Lab report - Mini project 2

In this project, we implemented the DDPG algorithm to investigate the effect of partial observability on learning performance in the `CartPoleContinuous-v1` environment. Specifically, we selectively hid one or both velocity-related features in agent's state, namely the velocity $\dot{x}$ and the angular velocity $\dot{\theta}$, by wrapping the environment with the `FeatureFilterWrapper`. We then explored different ways to mitigate this limitation by coding the `ObsTimeExtensionWrapper` to equip the agent with a memory of the past and the `ActionTimeExtensionWrapper` to extend the action space, allowing the critic to consider more complex actions.

1 Setup

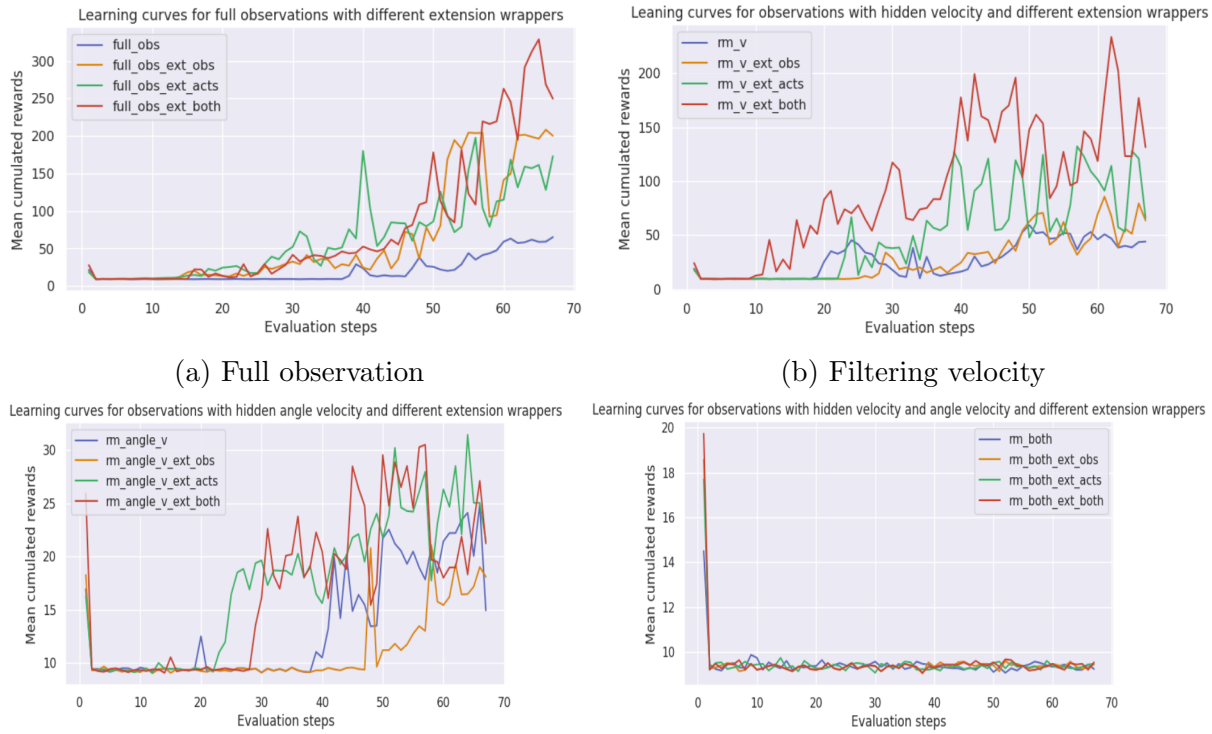
- The `CartPoleContinuous-v1` environment is a customized version of the standard `Gymnasium`'s Cartpole with continuous actions, defined in the `bbrl_gymnasium` package. When studying compensation for a partially observable environment in Section 2, we extend the observation space by adding only a memory of size 1 and expand the action space to size 2. We will further study the impact of larger temporal horizons for action sequences on learning performance in Section 3.
- We implemented the DDPG algorithm using the suggested configuration, with some important hyperparamenters listed as follows:
  - The training lasts for a maximum of 1500 epochs, and at each epoch, there are 100 transitions collected into a replay buffer by one agent. The training starts after the agent collects enough 10,000 transitions. Moreover, a Gaussian noise with a standard deviation of 0.1, is applied to encourage exploration during training.
  - Both the actor and critic networks have two hidden layers with sizes of [400, 300], and the Adam optimizer is used to update their weights, with a learning rate of 10^{-3} and a batch size of 64. The target critic is updated with each update of the learned critic with a coefficient $\tau = 0.05$.

2 Results

In the following, we present four sets of learning curves obtained for four different combinations of observation filtering: full observation (no filtering), filtering out the velocity feature, filtering out the angular velocity feature, and filtering both features. In each plot, we show the baseline curve for each case when no temporal extensions are applied, as well as additional curves when applying the `ObsTimeExtensionWrapper`, the `ActionTimeExtensionWrapper`, and both wrappers together.

The statistics used to compare performance is the cumulative rewards, averaged over eight evaluation agents in three different runs, each with three different seeds (this statistics is saved in the `eval_rewards` attribute of the `DDPG` class; please refer the source code for more information).
We also want to note that in fact, we evaluate the training every 2100 transitions, starting from the 10,3k-th step and ending at 148,9k-th step of the training process, resulting in a total of 67 evaluations. The order of the evaluation steps as shown the x axis of each plot is relative and does not reflect the actual step at which the model is evaluated in reality.

Please note that the scales of y-axis in the following figures are not the same.



(a) Full observation

(b) Filtering velocity

(c) Filtering angular velocity

(d) Filtering both velocity and angular velocity

Figure 1: Learning curves for different numbers of observable state features

The following conclusions are drawn from the resulting plots that we obtained when running the algorithm with three different seeds. They may not accurately reflect the reality, as three runs might not be sufficient to capture the true performance.

2.1 Effect of Partial Observability

In general, learning is less effective when we do not have full access to the observation features. Moreover, we can see that removing velocity slightly hinders training, but removing

angular velocity causes a more significant decline in learning efficiency. This difference can be explained by the fact that the angular velocity is more critical for balancing the pole in the task. Without it, the agent struggles to maintain balance, which impacts learning more severely compared to missing velocity information.

2.2 Improvement with Extensions

In general, adding extensions enhances learning performance, except when angular velocity is removed and only the observation space is extended. It can be obviously seen that using both extensions provides a greater improvement in training than using just one. While it might seem intuitive that extending the observation space - which includes past memories - would yield better results than extending the action space, which does not contain any past memories, the graphs reveal the opposite in this case.

When velocity is removed, extensions are not only able to compensate the limitation but also lead to better learning outcomes (especially when both observation and action spaces are extended), demonstrating that the agent adapts well despite the missing velocity feature. When angular velocity is removed, extensions improve learning but still cannot match the performance achieved with the full observation space. This reinforces the idea that angular velocity is crucial for solving the task effectively. In the case where both velocity features are removed, extensions are insufficient to recover performance, indicating that the agent struggles significantly with the absence of both features.

3 Extension: Different horizons for action sequences

In this section, we examine the impact of different temporal horizons for action sequences. We consider the standard `CartPoleContinuous-v1` environment with the action space extended to different sizes, namely 2, 3 and 4. We obtained the following results:

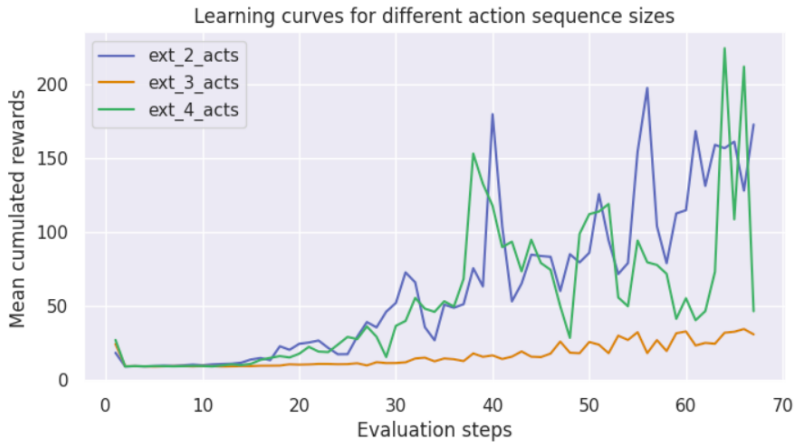


Figure 2: Learning curves for different value of hyperparameter M

It can be observed that the performances for $M = 2$ and $M = 4$ are relatively similar, while that for $M = 3$ is significantly worse. This indicates that training efficiency can vary with different values of M and a higher M does not always guarantee better results. This can be explained by the fact that we do not extend the agent's actions with past actions. Instead, we expand the action space to provide the critic network with more information for evaluation, while in practice, only the first action in the sequence is played.

Finally, we sincerely welcome any comments or suggestions from our readers regarding this report, as your feedback is invaluable in helping us improve and refine our work.