251.78	1998.8	0.509	4/5
4	256.85	318.61	1944.2	0.684	4/5
6	178.11	231.23	2357.0	0.544	1/5

Table 9: Exploratory statistical comparison.

Comparison	ΔAUC	p	Cliff's δ
FULL vs NoSpike	41.04	0.4206	0.36
FULL vs NoEventUpd	81.36	0.3095	0.44
FULL vs NoCollapse	235.61	0.0079	1.00
FULL vs Vanilla	188.04	0.0079	1.00

At the 100-episode level, consistent resolution is achieved only when events are used as learning-intervention variables rather than as a simple logger.

5.2 Event Extra Updates Sweep

Table 8 reports the sweep results obtained by varying extra_updates_on_event in the FULL variant.

The sweep results in Table 8 suggest that event-driven additional learning is not a monotonic effect where "more is better," but a non-linear mechanism in which an optimal level exists. When moving from 0 to 1 or 4, the AUC increases significantly and the collapse count decreases, but when increased to 6, the AUC drops sharply. This aligns with the theoretical interpretation that, while event-driven exploitation is necessary, excessive updates can lead to overshoot or overfitting.

The fact that the first solve appears fastest at 6 (2076) is misleading because the solve rate is only 1/5. Aggressive updates can therefore produce fast, lucky solves in some seeds while remaining unstable overall. We adopt 4 additional updates based on reproducibility across the full seed set rather than the fastest single solved seed.

5.3 Exploratory Statistical Comparison

Table 9 reports an exploratory comparison of the seed-level AUC between FULL and each comparison group in the CartPole-v1 main ablation. Since $n = 5$ is small, statistical significance should be interpreted only as a secondary indicator.

The comparison in Table 9 shows that the key comparison groups for FULL are NoCollapse and Vanilla, rather than NoSpike. If collapse suppression is removed, the learning process itself nearly collapses. In contrast, spike processing is closer to a secondary component and is primarily responsible for improving sample efficiency. Clearly stating this division of roles in the paper significantly strengthens the narrative of the ablation study.

5.4 MountainCar-v0 Ablation

Table 10 reports the MountainCar-v0 ablation results under the same experimental protocol.

Table 10: MountainCar-v0 ablation results.

Variant	AUC	Last100	Eval	Solve Ep
FULL	-149.50	-143.41	-146.00	1350
NoSpike	-158.10	-143.41	-161.00	632
NoEventUpd	-151.96	-155.77	-171.23	1386
Vanilla	-156.53	-168.39	-171.30	1485
NoCollapse	-163.37	-175.50	-191.20	–

This is consistent with the CartPole-v1 results. FULL achieves the best performance across all metrics. NoSpike shows comparable last100 performance but weaker evaluation, indicating that spike-based updates mainly contribute to improving policy quality rather than final return. In contrast, NoCollapse exhibits severe degradation, confirming that collapse protection is critical for stabilization. Overall, event-driven conditional updates are necessary for achieving both stable and high-performing policies in MountainCar-v0.

6 Theoretical Discussion

The core finding is that collapse prevention acts as a major stabilization mechanism for event-driven RL. Experimental results show that in the case of NoCollapse, performance remained low for most seeds, and that the removal of KL-anchors failed to control policy variability. For NoSpike, learning is possible, but it is disadvantageous in terms of early convergence. We conclude that the optimal overall structure is one that reduces downside risk through collapse suppression and accelerates learning during the rise phase by utilizing spikes.

Our observations motivate, but do not yet prove, the hypothesis that greater preference conflict may lead to more pronounced event-driven variability during learning. Let $g(w)$ and $g(w')$ be the policy gradients for multiple preferences w and w', respectively. If the inner product between the two gradients is negative, an update favorable to one preference may act unfavorably to the other.

$$\langle g(w), g(w') \rangle < 0. \tag{12}$$

For example, sharp performance drops correspond to collapse events, whereas an update that captures the substructure of a specific preference well may appear as a spike that creates a rapid increase in returns. In the current experiments, this interpretation should be read as a mechanistic hypothesis rather than causal evidence of pluralistic disagreement.

The first rule reinforces only positive advantage trajectories during spikes. It targets the behavior distribution immediately prior to the event for local exploitation. In particular, when gradient clipping is applied along with a limit on the number of additional updates, a specific episode cannot dominate the overall learning process; consequently, if the number of updates increases excessively, overfitting or policy overshoot may occur.

Second, we observe that introducing a KL-anchor term during collapses mitigates excessive deviation from the existing policy in a single update. From a trust-region perspective, this suggests that small policy shifts are crucial for controlling the upper bound of performance slumps. Under the bounded advantage assumption, mitigating event-triggered collapse acts as a local safeguard that limits one-step degradation.

While the argument presented in this paper does not guarantee monotonic improvement, it offers a mechanistic interpretation of a collapse prevention mechanism. This aligns with the perspective raised by pluralistic reward and disagreement modeling that "disagreements are meaningful structures, not mere noise" Chen et al. (2025); Zhang et al. (2025). However, this interpretation must be handled cautiously in a POMDP environment, as recurrent policies can enhance expressiveness by compressing history but simultaneously entail long-term dependencies and non-stationarity. The theoretical argument presented in this work should be interpreted as a local interpretation of event-driven stabilization and selective exploitation, rather than a guarantee of global convergence. Furthermore, this approach has several limitations.

1. Since the current event definition is a lagging detection method based on episode return deviations, it is not an indicator that directly measures preference conflicts.

2. Synthetic reward profiles are more limited than real-world preference signals. Future research requires additional metrics such as profile-conditioned gradient analysis or worst-case returns.

3. Since CartPole-v1 and MountainCar-v0 are simple environments, it is appropriate to interpret these results as a pilot-level validation. Further detailed verification of the practical applicability of this model is necessary as it is extended to real-world environments, task achievement, and the introduction of benchmark environments.

7 Conclusion

We propose an event-driven reinforcement learning framework as a step toward pluralistic alignment and treat spikes and collapses as candidate observable signals of latent optimization conflict in learning dynamics. The partial observation pilot results for CartPole-v1 show that the FULL variant outperforms the control group in terms of sample efficiency and solve rate. This indicates that collapse suppression is a key stabilization mechanism and that spike processing can lead to early convergence. The sweep study further suggests that there is an optimal range for additional event-driven updates and that excessive exploitation can lead to performance degradation. Future work should directly evaluate multi-reward settings by reporting worst-profile return, profile variance, event frequency under controlled conflict intensity, and comparisons with established MORL and RL stabilization baselines. These experiments are necessary to verify whether the event dynamics observed here are specifically attributable to pluralistic preference conflict.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pages 29304–29320, 2021.

Lucas N. Alegre, Florian Felten, El-Ghazali Talbi, Grégoire Danoy, Ann Nowé, Ana L. C. Bazzan, and Bruno C. da Silva. MO-Gym: A library of multi-objective reinforcement learning environments. In *Proceedings of the 34th Benelux Conference on Artificial Intelligence*, 2022.

Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. PAL: Sample-efficient personalized reward modeling for pluralistic alignment. In *International Conference on Learning Representations*, 2025.

Farama Foundation. Gymnasium documentation. `https://gymnasium.farama.org/`, 2024.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI Fall Symposium Series*, 2015.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897. PMLR, 2015.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L. Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. Position: A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46280–46302. PMLR, 2024.

Matthijs T. J. Spaan. Partially observable markov decision processes. In *Reinforcement Learning: State-of-the-Art*, pages 387–414. Springer, 2012.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Peter Vamplew, Conor F. Hayes, Cameron Foale, Richard Dazeley, and Hadassah Harland. Multi-objective reinforcement learning: A tool for pluralistic alignment, 2024.

Michael JQ Zhang, Zhilin Wang, Jena D. Hwang, Yi Dong, Olivier Delalleau, Yejin Choi, Eunsol Choi, Xiang Ren, and Valentina Pyatkin. Diverging preferences: When do annotators disagree and do models know? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 76193–76212. PMLR, 2025.