Prioritizing Datapoints in RL with Reducible Loss

Shivakanth Sujit, Somjit Nath, Pedro Braga, Samira Ebrahimi Kahou

CIFAR ÉTS Le génie pour l'industrie Mila

Motivation

- Prioritized experience replay (PER) prioritizes datapoints with high TD error
- However there can be points w/ high TD error that are noisy or not learnable
- That is, points w/ high TD ≠ points that the model can actually learn from
- Instead the agent should focus on points with reducible loss

Solution

- Propose a simple change to the priority used for sampling
- Instead of TD error, use a measure how much the TD can be potentially reduced as the priority.
- So you avoid repeatedly sampling points which the agent has been unable to learn from

Solution

- In practice, we use the difference in the TD error between the online model and the target network

Termed the Reducible Loss (ReLo)

Solution

- In practice, we use the difference in the TD error between the online model and the target network

$$TD_\theta = \left(V_\theta(s_t) - (r_t + \gamma V_{\theta_t}(s_{t+1}))\right)^2$$

Where θ is the network considered and θ_t is the target network.

Solution

- In practice, we use the difference in the TD error between the online model and the target network
- That is, we compute the TD error with respect to the online model (TD_{online}) and the target network (TD_{target}) and

$$\mathrm{ReLo} = TD_{online} - TD_{target}$$

Rationale

- ReLo ensures that points that were unimportant under PER, remain so.
- If the TD error was already low, then the ReLo will also be low
- However, the difference lies when considering points with high TD

Rationale

- If points that previously had high TD continue to do so even after several updates, then those points might be noisy or not learnable
- In this case, ReLo will be low since TD_{online} and TDt_{arget} will both be high

Rationale

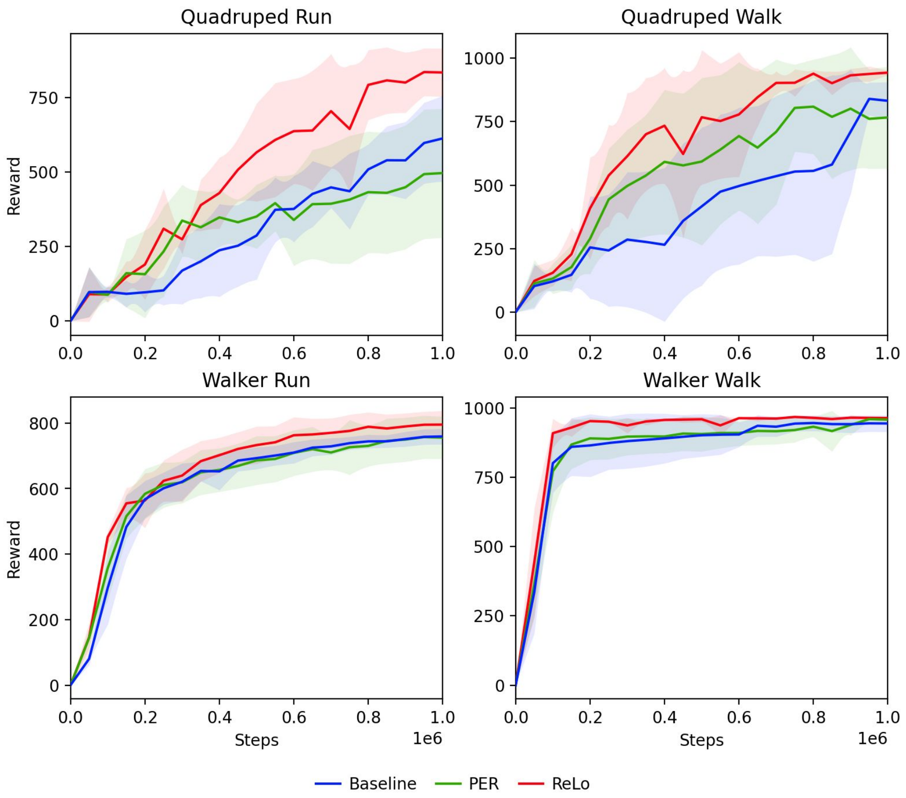
- If the points were forgotten, then their current TD error could have increased, but because TD_{target} is lower, we know there is potential to reduce the loss
- Hence we should prioritize these points for learning.

Results

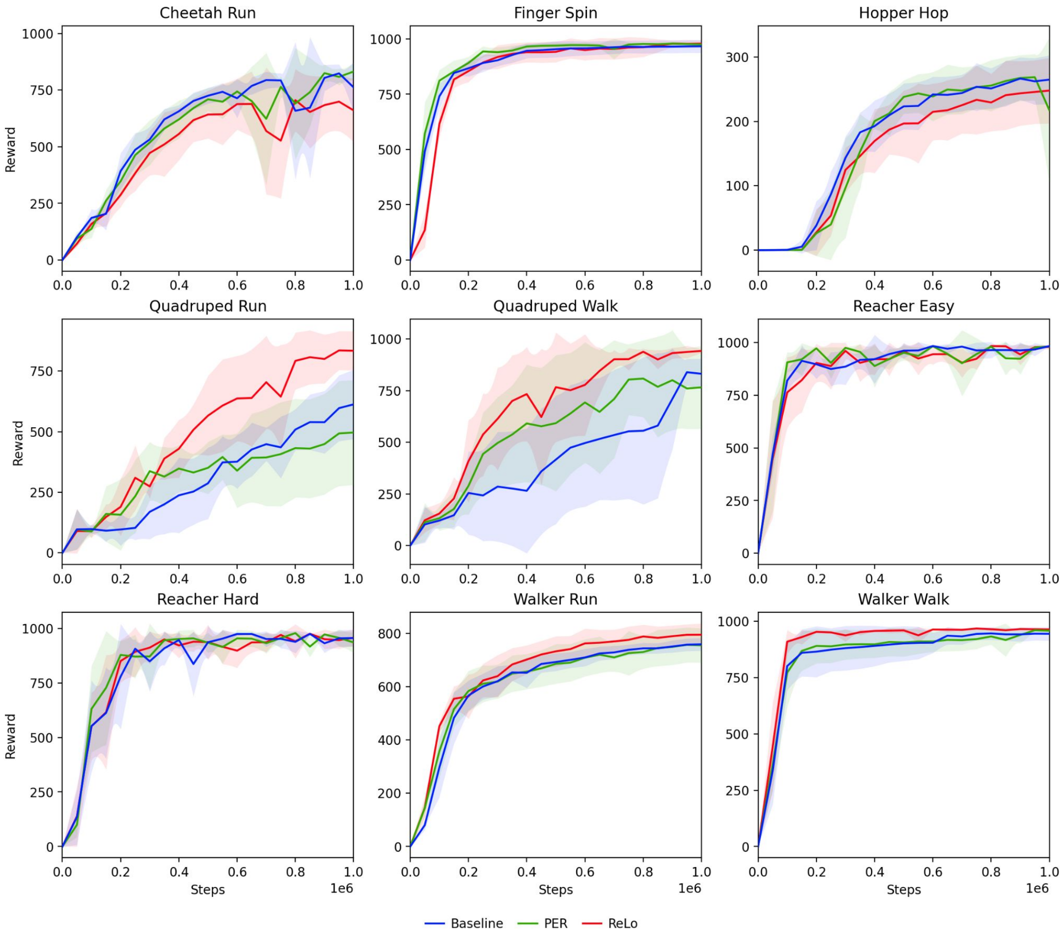
- We compared ReLo with PER in continuous and discrete control tasks.
- Our experiments show that ReLo improves performance over PER
- This is especially true in cases where adding PER actually hurts performance

Continuous Control

- Using Soft Actor Critic as a baseline, we compared with PER and ReLo on the DeepMind Control Suite
- ReLo generally leads to improved performance over PER and baseline SAC



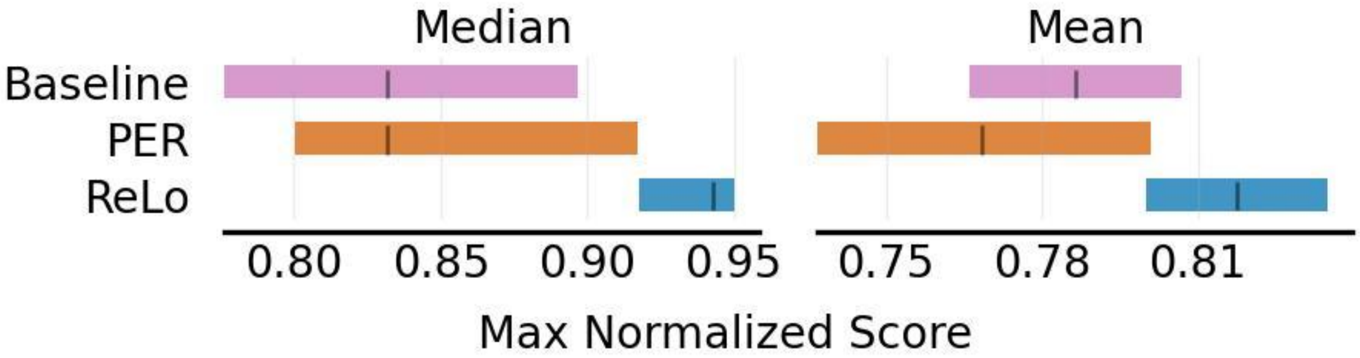
Shaded region corresponds to 1 std dev across 5 seeds



Shaded region corresponds to 1 std dev across 5 seeds

Continuous Control

- Furthermore, we include aggregated metrics based on rliable[1]
- They model the performance across runs as a random variable and report statistical measures with interval estimates

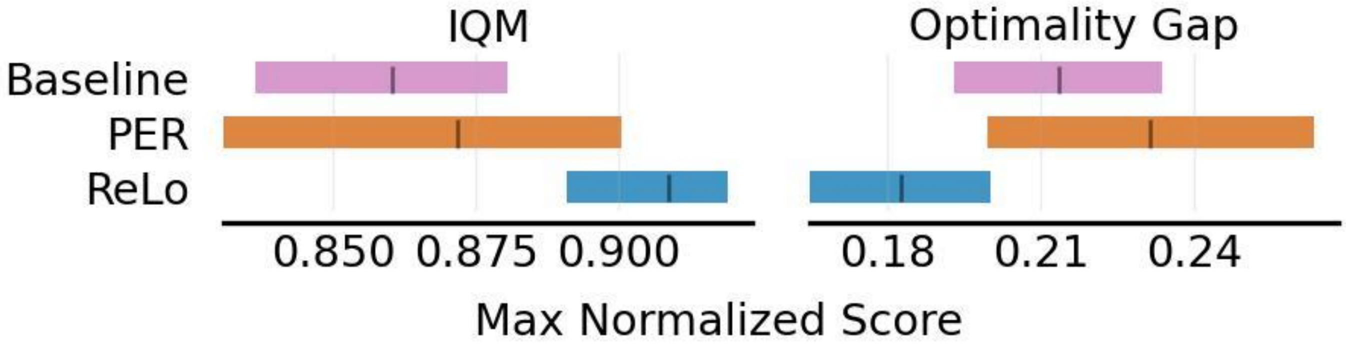


Scores are normalized based on the max score in DMC, i.e. 1000

[1] Aggarwal et. al, Deep Reinforcement Learning at the Edge of the Statistical Precipice, NeurIPS 2021

Continuous Control

- IQM (Interquartile Mean): Mean across the middle 50% of runs
- Optimality Gap: Measure of runs with normalized scores < 1, i.e how far off are the runs from optional behaviour. (Lower is better)

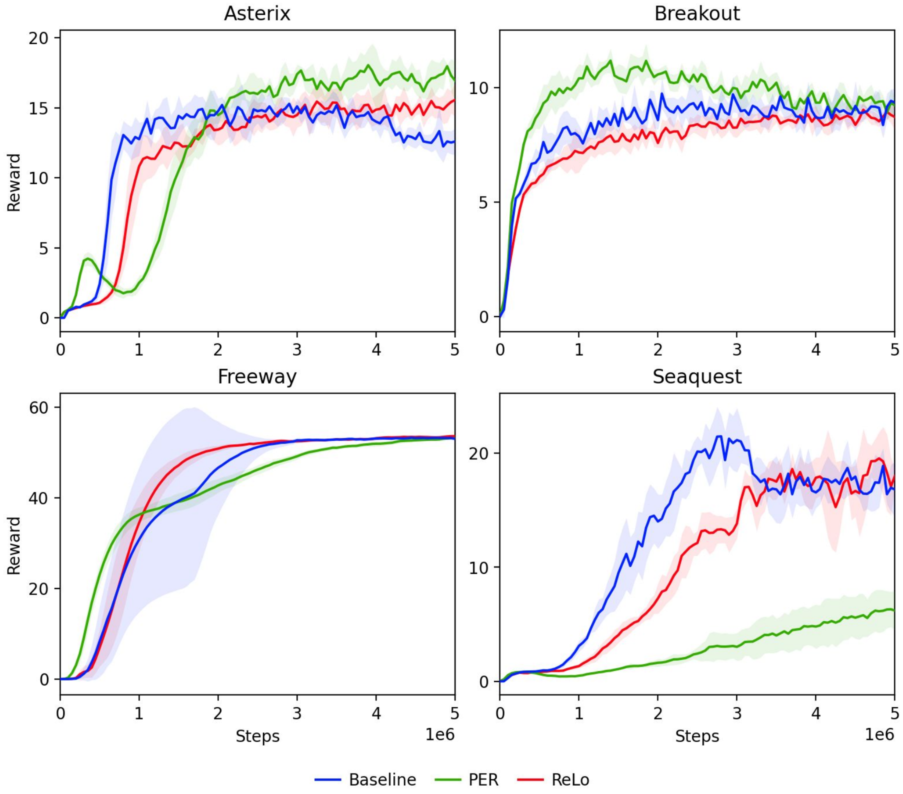


Scores are normalized based on the max score in DMC, i.e. 1000

[1] Aggarwal et. al, Deep Reinforcement Learning at the Edge of the Statistical Precipice, NeurIPS 2021

Discrete Control

- Using DQN as a baseline, we compared with PER and ReLo on the MinAtar benchmark

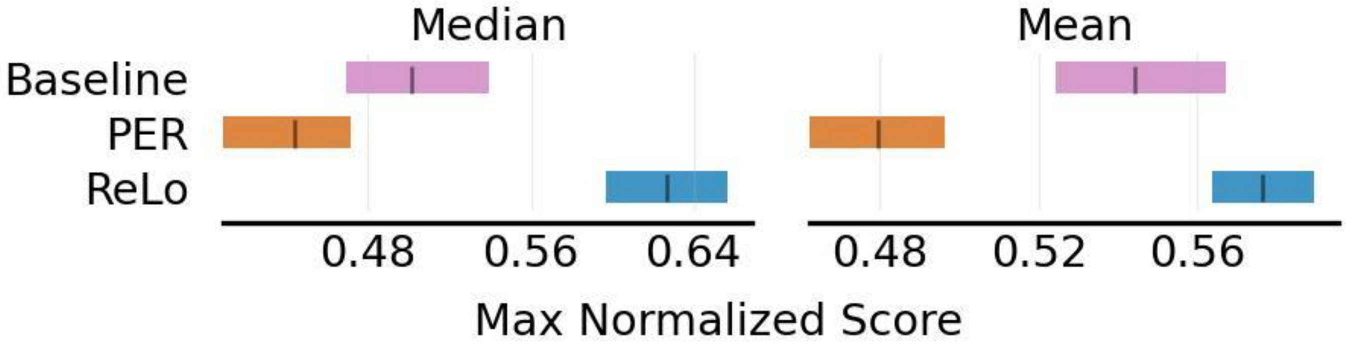


Shaded region corresponds to 1 std dev across 5 seeds

[1] Young and Tian, MinAtar: An Atari-Inspired Testbed for Thorough and Reproducible Reinforcement Learning Experiments, arXiv:1903.03176

Discrete Control

- While naive prioritization hurts performance in MinAtar, ReLo matches or exceeds performance of the baseline.

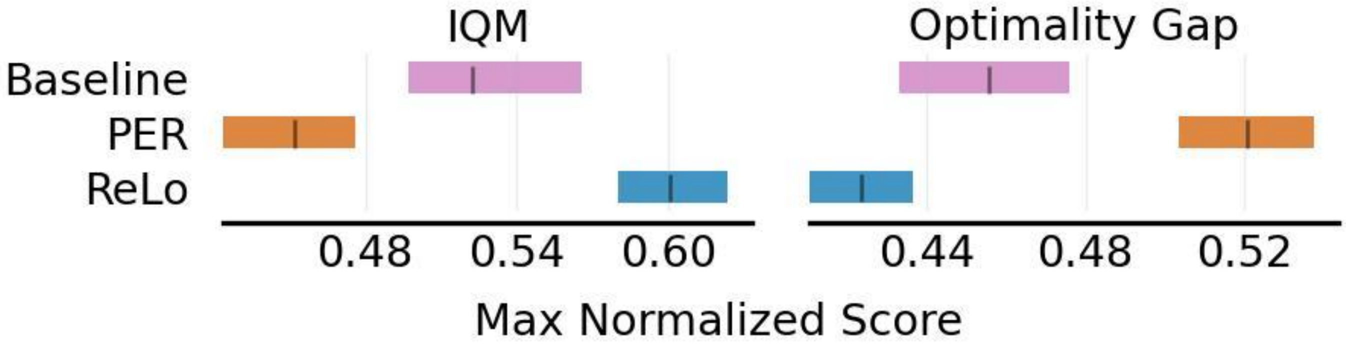


Scores are normalized based on max scores from MinAtar [1]

[1] Young and Tian, MinAtar: An Atari-Inspired Testbed for Thorough and Reproducible Reinforcement Learning Experiments, arXiv:1903.03176

Discrete Control

- IQM (Interquantile Mean): Mean across the middle 50% of runs
- Optimality Gap: Measure of runs with normalized scores < 1, i.e how far off are the runs from optional behaviour. (Lower is better)

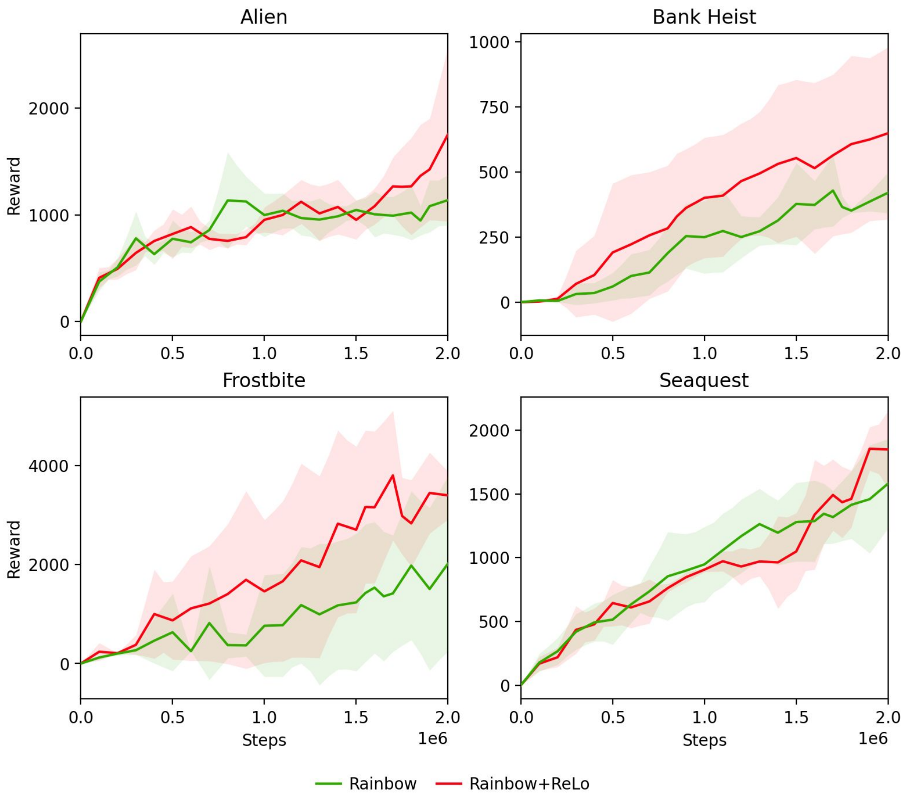


Scores are normalized based on max scores from MinAtar [1]

[1] Young and Tian, MinAtar: An Atari-Inspired Testbed for Thorough and Reproducible Reinforcement Learning Experiments, arXiv:1903.03176

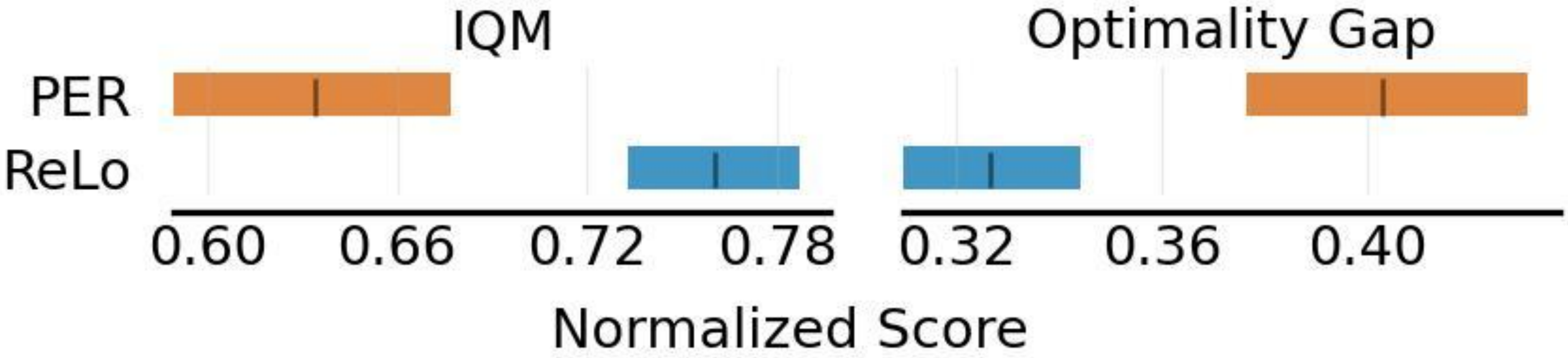
Arcade Learning Environment

- We evaluated ReLo on a subset of tasks from ALE in the compute constrained setting of 2M frames.
- Rainbow + ReLo achieves better performance than vanilla Rainbow in nearly all the tested environments



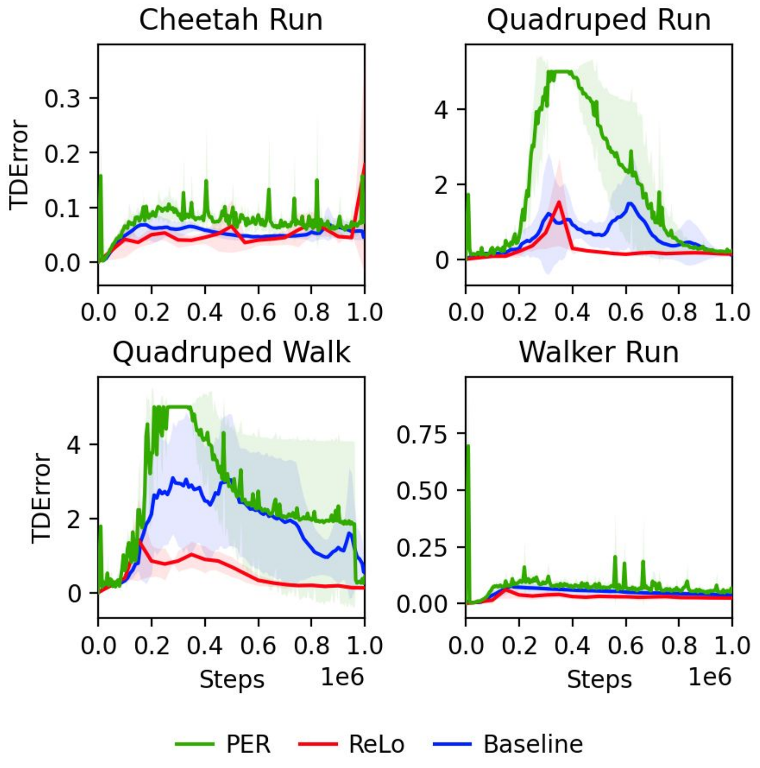
Shaded region corresponds to 1 std dev across 5 seeds

- When aggregated across 21 environments, ReLo shows a clear improvement over PER.



Comparison of TD Loss Minimization

- While PER has higher TD error during training, consistently leads to lower TD error across environments and benchmarks.
- This empirically validates our claim that ReLo prioritizes samples whose loss can be reduced



Conclusion

- Vanilla PER is based on high loss, not if a sample is learnable.
- ReLo prioritizes samples that have the highest potential for loss reduction, retaining positive behaviors of PER while addressing the above issue.
- Can be used with any off policy Q learning method with minimal code additions and computational overhead above PER.

Conclusion

- Empirically validated across diverse tasks in continuous and discrete control.
- Future work could analyse the difference in points sampled between ReLo and PER as well as dynamics of priority during training.

github.com/shivakanthsujit/reducible-loss