On-Policy Reinforcement Learning From Failure via Sparse Reward Densification

Extended Abstract

Mingkang Wu
The University of Texas at San Antonio
San Antonio, TX, USA
mingkang.wu@utsa.edu

Yongcan Cao
The University of Texas at San Antonio
San Antonio, TX, USA
yongcan.cao@utsa.edu

ABSTRACT

This paper proposes a new reinforcement learning method that learns from failure under sparse reward environments. While traditional approaches rely on costly expert demonstrations to guide learning in sparse reward environments, this method uses readily available failures. The method trains a discriminator to measure the dissimilarity between the agent's behaviors and failures, generating dense rewards. The method then uses this information to guide policy learning. Experimental results show this failure-based learning approach performs competitively with existing methods.

KEYWORDS

Reinforcement Learning; Sparse Reward Environment; Failed Experience; Reward Densification

ACM Reference Format:
Mingkang Wu and Yongcan Cao. 2025. On-Policy Reinforcement Learning From Failure via Sparse Reward Densification: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

This work tackles the sparse reward problem in reinforcement learning [6, 8, 10, 11] with a novel approach using failed experiences rather than expert demonstrations. While traditional methods rely on expert demonstrations to guide learning [4, 5], which is expensive and resource-intensive [7]. Instead, this work proposes using readily available failed experiences to guide exploration through a GAN-based architecture [1] and reward densification method. The system works by training a discriminator to measure the dissimilarity between the agent's behaviors and failures, generating dense rewards from sparse ones. This approach differs from previous failure-based methods like BIRL [2, 7] which still required expert data. Experiments across multiple environments show this pure failure-based approach can match or exceed methods using expert demonstrations, while being much more practical to implement due to the easy availability of failure data. Our contributions are summarized as follows. First, we develop a GAN-based algorithm where the discriminator measures how different the agent's behaviors are from failures, while the generator generates contrasting actions. Second, we introduce a new approach to densify the sparse rewards by promoting exploration away from failed behaviors. Third, our experiments across multiple environments prove that learning from failures alone can be as effective as, or better than, using expert demonstrations.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

2 PRELIMINARY

In this paper, we consider a Markov Decision Process (MDP) defined by $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where the RL agent takes an action $a \in \mathcal{A}$, receives a reward r from $R(s, a)$, and moves to the next state s' determined by $P(s'|s, a)$. The goal is to learn a policy π to maximize the expected cumulative rewards with a discounted factor $\gamma \in [0, 1)$, which is formulated as $\pi^* = \arg\max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t\right]$, where π^* denotes the optimal policy π, r_t is the reward received at time step t and T is the time horizon.

Standard reinforcement learning struggles with exploration under sparse reward environments due to the infrequently occurred rewards. While expert demonstrations can guide exploration [12], they show expensive to obtain. Our approach instead uses easily accessible failed experiences. We introduce an augmented reward function with a discriminator that measures dissimilarity to failures, allowing the agent to maximize rewards while avoiding known failure behaviors.

3 TECHNICAL APPROACH

Our approach builds on the GAN framework [1], where the generator learns actions opposite to failures, while the discriminator differentiates between generated actions and failures. The objective function is formulated as

$$\min_\theta \max_\omega L = -\mathbb{E}_{\pi_\theta}[r_d(s,a)] - \lambda_1(\mathbb{E}_{\pi_\theta}[\log(D_\omega(s,a))] + \mathbb{E}_{\pi_f}[\log(1 - D_\omega(s,a))]) - \lambda_2 H(\pi_\theta), \quad (1)$$

where $\mathbb{E}$ denotes the expectation operator, r_d is the densified reward, D_ω measures dissimilarity from failures, and $H(\pi_\theta)$ is an entropy regularizer [13] to prevent overfitting. The weights λ_1 and λ_2 balance the GAN objective and entropy. The proposed objective function optimizes policy parameters θ and discriminator parameters ω to maximize rewards while avoiding behaviors similar to failures. In sparse reward environments where $r(s, a)$ is limited, we propose a densification technique using failed experiences and the discriminator to create a new dense reward,

$$r_d(s,a) = r'(s,a) + \lambda_3 \frac{1}{1 + e^{-A'(s,a)}}, \quad (2)$$

Environment	Empirical Return		
	TRPO	PPO	GAIL
HalfCheetah	24.03 ± 2.35	**1696.59 ± 371.39**	66.99 ± 136.04
Hopper	17.72 ± 4.26	2412.90 ± 360.03	**3570.98 ± 57.75**
Humanoid	5.56 ± 0.23	559.42 ± 35.28	651.79 ± 65.86
Walker2D	13.88 ± 2.65	3122.98 ± 469.62	3038.60 ± 396.58
HumanoidStandup	30.35 ± 7.49	**125178.96 ± 9202.7**	93213.95 ± 2985.37
Pendulum	**−177.31 ± 30.83**	−1267.22 ± 351.57	−879.35 ± 217.79
	POfD	Ours (TRPO)	Ours (PPO)
HalfCheetah	376.73 ± 292.10	1308.80 ± 124.39	1349.89 ± 111.08
Hopper	3378.17 ± 184.31	3441.69 ± 69.09	3488.19 ± 38.12
Humanoid	3017.90 ± 239.83	5578.16 ± 305.97	**5594.70 ± 299.51**
Walker2D	3365.82 ± 552.39	4838.83 ± 197.25	**4943.86 ± 72.16**
HumanoidStandup	92882.74 ± 3085.48	98273.71 ± 7067.30	94600.99 ± 4783.22
Pendulum	−959.84 ± 211.41	−443.06 ± 493.10	−1007.57 ± 560.31

Table 1: Empirical return comparison among different algorithms.

where

$$r'(s,a) = \begin{cases} r(s,a) + \lambda_1 \log(D_{\omega_i}(s,a)), & \text{if } r(s,a) \text{ is available} \\ \lambda_1 \log(D_{\omega_i}(s,a)), & \text{otherwise} \end{cases} \tag{3}$$

is the sparse densification term. D_{ω_i} measures dissimilarity from failures at timestep i, $A'(s,a)$ is the advantage function computed using $r'(s,a)$ with weight λ_3 [3]. The term $\frac{1}{1+e^{-A'(s,a)}}$ in (2) is the sigmoid-transformed [3] $A'(s,a)$ to match the discriminator's output dimension. This reward densification guides the agent away from failures while exploring promising actions. The pseudo code of the proposed approach can be found in Algorithm 1.

4 EXPERIMENTS AND RESULTS

We evaluate our method across six environments, namely, HalfCheetah, Hopper, Humanoid, Walker2D, HumanoidStandup, and Pendulum [9], using failed experiences collected from a random policy and evaluated by users with basic environment understanding.

Our method uses TRPO and PPO for generator updating, comparing against baseline TRPO, PPO, GAIL, and POfD which uses GAN structure for expert demonstration learning. We conducted 10 runs per algorithm with different random seeds. Table 1 shows that 'Ours (TRPO)' and 'Ours (PPO)' outperform GAIL and POfD in most environments, except in Hopper, where GAIL peroformances the best among all methods, and in Pendulum, where TRPO or PPO outperform others with dense rewards. The performance of the proposed approach is lower in simpler environments like Pendulum, where dense rewards provide better learning signals.

Based on the experiments, our approach has three main limitations. First, failure definition can be subjective, varying between human evaluators. This could be addressed through crowdsourcing to obtain more objective assessments. Second, the quality of failure data depends on human factors like fatigue and expertise, suggesting a need for noise-resistant learning methods. Finally, our approach is less effective in simple environments where failures provide limited information, similar to how humans learn less from failures in straightforward tasks.

Algorithm 1 Policy Optimization with Failure

Require: failed experiences, initial generator and discriminator parameters θ_0 and ω_0, discriminator's weight λ_1, reward densification parameters λ_1 and λ_2, the weight of advantage-based reward densification term λ_3, total training cycles T, learning rate α

Collect failed experiences

Initialize generator's policy π_{θ_0}, discriminator D_{ω_0}

for i=1, T **do**

Sample trajectories τ from generator

Sample trajectories τ_f from failed experiences

Update discriminator parameter $\omega_{i+1} \leftarrow \omega_i$ with the gradient

$$-(\mathbb{E}_{\tau\sim\pi_\theta}[\nabla_\omega \log(D_\omega(s,a))] + \mathbb{E}_{\tau_f\sim\pi_f}[\nabla_\omega \log(1 - D_\omega(s,a))])$$

Update the reward function

$$r_d(s,a) = r'(s,a) + \lambda_3 \frac{1}{1+e^{-A'(s,a)}},$$

where

$$r'(s,a) = \begin{cases} r(s,a) + \lambda_1 \log(D_{\omega_i}(s,a)), & \text{if } r(s,a) \text{ is available} \\ \lambda_1 \log(D_{\omega_i}(s,a)), & \text{otherwise} \end{cases}$$

Update the policy of the generator with policy gradient methods, such as TRPO and PPO

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log(\pi_\theta) r_d] - \lambda_2 H(\pi_\theta), \tag{4}$$

$$\theta_{i+1} \leftarrow \theta_i + \alpha \nabla_{\theta_i} J(\pi_{\theta_i})$$

end for

ACKNOWLEDGMENTS

This work was supported by the Army Research Office under Grants W911NF2110103 and W911NF2310363, and the Office of Naval Research under Grant N000142212474.

REFERENCES

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[2] Daniel H Grollman and Aude Billard. 2011. Donut as i do: Learning from failed demonstrations. In *International Conference on Robotics and Automation*. 3804–3809.

[3] Jun Han and Claudio Moraga. 1995. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International Workshop on Artificial Neural Networks*. 195–201.

[4] Bingyi Kang, Zequn Jie, and Jiashi Feng. 2018. Policy optimization with demonstrations. In *International Conference on Machine Learning*. 2469–2478.

[5] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Overcoming exploration in reinforcement learning with demonstrations. In *International Conference on Robotics and Automation*. 6292–6299.

[6] Warren B Powell. 2012. AI, OR and control theory: A rosetta stone for stochastic optimization. *Princeton University* (2012), 12.

[7] Kyriacos Shiarlis, Joao Messias, and Shimon Whiteson. 2016. Inverse reinforcement learning from failure. (2016).

[8] Umer Siddique, Abhinav Sinha, and Yongcan Cao. 2023. Fairness in Preference-based Reinforcement Learning. *arXiv preprint arXiv:2306.09995* (2023).

[9] Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. 2023. Gymnasium. https://doi.org/10.5281/zenodo.8127026

[10] Devin White, Mingkang Wu, Ellen Novoseller, Vernon J Lawhern, Nicholas Waytowich, and Yongcan Cao. 2024. Rating-based reinforcement learning. In *AAAI Conference on Artificial Intelligence*. 10207–10215.

[11] Mingkang Wu, Umer Siddique, Abhinav Sinha, and Yongcan Cao. 2024. Offline Reinforcement Learning with Failure Under Sparse Reward Environments. In *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*. IEEE, 1–5.

[12] Xu Xie, Changyang Li, Chi Zhang, Yixin Zhu, and Song-Chun Zhu. 2019. Learning virtual grasp with failed demonstrations via bayesian inverse reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1812–1817.

[13] Brian D Ziebart. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University.