ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

DEFENDER: DTW-Based Episode Filtering Using Demonstrations for Enhancing RL Safety

André Correia and Luís A. Alexandre *

Universidade da Beira Interior and NOVA LINCS
Covilhã - Portugal

Abstract. Deploying reinforcement learning agents in the real world can be challenging due to the risks associated with learning through trial and error. We propose a task-agnostic method that leverages small sets of safe and unsafe demonstrations to improve the safety of RL agents during learning. The method compares the current trajectory of the agent with both sets of demonstrations at every step, and filters the trajectory if it resembles the unsafe demonstrations. We perform ablation studies on different filtering strategies and investigate the impact of the number of demonstrations on performance. Our method is compatible with any stand-alone RL algorithm and can be applied to any task. We evaluate our method on three tasks from OpenAI Gym's Mujoco benchmark and two state-of-the-art RL algorithms. The results demonstrate that our method significantly reduces the crash rate of the agent while converging to, and in most cases even improving, the performance of the stand-alone agent.

1 Introduction

Reinforcement learning (RL) [1] enables agents to learn how to behave in an environment through trial and error, but safety concerns have limited RL deployments to simulations. Ensuring safety in unknown environments remains a challenge in RL, particularly in safety-critical domains such as healthcare and autonomous driving. To address these issues, safe RL studies the RL problem subject to certain constraints, with the agent aiming to maximize task reward while limiting constraint violations. However, deploying agents with complete knowledge of the environment to perform their tasks is unrealistic. Alternatively, in demonstration learning (DL) the agent learns from expert demonstrations without direct environment interaction. However, the quality of the data set plays a crucial role in the performance. Due to the difficulty of collecting a demonstration data set that covers the entire state space, pure DL policies often underperform compared to RL policies.

In this paper, we propose a novel task-agnostic algorithm that enhances existing RL algorithms by promoting safety during interactions with the environment. Our algorithm uses a small data set of good and bad demonstrations to filter unsafe actions, terminate episodes with unsafe trajectories, and encourage the agent to explore different trajectories. We conduct ablation studies to evaluate filtering strategies and demonstrate the utility of our

*This work is supported by NOVA LINCS (UIDB/04516/2020) with the financial support of 'FCT - Fundação para a Ciência e Tecnologia' and also through the research grant '2022.14197.BD'.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

method on four tasks from OpenAI's MuJoCo environment. Our contributions in this paper are: (1) enhancing RL algorithms with safety filtering, (2) performing ablation studies on filtering strategies, (3) demonstrating the utility of our method on OpenAI's MuJoCo tasks, and (4) providing the code implementation: https://github.com/meowatthemoon/DEFENDER.

2 Related Work

Reinforcement learning has gained significant attention in recent years due to its wide range of applications. However, its trial-and-error nature can pose safety risks.One approach to safe reinforcement learning is to keep the agent within a safe distribution of states. For example, [2] proposes learning a manifold that captures natural variations in the environment and uses a secondary policy to bring the agent back into the distribution of visited states. Safety can be achieved through specification of constraints. For instance, [3] proposes learning a barrier function that constrains the agent's policy to stay within a set of states that do not violate constraints. Alternatively, [4] propose a zero-sum game where a second player perturbs the transition probabilities of the agent to optimize the worst transitions to produce a more robust policy. Some methods leverage a data set of expert demonstrations. [5] proposes learning a Lyapunov function that ensures the agent's policy remains within the distribution of states of the data set. In [6], the authors use a pre-existing expert policy to filter the agent's action if it differs from the expert's. Our method uses a demonstration data set to improve safety and does not require access to task constraints or an expert.

3 Preliminaries

3.1 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning technique that enables an agent to learn to act in an unknown environment through trial-and-error interactions. RL is often formulated as a Markov Decision Process (MDP) described by the tuple $< S, A, R, P, \gamma >$ [1], consisting of states $s \in S$, actions $a \in A$, transition function $P(s_{t+1} \mid s_t, a_t)$, reward function $r_t = R(s_t, a_t)$ and discount factor γ. At each timestep t, the agent receives the state s_t, selects an action $a_t = \pi(s_t)$ based on its policy, receives a reward $r_t = R(s_t, a_t)$, and transitions to a new state s_{t+1} with probability $P(s_{t+1} \mid s_t, a_t)$. A trajectory τ is a sequence of states and actions. The goal of RL is to learn a policy π that produces trajectories τ that maximize the expected return $\mathbb{E}_\pi[R\tau]$. Standard RL algorithms optimize a policy to maximize the expected rewards disregarding any safety concerns.

3.2 Dynamic Time Warping

Dynamic Time Warping (DTW) [7] measures the similarity between two sequences of temporal data, allowing for distortions in time and variations in speed. Making it particularly useful for comparing sequences with different

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Algorithm 1 RL loop with DEFENDER enhancement

Input: Policy π; Memory β; Dynamics θ; Constant R_{task}; Alignment cost functions: $Safe, Unsafe$; Task environment Env.
while EPISODE not DONE **do**
  $a \leftarrow \pi(s)$
  $\tau \cup$ s or (s, a)
  if $Safe(\tau) < Unsafe(\tau)$ **then**
    $s', r, done = Env(a)$
  else
    $s', r, done = \theta(s, a), R_{task}, True$
  end if
  $\beta \leftarrow (s, a, r, s')$
  Optimize π and θ
end while
return π

lengths, speeds, or underlying shapes. In practice we use FastDTW [8], an approximate to DTW with reduced time and memory costs. The algorithm finds the optimal warping path which minimizes the cumulative distance between corresponding points on the two sequences. The alignment cost of the path measures the similarity between two sequences.

4 Proposed Approach

We propose DEFENDER: DTW-Based Episode Filtering Using Demonstrations for Enhancing RL Safety, a method to improve the safety of any RL algorithm during learning by leveraging limited demonstration data sets without task-specific constraints. The algorithm keeps track of the current trajectory which is either the sequence of states or state-action pairs. We performed ablation studies to evaluate the type of trajectory which results in better performance. We assume access to a demonstration data set $D_{demo} = \{\tau_i\}^N$ containing N demonstrations, where $N > 1$ is potentially a small number that would not suffice for demonstration learning. The data set must contain both safe and unsafe trajectories. Unsafe trajectories terminated due to reaching unsafe states. Safe trajectories do not have to be perfect but must avoid unsafe states.

DEFENDER measures the alignment cost of the current trajectory at every step with each demonstration from both groups using the DTW algorithm. If the trajectory aligns better with the unsafe group the episode is terminated and the agent is discouraged from re-sampling this trajectory. If the trajectory is terminated, the agent must be encouraged not to sample the same action that would have caused termination for the state. Otherwise, the agent will try again. A negative reward is assigned to discourage sampling the same action again. We use the lowest reward from the reward function of the respective task.

RL algorithms use the next state to perform temporal learning. Since the

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Table 1: Performance, safety and computation time of SAC and TD3 agents enhanced with our algorithm using different filters for state trajectories.

Algorithm		Hopper			InvertedDoublePendulum			Walker2d		
		Acc Reward	% Crash	% Time	Acc Reward	% Crash	% Time	Acc Reward	% Crash	% Time
SAC		3474±85.0	64±6.0	100	**9359±0.2**	14±2.7	100	3326±254.6	40±4.0	100
MinDemoW5	MinDemoW10	50±4.0	3±0.2	143	9357±1.5	**1±0.2**	144	4149±106.3	**3±0.6**	157
MeanBothW5	MeanBothW5	3557±50.8	9±0.3	**119**	9357±0.7	21±0.6	**121**	4178±53.7	8±0.3	**122**
MeanBothW10	MeanBothW10	3500±16.4	**8±0.7**	138	9357±0.3	20±0.8	141	4215±76.9	12±0.5	150
MeanDemoW5	MeanDemoW10	**3600±11.9**	11±0.8	143	9357±0.6	22±1.5	144	**4281±23.2**	13±0.1	156
MinDemoW10	MinDemoW10	1±0.0	0±0.0	152	9356±1.5	**1±0.0**	157	3998±105.1	**3±0.4**	168
TD3		2561±1311.6	93±9.9	100	**9355±0.2**	47±7.9	100	4783±230.9	35±8.2	100
MinDemoW5	MinDemoW10	49±9.9	**8±1.7**	198	9353±0.4	**2±0.2**	184	4978±119.1	**6±2.5**	221
MeanBothW5	MeanBothW5	3652±67.4	16±1.9	**143**	9352±1.4	62±3.3	**139**	5122±30.3	13±4.1	**147**
MeanBothW10	MeanBothW10	3690±55.2	14±1.0	188	9352±1.1	55±0.5	177	5159±102.0	15±3.0	204
MeanDemoW5	MeanDemoW10	**3821±47.1**	15±0.2	198	9351±0.5	58±1.5	184	**5263±138.4**	18±2.0	218
MinDemoW10	MinDemoW10	1±0.0	0±0.0	219	9350±0.9	**3±0.5**	208	4571±67.5	**5±1.6**	242

filtered action is not performed, the agent does not transition to the next state. We propose predicting the next state using a dynamics model $p_\theta(s_{t+1} \mid s_t, a_t)$. The dynamics model is a simple fully-connected feed-forward network with two hidden layers parameterized by θ trained with L2 loss to imitate the true transition function of the MDP. DEFENDER is summarized in Algorithm 1.

5 Filtering Strategies

We evaluated different filtering strategies for DEFENDER. One compares the current trajectory with the complete demonstration. A second compares the trajectory with the demonstration using an equal length window. For instance, if the current trajectory has length L, we compare the trajectory with the last L transitions of the demonstration. We also test using a fixed window of 5 and 10 transitions applied to only the trajectory, the demonstration and both. After computing the cost between the trajectory and each demonstration, we obtain a set of values for each group. We test selecting the minimum, maximum and average value from each set. This results in 24 methods to compare the trajectory with a group of demonstrations. We can use one method to compare the trajectory with the safe demonstrations and another method to compare it with the unsafe demonstrations, resulting in 576 filtering strategies.

To determine the best filtering strategy, we trained a SAC agent on three tasks and saved the transitions of the episodes. We simulated how the agent would perform the episodes with each strategy. For each episode, we measured its length with the filter active and divide it by the length of the episode without the filter. We also determined if the filter prevents a crash. At the end, we obtained the average episode length percentages and multiplied it by the safe episode rate. We ranked the strategies by the average score for the three tasks, and selected the top 5 strategies for both state and state-action trajectories.

6 Experiments

In this section, we evaluate the effectiveness of our method in enhancing the safety of the underlying RL algorithms, SAC [9] and TD3 [10]. We selected three

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Table 2: Performance, safety and computation time of SAC and TD3 agents enhanced with our algorithm using different filters for state-action trajectories.

Algorithm		Hopper			InvertedDoublePendulum			Walker2d		
		Acc Reward	% Crash	% Time	Acc Reward	% Crash	% Time	Acc Reward	% Crash	% Time
SAC		3474±85.0	64±6.0	100	**9359±0.2**	**14±2.7**	100	3326±254.6	40±4.0	100
MeanBothW5	MeanBothW5	3475±17.7	9±0.8	**119**	9357±0.5	21±1.4	**120**	4207±32.8	**11±1.0**	**123**
MeanBothW10	MeanBothW10	3486±17.0	**8±0.1**	141	9356±0.2	21±1.0	141	**4261±13.0**	14±0.4	155
MinTrajW5	MinTrajW10	1±0.0	0±0.0	1119	8±0.0	0±0.0	728	-4±0.0	0±0.0	1642
MinBoth	MinTrajW10	**3602±0.0**	53±8.4	602	201±4.3	1±0.1	196	3878±0.0	24±4.3	824
MinBoth	MinTrajW5	3443±13.1	11±0.2	460	189±19.2	1±0.2	179	3944±91.7	29±9.2	621
TD3		2561±1311.6	93±9.9	100	**9355±0.2**	**47±7.9**	100	4783±230.9	35±8.2	100
MeanBothW5	MeanBothW5	3645±29.8	**14±1.1**	**143**	9351±0.9	59±12.8	**139**	5196±55.0	17±2.4	**149**
MeanBothW10	MeanBothW10	3651±35.1	15±1.5	193	9351±0.3	58±5.2	178	**5253±78.2**	17±1.9	216
MinTrajW5	MinTrajW10	1±0.0	0±0.0	2442	8±0.0	0±0.0	1297	-4±0.0	0±0.0	3345
MinBoth	MinTrajW10	3759±0.0	60±0.3	1253	204±7.5	0±0.0	282	4991±0.0	**13±2.2**	1623
MinBoth	MinTrajW5	**3849±0.0**	68±0.3	928	167±0.0	0±0.0	251	5220±37.3	**13±3.4**	1196

Table 3: Performance and safety of SAC agent with DEFENDER using and MeanDemoW5 and MeanDemoW10 filters, varying number of demonstrations.

#Demonstrations	Hopper		InvertedDoublePendulum		Walker2d	
	Acc. Reward	% Crash	Acc. Reward	% Crash	Acc. Reward	% Crash
10	3541±25.2	16±0.8	**9356±0.2**	**22±2.1**	4225±46.2	14±0.9
20	3529±13.5	13±1.1	**9356±0.8**	**22±1.7**	4259±26.6	13±1.1
50	**3600±11.9**	**11±0.8**	**9357±0.6**	**22±1.5**	**4281±23.2**	**13±0.1**

tasks from OpenAI Gym: Hopper, Inverted Double-Pendulum, and Walker2d. The episode horizon for these tasks is 1000 steps, but an episode can end early if the agent reaches an unsafe state, leading to a crash if the transition was not filtered. We measured the performance and safety of RL agents by accumulated reward and crash rate, respectively. Agents were trained for 5000 episodes, and each experiment was repeated three times for seed dependency. We used a learning rate of 3^{-4}, batch size of 256, and all the networks have 2 hidden layers with 256 neurons each. We used 50 demonstrations for both safe and unsafe groups, obtained by training a SAC agent and generating a demonstration after every episode. These tasks are non-visual, however DEFENDER can be applied to visual tasks by utilizing an image encoder to convert images into features.

We trained SAC and TD3 agents with and without our algorithm, using the top 5 filtering strategies from the ablation study for both state and state-action trajectories. Results are shown in Tables 1 and 2. Our algorithm incurs extra computational cost which we present in the tables. Overall, both SAC and TD3 agents vary consistently across filtering strategies for the same task. Results show that state trajectories are preferred over state-action trajectories. Not only are they less computationally expensive, but they consistently lead to far fewer crashes for the same task and filtering strategy. Some filtering strategies are too strong and prevent the agent from interacting with the environment and learning the task. Those that allow the agent to interact with the environment lead the agent to the same performance in the case of InvertedDoublePendulum and to higher performance in the case of Hopper and Walker. More importantly, the crash rate is significantly decreased with some exceptions in the Inverted-DoublePendulum task. With an appropriate filtering strategy, DEFENDER is

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

able to significantly reduce the crash rate. However, selecting a good filtering strategy for the task may require some trial and error.

Lastly, we evaluated the impact of the number of demonstrations on the performance by training an agent with DEFENDER using 'MeanDemoW5' for the safe set, and 'MeanDemoW10' for the unsafe set as the filtering strategy, varying the number of demonstrations. Results are shown in Table 3. DEFENDER is able to increase the agent's safety with small data sets, and it can be further improved by increasing the demonstration data set size.

7 Conclusion

In this paper, we introduced the DEFENDER algorithm that can be integrated with any RL algorithm for improving the safety of agents during learning. Ablation studies helped identify effective filtering strategies, which we evaluated on state-of-the-art RL algorithms and multiple tasks. The results demonstrate significant enhancement in the safety of the learning agent while maintaining or improving performance. However, there is room for improvement, as the filtering strategy may be overly protective for certain tasks, requiring many iterations to select an appropriate strategy. Our work highlights the potential of using demonstrations for task-agnostic enhancement of RL algorithm safety.

References

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[2] Alfredo Reichlin, Giovanni Luca Marchetti, Hang Yin, Ali Ghadirzadeh, and Danica Kragic. Back to the manifold: Recovering from out-of-distribution states. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[3] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 2019.

[4] Mathieu Godbout, Maxime Heuillet, Sharath Chandra Raparthy, Rupali Bhati, and Audrey Durand. A game-theoretic perspective on risk-sensitive reinforcement learning. In *SafeAI@ AAAI*, 2022.

[5] Katie Kang, Paula Gradu, Jason J Choi, Michael Janner, Claire Tomlin, and Sergey Levine. Lyapunov density models: Constraining distribution shift in learning-based control. In *International Conference on Machine Learning*. PMLR, 2022.

[6] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*, 2011.

[7] Richard Bellman and Robert Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2), 1959.

[8] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[9] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 2018.

[10] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 2018.