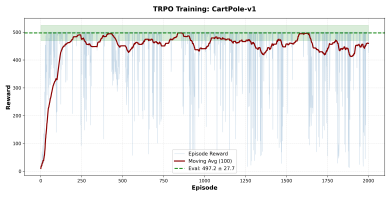
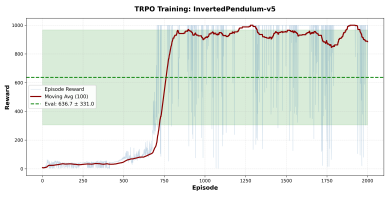
TRPO, PPO, and VPG: Results on Five Datasets Each

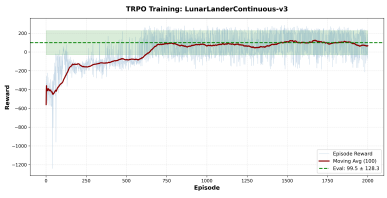
TRPO



(a) CartPole-v1

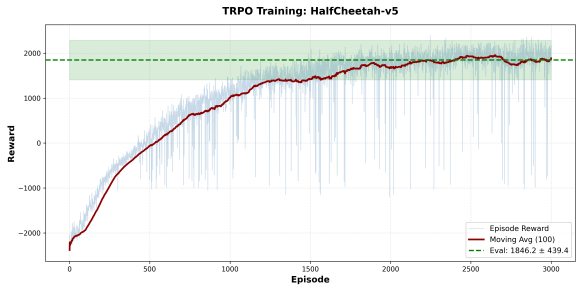


(b) InvertedPendulum-v5

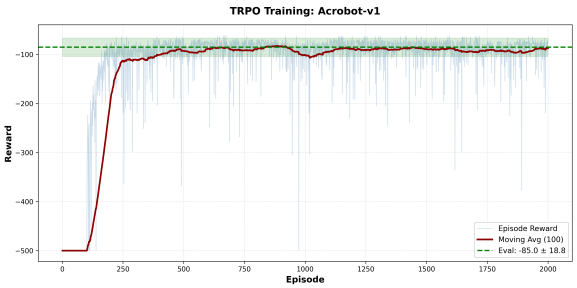


(c) LunarLanderContinuous-v3

Figure 1: TRPO performance (1/2).



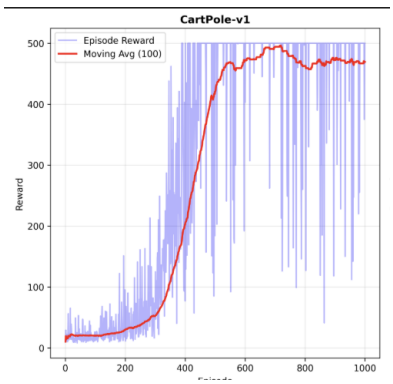
(d) HalfCheetah-v5



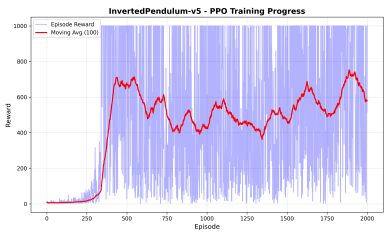
(e) Acrobat-v1

Figure 1: TRPO performance (2/2).

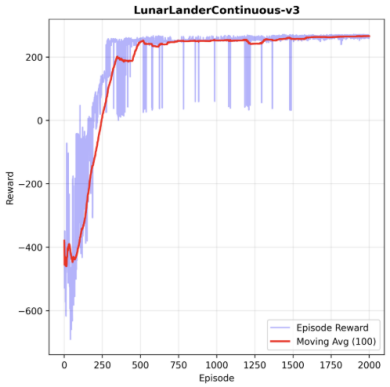
PPO



(a) CartPole-v1

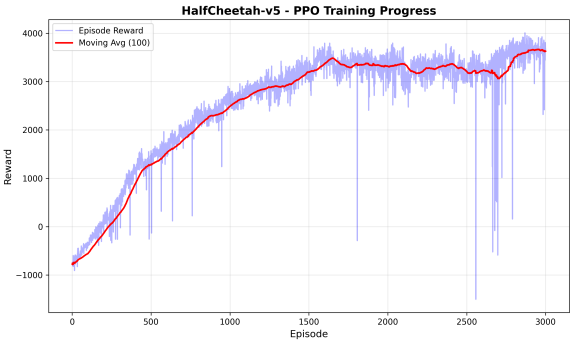


(b) InvertedPendulum-v5

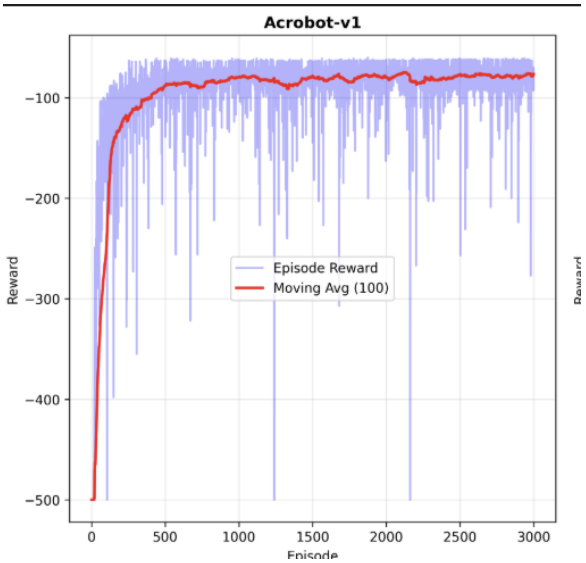


(c) LunarLanderContinuous-v3

Figure 2: PPO performance (1/2).



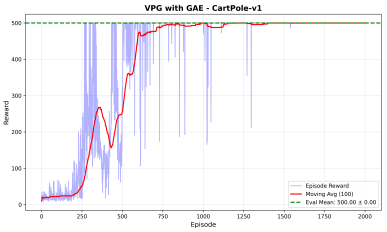
(d) HalfCheetah-v5



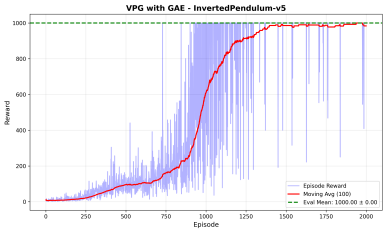
(e) Acrobat-v1

Figure 2: PPO performance (2/2).

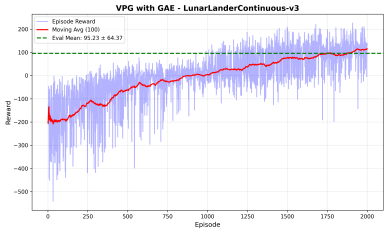
VPG



(a) CartPole-v1

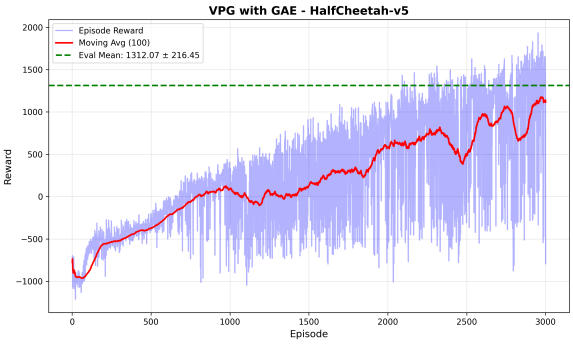


(b) InvertedPendulum-v5

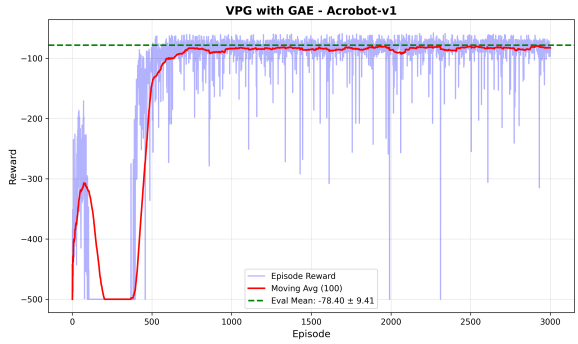


(c) LunarLanderContinuous-v3

Figure 3: VPG performance (1/2).



(d) HalfCheetah-v5



(e) Acrobot-v1

Figure 3: VPG performance (2/2).

Discussion and Results

Across datasets all three reached strong policies, but differences came from tuning and budget. In a few cases VPG+GAE briefly outscored PPO and TRPO; this was driven by suboptimal hyperparameters and tight timing, not a fundamental edge. With comparable tuning the ranking settled: PPO gave the best stability–speed trade-off and lowest time-to-threshold, TRPO was most monotonic but slower per update, and VPG+GAE had the highest variance. Implementation effort in this project was lowest for PPO, then VPG+GAE, and highest for TRPO, since PPO used a single clipped objective and plain SGD, VPG+GAE required careful advantage

and baseline handling, and TRPO needed conjugate-gradient, Fisher-vector products, and line search. Simple tasks let VPG match or momentarily exceed PPO; harder tasks favored PPO, with TRPO close when fully tuned. Discrepancies are best explained by hyperparameters, seeds, and wall-clock limits rather than algorithm correctness.