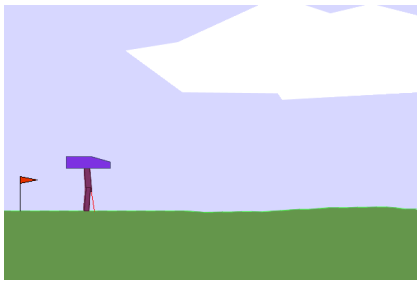
Name: Troy Tian **Email**: troyhtian@g.ucla.edu

Prior experience: was a fellow for fall technical track with AI Safety @ UCLA. Was introduced to how RL works intuitively and mathematically + given sample exercises to practice implementation. Beyond this, I had neither learned about nor coded any deep learning or other RL models.

Timeline: Spent a total of 10 hours (broken up over several days) on this project.

- 1-3: reviewed requirements. Studied PPO implementation as a surrogate objective function (Schulman et al., 2017). Studied Stable-Baselines3 (Raffin et al., 2021). Studied background and justifications for PPO, primarily Generalized Advantage Estimation (GAE) technique (Schulman et al., 2016). Studied reward structure and observation/action spaces of BipedalWalker-v3 environment (Farama Foundation, 2023). Generated basic implementation, environment rendering + sample gifs. Organized GitHub repo and file structure.
- 4-6: researched general areas for improvement outlined in requirements. Wrote a script that evaluates a model's true score using a 10-episode exam (absent scaling from reward wrappers). Ran basic experiments + ablations.
- 7-8: Intensified training timescales and scope somewhat, aiming to fully stabilize model + solve environment. Ran more detailed experiments examining later-stage training.
- 9-10: Final phases of training. Solidified conclusions about techniques studied. Researched possible next steps.

Best performance:
Trained over 1 million timesteps



Compute resources: CPU of MacBook Air.
Chip: Apple M4
Memory: 24 GB
Total cores: 10 (4 performance, 6 efficiency)

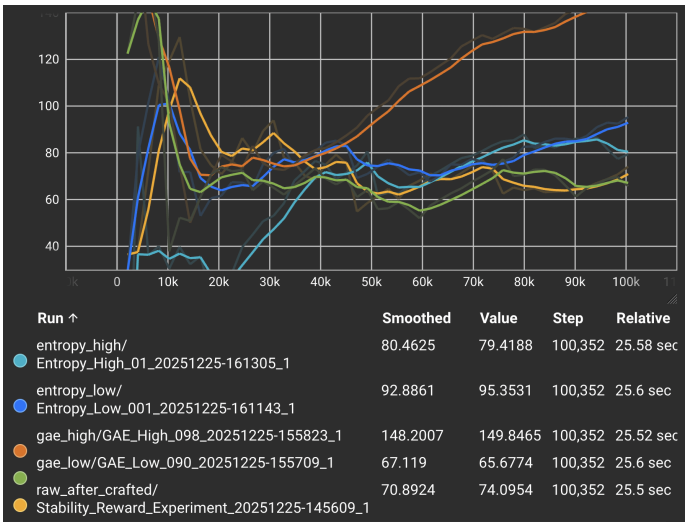
Techniques and justifications

- Basic reward crafting: punishment for tilting hull too far helped reduce frequency of falls and led to significant reward increase; precise coefficient came to be adjusted (see curriculum learning).
- Curriculum learning: employed throughout training based on research + empirical observations, somewhat enabling escape from suboptimal gaits + allowing for significantly faster convergence.
- Entropy regularization: entropy rewards uncertainty/stochasticity (Schuman et al., 2017; OpenAI, 2018). Annealing regularization coefficient progressively over training created implicit entropy scheduling, assisting model transition from exploration to exploitation phases.
- Advantage estimation tuning: GAE lambda governs advantage estimates' focus on either longer-term reward information or shorter-term temporal difference errors (Schuman et al, 2016). Raising GAE coefficient and thus aggressively propagating delayed rewards accelerated early learning but later created instability, requiring curriculum-style optimization of bias-variance tradeoff.
- Generalized State-Dependent Exploration: gSDE predicates sampled action noise on current state (Raffin et al., 2020), encouraging structured exploration early on while allowing a natural taper into exploitation in continuous action spaces. Did not always improve gait or robustness, tending instead to reduce exploratory diversity and somewhat encourage policy drift or convergence to a local optima (discussed in issues encountered).
- Observation normalization: stabilizes training by rescaling observation dimensions to approximately zero mean and unit variance, improving conditioning (Schuman et al., 2017; OpenAI, 2018).
- Parallel environment sampling: allows independent duplicate environments to run simultaneously, improving experience collection rate (Raffin et al., 2021). Did not necessarily improve final performance due to a tendency to accelerate convergence to local optima (discussed in issues encountered).
- PPO stabilization: enforces conservative learning. Tuning hyperparameters e.g. action clipping via PPO objective as well as target KL (creating early stopping), batch size, and epochs ensured more stable policy refinement.
- Advanced reward crafting: from empirical observations, it became helpful to add and adjust rewards and penalties for values like x- and y-velocity, leg distance, and time alive.

Ablation studies (every graph is of mean episode reward)

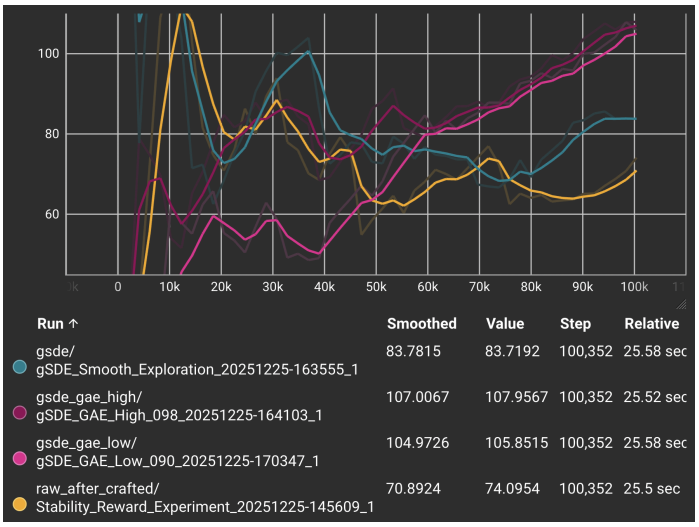
Although I don't have room to showcase most of my studies, these ones were most important to my conclusions.

High entropy, low (but still >0) entropy, high GAE, low GAE, respectively, relative to a basic reward-crafted model. High GAE seems to show a clear advantage; various combinations of entropy and GAE values were tested but deemed ineffective (discussed in issues encountered).



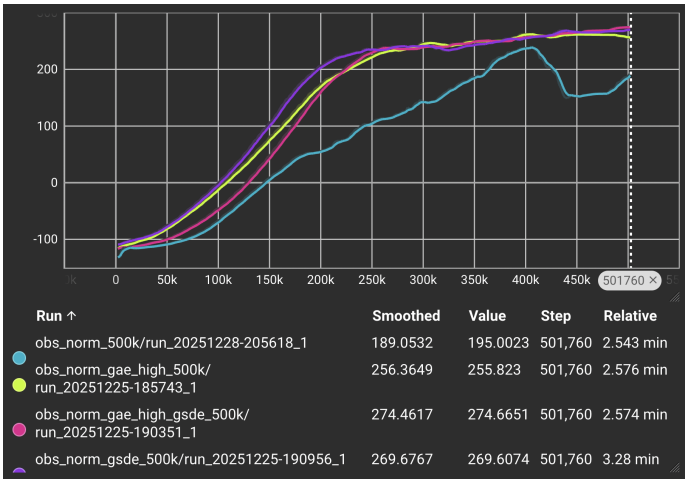
Run	Smoothed	Value	Step	Relative
entropy_high/ Entropy_High_01_20251225-161305_1	80.4625	79.4188	100,352	25.58 sec
entropy_low/ Entropy_Low_001_20251225-161143_1	92.8861	95.3531	100,352	25.6 sec
gae_high/GAE_High_098_20251225-155823_1	148.2007	149.8465	100,352	25.52 sec
gae_low/GAE_Low_090_20251225-155709_1	67.119	65.6774	100,352	25.6 sec
raw_after_crafted/ Stability_Reward_Experiment_20251225-145609_1	70.8924	74.0954	100,352	25.5 sec

gSDE, gSDE with a high GAE, and gSDE with a low GAE, respectively, relative to a basic model. gSDE is an improvement, but with high or low GAE seems better. Not as rewarding as purely high GAE (discussed in issues encountered).



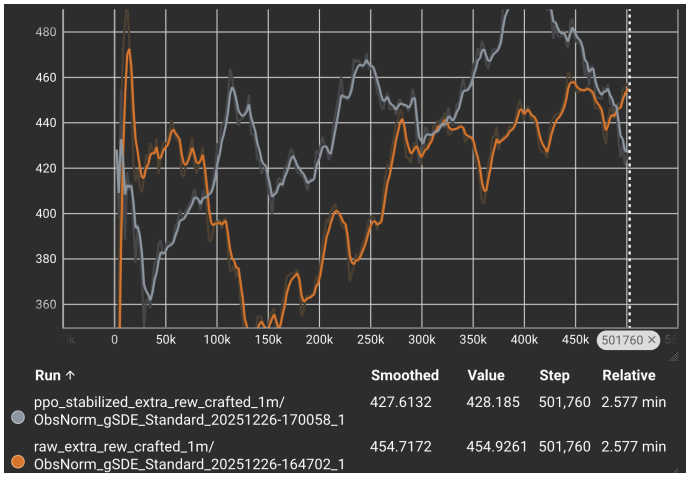
Run	Smoothed	Value	Step	Relative
gsde/ gSDE_Smooth_Exploration_20251225-163555_1	83.7815	83.7192	100,352	25.58 sec
gsde_gae_high/ gSDE_GAE_High_098_20251225-164103_1	107.0067	107.9567	100,352	25.52 sec
gsde_gae_low/ gSDE_GAE_Low_090_20251225-170347_1	104.9726	105.8515	100,352	25.58 sec
raw_after_crafted/ Stability_Reward_Experiment_20251225-145609_1	70.8924	74.0954	100,352	25.5 sec

Observation normalization alone, along with ON + high GAE, ON + gSDE, and ON + high GAE + gSDE, respectively; this time on a longer timescale of 500k steps. Although all those outside of bare ON seem to be performing similarly well, later-stage training graphs indicate that both of the combinations involving gSDE create issues as training continues (discussed in issues encountered).



Run	Smoothed	Value	Step	Relative
obs_norm_500k/run_20251228-205618_1	189.0532	195.0023	501,760	2.543 min
obs_norm_gae_high_500k/ run_20251225-185743_1	256.3649	255.823	501,760	2.576 min
obs_norm_gae_high_gsde_500k/ run_20251225-190351_1	274.4617	274.6651	501,760	2.574 min
obs_norm_gsde_500k/run_20251225-190956_1	269.6767	269.6074	501,760	3.28 min

Non-stabilized and stabilized, respectively; this time on a longer timescale of 1M steps. Note that this uses a different reward scale and so is no longer close to reflecting the true score of this model. Clearly, though, PPO stabilization via targeted hyperparameter tuning improves late-stage training, peaking far higher here.



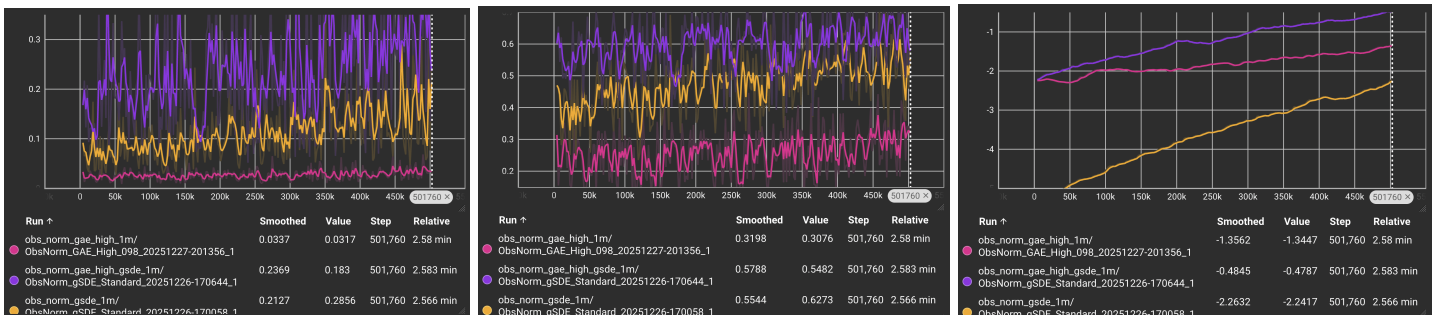
Run	Smoothed	Value	Step	Relative
ppo_stabilized_extra_rew_crafted_1m/ ObsNorm_gSDE_Standard_20251226-170058_1	427.6132	428.185	501,760	2.577 min
raw_extra_rew_crafted_1m/ ObsNorm_gSDE_Standard_20251226-164702_1	454.7172	454.9261	501,760	2.577 min

Issues encountered

- Observation normalization and reward normalization were both initialized into the yaml files I used for my experiments at the same time. Although not immediately proven ineffectual, as I added more carefully optimized coefficients for advanced reward-crafting, it quickly stopped being compatible with my granular shaping terms.

- As mentioned above, I tested various combinations of GAE and entropy values. However, although any increase in entropy as well as especially a high GAE value seemed to individually improve the mean reward the most, testing them in combination yielded suboptimal results. This is explainable due to the fact that any increase in entropy alongside GAE lambda would both individually inject variance into PPO updates, adding unnecessary noise in both advantage estimation and action selection and introducing a large amount of instability into the learning signal. For a contact-rich task like learning to walk, this quickly becomes untenable, as was evidenced by the unstable updates and degraded performance I observed as these models continued to train.
- As mentioned above, gSDE, and in particular the combination of gSDE with a high GAE (which empirically seems to most improve results overall), becomes more unstable than is justifiable in later training during what should be a policy refinement as opposed to exploratory phase. gSDE by itself seems somewhat prone to policy drift and, while not necessarily becoming brittle, locks in a suboptimal gait. The combination of gSDE with GAE, similarly to the combination of raising GAE and entropy detailed above, lead to even more unstable training. Increased stochasticity during trajectory generation combined with amplified propagation of long-horizon reward information degraded credit assignment and created a model that was both brittle and too high in variance. I initially believed this combination might be potent, but I was instead compelled to pivot to more rigorous ablation testing, eventually arriving at the conclusion not to continue using gSDE.

approx_kl **clip_fraction** **entropy_loss**



- As mentioned above, I attempted to implement parallel environment sampling in order to train more efficiently at one point during training; the technique increases the rate and diversity of data collection. However, once again, there is a tradeoff of stability, this time with throughput, and once again in late training it becomes unusable as it causes the model to optimize too aggressively and even become brittle as its policy drifts. I had also thought this would be a good idea initially, but I was instead simply forced to return to what I'd had before as I was too time-constrained this time to investigate ways to absolve this issue specifically. I would, however, like to do exactly that if given more time (see conclusions and next steps).

approx_kl **clip_fraction** **entropy_loss**

Conclusions and next steps: Overall, I conclude that curriculum learning-style optimizations of tradeoffs between exploration/exploitation and advantage/reward are among the most crucial parts of training an MLP for this environment. Observation normalization unequivocally improved training stability, and the phase-training structure I had employed was well-suited to both early exploration as well as the tuning of GAE lambda, clip range, target KL, and other hyperparameters for late-stage robustness. The effectiveness of gSDE seemed limited by the dominance of stability and

reward-structure challenges rather than insufficient exploration noise; similarly, parallel environment sampling seemed ill-suited to the training of a stable model given the scope of this project. Although I was constrained by compute and especially by demanding time constraints (family obligations etc.), I believe there are many improvements to be made with further investigation, especially in novel approaches to parallel training, including: correlating optimization epochs with rollouts and other training data; exploiting parallel data collection using off-policy algorithms with experience replay; or further tweaking KL-based trust-region constraints to preserve training stability (Schulman et al., 2017). Additionally, off-policy algorithms with experience replay may better exploit parallel data collection without destabilizing learning, suggesting a promising direction for further exploration (Haarnoja et al., 2018; Fujimoto et al., 2018).

Works cited

- Farama Foundation. (2023). *Gymnasium: A Standard API for Reinforcement Learning Environments*. https://gymnasium.farama.org
- Fujimoto, S., Hoof, H., & Meger, D. (2018). *Addressing Function Approximation Error in Actor-Critic Methods*. Proceedings of ICML.
- Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. Proceedings of the 35th International Conference on Machine Learning (ICML).
- OpenAI. (2018). *Spinning Up in Deep Reinforcement Learning*. https://spinningup.openai.com
- Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). *Stable-Baselines3: Reliable Reinforcement Learning Implementations*. Journal of Machine Learning Research, 22(268), 1–8.
- Raffin, A., Kober, J., & Stulp, F. (2020). *Smooth Exploration for Robotic Reinforcement Learning*. arXiv preprint arXiv:2005.05719. https://arxiv.org/pdf/2005.05719
- Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2016). *High-Dimensional Continuous Control Using Generalized Advantage Estimation*. arXiv preprint arXiv:1506.02438. https://arxiv.org/pdf/1506.02438
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal Policy Optimization Algorithms*. arXiv preprint arXiv:1707.06347. https://arxiv.org/pdf/1707.06347