Proximal Curriculum Learning with State Novelty

- Use zone of proximal development (pick start state with probability of success $PoS \approx 0.5$) [Tzannetos et al. 2023]
- approximate PoS via value-function [Tzannetos et al. 2023]

Proximal Curriculum Learning with State Novelty

- Use zone of proximal development (pick start state with probability of success $PoS \approx 0.5$) [Tzannetos et al. 2023]
- approximate PoS via value-function [Tzannetos et al. 2023]
- *Own Idea:* already using state to guide student, why not pick novel states

→ should help value-function converge

→ should still pick mostly promising states

Proximal Curriculum Learning with State Novelty

- Use zone of proximal development (pick start state with probability of success *PoS* ≈ 0.5) [Tzannetos et al. 2023]
- approximate *PoS* via value-function [Tzannetos et al. 2023]
- *Own Idea:* already using state to guide student, why not pick novel states

→ should help value-function converge

→ should still pick mostly promising states

- trade off via hyperparameter between *PoS* and novelty
- alternatively incorporate state novelty as intrinsic reward: directly influences *PoS*

Practical Considerations

- *Implementation Overhad:* no given code, but modifications (including hyperparameters) to stable-baselines listed for [Tzannetos et al. 2023]

Practical Considerations

- *Implementation Overhad:* no given code, but modifications (including hyperparameters) to stable-baselines listed for [Tzannetos et al. 2023]
- *Compute Overhead:* should be reasonably cheap depending on the state novelty algorithm and environment

Practical Considerations

- *Implementation Overhad:* no given code, but modifications (including hyperparameters) to stable-baselines listed for [Tzannetos et al. 2023]
- *Compute Overhead:* should be reasonably cheap depending on the state novelty algorithm and environment
- *Reproducibility:* Should reproduce given limited modifications to stable-baselines. If not try to use rollouts to approximate PoS.

Practical Considerations

- *Implementation Overhad:* no given code, but modifications (including hyperparameters) to stable-baselines listed for [Tzannetos et al. 2023]
- *Compute Overhead:* should be reasonably cheap depending on the state novelty algorithm and environment
- *Reproducibility:* Should reproduce given limited modifications to stable-baselines. If not try to use rollouts to approximate PoS.
- MiniGrid should be good, sparse and many states