RL-Course 2024/25: Final Project Report

PuckAI: Kim Isabella Zierahn*

February 26, 2025

1 Introduction

From developing autonomous cars to defeating the world champion in Go, reinforcement learning (RL) has become one of the most powerful techniques in artificial intelligence[3]. RL is next to supervised and unsupervised learning one of the three primary areas in the field of machine learning. It learns how to behave in an environment using interactions to maximize the expected reward. In this paper, we aim to develop a RL agent that is able to solve simple tasks and a more complex, simulated hockey game.

1.1 The Hockey Environment

The hockey environment[1] is a custom environment of the gymnasium API[7] created by the Autonomous Learning Group at the University of Tübingen. It is a simulated environment designed for reinforcement learning applications, consisting of two players playing laser-hockey against each other. Player 1 (left player) can be controlled by our agent, while player 2 (right player) can be set to a weak or strong basic opponent, a human opponent or a second RL agent. The environment provides different training modes: train shooting, train defense and normal game mode.
The environment consists of 18-dimensional observation spaces and 4-dimensional continuous action spaces for each agent, with action values constrained to the interval [-1, 1]. The environment implements four distinct aspects of the reward. Terminal rewards are determined by the winner of the game, with values of +1 for victory, -1 for defeat, and 0 for draws. Additional rewards are awarded based on closeness to the puck, contact with the puck and direction of puck. The maximum total reward is +10.

2 Method

The aim of this paper is to develop a RL agent that is able to handle simple and complex environments. The base model is a Proximal Policy Optimization (PPO) model which will be modified and enhanced by using ideas from Phasic Policy Gradient (PPG), and common failure modes of PPO.

2.1 Proximal Policy Optimization (PPO)

While traditional policy gradient methods struggle with instability due to large step sizes, inefficient updates, and the challenge of selecting an appropriate learning rate, PPO improves training stability by constraining policy updates[6]. By using a clipped surrogate objective, PPO prevents excessively

*Matrikelnummer 6635183, kim-isabella.zierahn@student.uni-tuebingen.de, Quantitative Data Science Methods

[1]Simple Hockey Environments by Georg Martius, Autonomous Learning Group, Uni Tuebingen (2018): https://github.com/martius-lab/hockey-env

large updates, ensuring more stable and efficient learning. PPO is a model-free, on-policy RL algorithm because it does not explicitly model the environment and it updates the policy only with data collected using the current policy. It is an actor-critic method because it uses a policy, which is updated by the actor, and a value function, which is updated by the critic. PPO simplifies trust region policy optimization (TRPO) by using a clipped surrogate objective. The ratio of probability between the old and the new policies are the importance weights:

$$\rho_t(\theta) = \frac{\pi_\theta(s_t, a_t)}{\pi_{\theta old}(s_t, a_t)} \tag{1}$$

PPO's objective function takes the minimum between the original and clipped values, preventing the policy update from becoming excessively large. This ensures that the ratio ρ lies within [1 - ϵ, 1 + ϵ]. The total PPO loss is augmented with an error term on the value estimation and an entropy term.

$$L^{CLIP}(\theta) = \mathbb{E}\left[\min(\rho_t(\theta) \cdot A_t,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon) \cdot A_t)\right] \tag{2}$$

$$L = \mathbb{E}_{\pi_{\theta old}}\left[L^{CLIP}(\theta) - c_1 L^{VF}(w) + c_2 \mathcal{H}[\pi_\theta]\right] \tag{3}$$

with A_t being the advantage function at timestep t, $L^{VF}(w)$ being the square loss towards MC targets, and $\mathcal{H}[\pi_\theta]$ being the entropy of policy.

2.1.1 Phasic Policy Gradient (PPG)

Phasic Policy Gradient (PPG) by Cobbe et al. (2020) extends the PPO framework by separating policy and value function training into distinct phases[1]. PPG consists of two training phases: a policy phase and a periodic auxiliary phase. During the policy phase, we are training the agent with vanilla PPO, optimizing the clipped surrogate objectives (2) and (3). During the auxiliary phase, we distill features from the value function into the policy network. We optimize the policy using a joint objective:

$$L^{joint} = L^{aux} + \beta_{clone} \cdot \hat{\mathbb{E}}_t[KL[\pi_{old}(\cdot|s_t), \pi_\theta(\cdot|s_t)]] \tag{4}$$

$$L^{aux} = \frac{1}{2} \cdot \hat{\mathbb{E}}_t\left[V_{\theta_\pi}(s_t) - \hat{V}_t^{targ})^2\right] \tag{5}$$

with a behavioral cloning loss, L^{aux} being an arbitrary auxiliary loss, V_{θ_π} being an auxiliary value head of the policy network, and $\hat{V}_t^{targ}$ being value function targets.

2.1.2 Three Failures of PPO

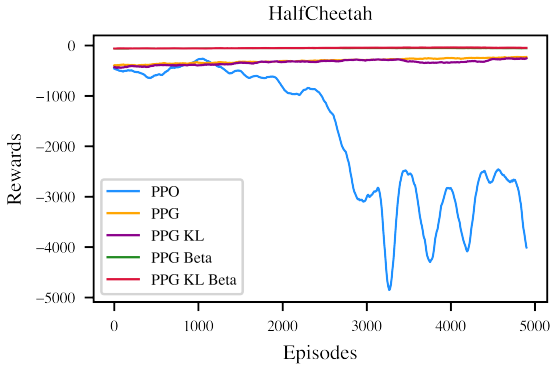
Hsu et al. (2020) identify three common failures modes of PPO[2]. Firstly, the combination of Gaussian policy parameterization and clipping can lead to unstable PPO when rewards vanish outside bounded support. Secondly, standard softmax policy parameterization can lead to standard PPO with clipping converging to suboptimal actions in high-dimensional disrecte action spaces. Lastly, vanilla PPO is sensitive to initialization and can get stuck in suboptimal actions when locally optimal actions are close to initialization. To prevent the first and last failure modes that are associated with a Gaussian policy, a Beta distribution can be implemented instead. The Beta distribution is defined on a bounded interval, and has two parameters α and β. To prevent the first and second failure modes, a KL-regularized surrogate objective can be used instead of the normal clipped surrogate objective:

$$L^{\text{KL, reverse}} = \mathbb{E}\left[\rho_t(\theta) \cdot A_t - c_1 L^{VF}(w) + c_2 \mathcal{H}[\pi_\theta] - \beta_{KL} \cdot D_{KL}(\pi_\theta || \pi_{\theta old})\right] \tag{6}$$

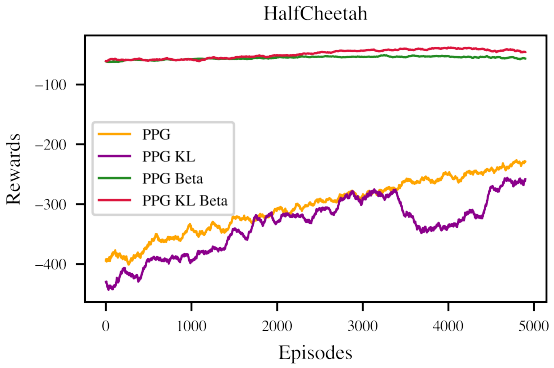
It does not use a clipped loss but it uses a soft constraint on the reversed KL distance between the initial and the updated policy for regularization, making it closely related to TRPO.

3 Experimental Evaluation

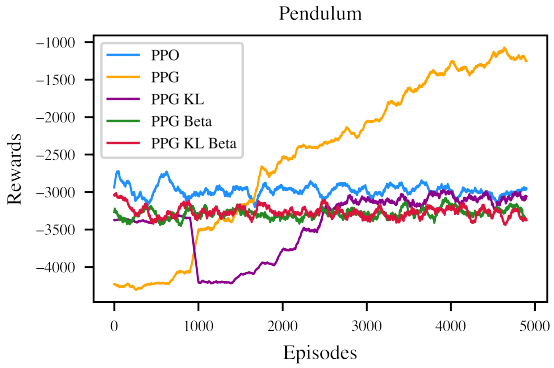
All analyses were conducted in python and can be found in my GitHub repository[2]. In order to compare the different proposed modifications, the performance of five successively optimized models will be analyzed (Appendix A, Table 2): vanilla PPO, PPG, PPG with a KL-regularized objective, PPG with a Beta policy parametrization and PPG with both a KL regularization and Beta distribution.



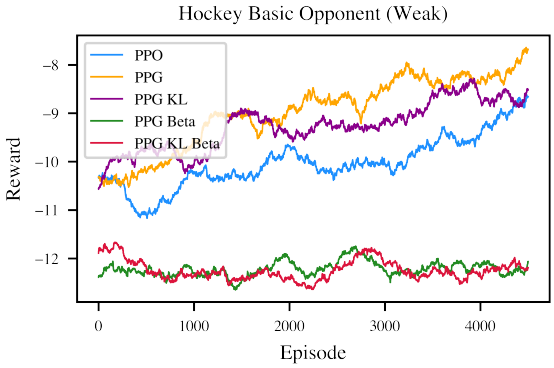
(a) Comparison of models in HalfCheetah environment: all models



(b) Comparison of models in HalfCheetah environment: excluding PPO



(c) Comparison of models in Pendulum environment



(d) Comparison of models in hockey environment

Figure 1: Training comparisons in different environments

3.1 Comparison of Models

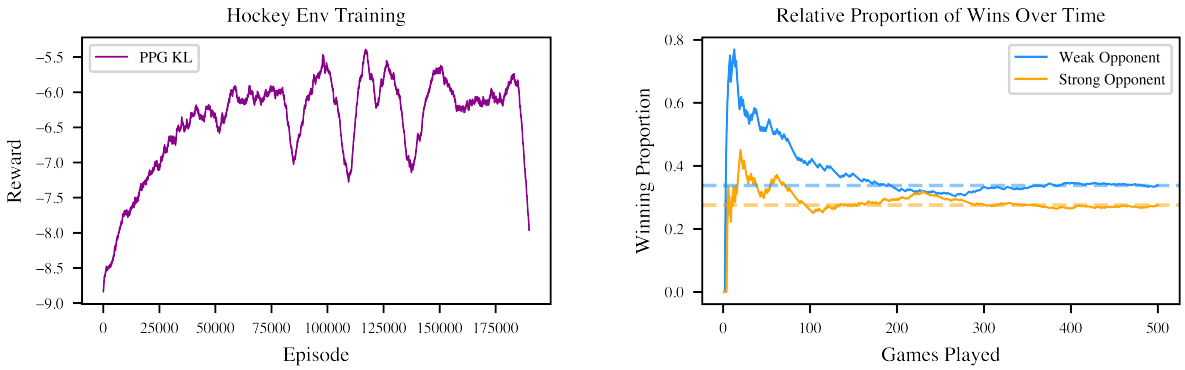
The performance of all five models is tested in the Pendulum, HalfCheetah and hockey environment using the weak opponent. All models are trained for 5000 episodes. As seen in Figure 1a for the HalfCheetah environment, vanilla PPO shows a high training uncertainty and an unstable training process with decreasing and oscillating rewards. Figure 1b illustrates that, when examining the modified PPG models exclusively, both models employing a Beta distribution as policy parametrization demonstrate a smooth training process and better initialization but nearly constant average rewards and minimal training. PPG and PPG with a KL distribution show a relatively stable training process. Results for the Pendulum environment can be found in Figure 1c. In this setting, PPG is the only model showing

[2]https://github.com/kimmizi/PuckAI

sensible training behavior. Models with a Beta distribution, KL divergence or PPO show stagnating training behavior.
Figure 1d shows for the hockey environment using the weak basic opponent that the Beta parametrized models again exhibit poor training behavior, with lower almost constant rewards, no learning, and an average reward of -12. PPO, PPG and PPG with KL regularization show a more sensible training behavior with increasing average rewards over episodes. PPG and PPG with a KL regularization, both having an average reward of -9, perform slightly better than PPO, which has an average of -10. Based on these results, PPG models using an additional auxiliary loss outperform PPO in most settings. Beta parametrized models show a stable but very slow or stagnating learning behavior.

3.2 Hockey Environment: Basic Opponent (Weak and Strong)

The final model is a PPO model with an auxiliary phase (PPG), as well as KL-regularized instead of a clipped objective, Normal policy parametrization and tuned parameters. Monte Carlo estimates are used for calculating the rewards and Generalized Advantage Estimation (GAE)[5] for the advantages. The final model is trained on 200,000 episodes using randomly the weak or strong opponent in the training process. The final model is evaluated in the hockey environment, using the win rate and the average reward of 500 games played against each of the weak and strong basic opponent.



(a) Average rewards over episodes

(b) Win rate (weak and strong opponent)

Figure 2: Average rewards and win rate of final model

Win rate	0.34
Draw rate	0.39
Loss rate	0.28
Average reward	-3.0

(a) Weak basic opponent

Win rate	0.28
Draw rate	0.34
Loss rate	0.38
Average reward	-6.0

(b) Strong basic opponent

Table 1: Evaluation game statistics

Figure 2a illustrates the average rewards over the training episodes. Initially, PPG with KL divergence demonstrates a steady increase in the average reward from approximately -9 to approximately -6. However, after that the average reward oscillates and does not increase anymore. It never exceeds an average reward of -5 until it finally decreases again. As demonstrated in Figure 2b, the win rate is higher for the weak than for the strong opponent, which can also be observed in Table 1. Although the win rate

against the weak opponent is higher than the loss rate, the agent is not able to win consistently against the weak opponent. The average reward of all games is negative, which shows that the agent has not learned how to handle or shoot the puck effectively. In conclusion, the performance of the final model is unsatisfactory. The model shows instability in its training process and a low percentage of games won.

4 Discussion

4.1 Modifications to PPO

The experimental results show that the different modifications to PPO reveal distinct performances and learning dynamics. While vanilla PPO may exhibit instability or stagnation in the training process, PPG shows generally a smoother learning curve and more robust training behavior. This suggests that implementing auxiliary phases introduces more stability and additional structure to the training.
Furthermore, replacing the clipped surrogate objective with a KL-regularized one may introduce some additional instability into the training process. However, PPG with KL objective performs as well as normal PPG in most of the tested environments. This modification helps to prevent convergence to suboptimal actions, and thereby improving performance. Therefore, using a KL divergence, as proposed by Hsu et al. (2020), can be a promising alternative to the standard clipped objective.
Models using a Beta distribution instead of a Normal distribution for policy optimization generally show significantly worse training behavior. However, PPG with a Beta distribution also demonstrates two promising advantages: more training stability, as shown by smoother learning curves, and partially better initialization, as seen in the HalfCheetah environment. However, learning is very slow, yielding in nearly constant rewards. In conclusion, using a Beta policy parametrization as proposed by Hsu et al. (2020) does not lead to better performances in our cases. The specific design choices of PPO seem to be dependent on the chosen environment and should be tested in practice.

4.2 Limitations

Despite various modifications to PPO, our final model exhibits poor performance in the hockey environment. Several limitations could have contributed to this unsatisfactory result. Firstly, our model was trained on 200,000 episodes, which might not have been sufficient for training an agent that is able to handle the hockey environment. Secondly, the performance of PPO and PPG is sensitive to hyperparameter tuning. Hyperparameters might not have been optimally selected, leading to suboptimal results. In addition to that, the hockey environment is a complex, multi-agent environment. While PPO is a state of the art RL method, PPG with a KL regularization might not have been the best choice in this setting. Finally, the final model shows oscillations in reward behavior, suggesting an imbalance between exploration and exploitation. More advanced exploration strategies, such as intrinsic motivation[8] or curiosity-driven learning[4], could solve this issue. Future work might focus on improving the training behavior by using a more enhanced model, more training episodes, and more efficient exploration strategies.

4.3 Conclusion

This study investigated various modifications to PPO, including PPG, KL- regularized objectives, and Beta distribution policy parametrization. Models were tested and evaluated in both simple environments and a complex hockey environment. While some modifications like additional auxiliary phases and KL-regularized objectives provided stability and improved convergence, the final model exhibited poor performance in the hockey environment, with a low win rate and unstable rewards.

References

[1] K. Cobbe, J. Hilton, O. Klimov, and J. Schulman. Phasic policy gradient, 2020.

[2] C. C.-Y. Hsu, C. Mendler-Dünner, and M. Hardt. Revisiting design choices in proximal policy optimization, 2020.

[3] E. F. Morales and H. J. Escalante. Chapter 6 - a brief introduction to supervised, unsupervised, and reinforcement learning. In A. A. Torres-García, C. A. Reyes-García, L. Villaseñor-Pineda, and O. Mendoza-Montoya, editors, *Biosignal Processing and Classification Using Computational Learning and Intelligence*, pages 111–129. Academic Press, 2022.

[4] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction, 2017.

[5] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018.

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[7] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. D. Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, H. Tan, and O. G. Younis. Gymnasium: A standard interface for reinforcement learning environments, 2024.

[8] Z. Yang, T. M. Moerland, M. Preuss, and A. Plaat. First go, then post-explore: the benefits of post-exploration in intrinsic motivation, 2023.

A Overview of all PPO models

Model	**Training phases**	**Surrogate objective**	**Policy parameterization**
Vanilla PPO	Policy phase only	Clipped surrogate objective	Normal distribution
PPG	Policy phase and auxiliary phase	Clipped surrogate objective, behavioral cloning loss, and arbitrary auxiliary loss	Normal distribution
PPG KL	Policy phase and auxiliary phase	Reversed KL divergence, behavioral cloning loss, and arbitrary auxiliary loss	Normal distribution
PPG Beta	Policy phase and auxiliary phase	Clipped surrogate objective, behavioral cloning loss, and arbitrary auxiliary loss	Beta distribution
PPG KL Beta	Policy phase and auxiliary phase	Reversed KL divergence, behavioral cloning loss, and arbitrary auxiliary loss	Beta distribution

Table 2: Overview of PPO models and their modifications