

1 Experiments and Applications

1.1 Toy Task: The Rooms Problem

To provide an illustration of HRL in action, we apply the subgoal discovery algorithm to the *Rooms problem* introduced by (Sutton and Barto, 1998). Given initial state S and goal state G , the agent’s task is to navigate through a set of rooms interconnected by doorways, in order to reach the goal state G (Figure 1). In each state, the agent can select any of eight deterministic primitive actions, each of which moves the agent to one of the adjacent squares (unless a wall prevents this movement). The rooms are reachable through *doors* and agent is only allowed to use these doors to reach the other room. Agent receives reward of -1 for each action, -2 for actions that results bumping to the walls and 0 when reaching to G .

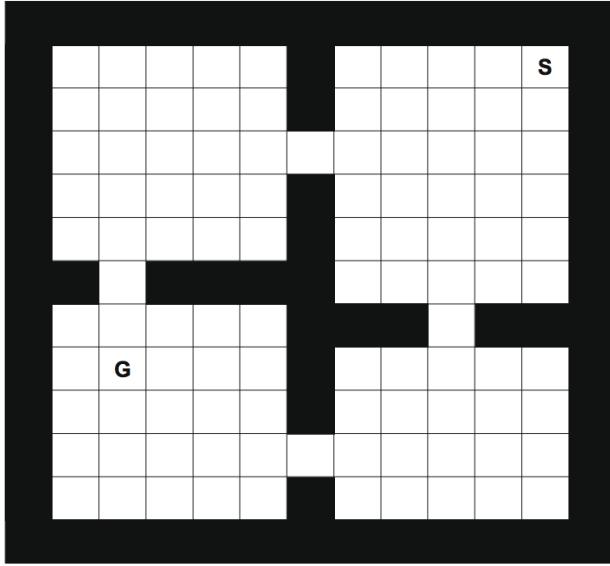


Figure 1: The rooms problem, adapted from (Botvinick et al., 2009)

1.2 ATARI Games: Montezuma’s Revenge

The Arcade Learning Environment (ALE) is an evaluation platform that provides an interface to hundreds of Atari 2600 game environments. It supports a variety of different problem settings and it has been receiving increasing attention from the scientific community, leading to some high-profile success stories such as the much publicized Deep Q-Networks (DQN) (Mnih et al., 2013; Bellemare et al., 2013). Montezuma’s revenge is one of the hardest games available through ALE. The game is infamous for challenging agents with lethal traps and sparse rewards (Vezhnevets et al., 2017). The game (Figure 2) requires the player to navigate the explorer (in red) through several rooms while collecting treasures. In order to pass through doors (in the top right and top left corners of the figure), the player has to first pick up the key (Kulkarni et al., 2016). The hierarchical nature of this game made it a common experiment for testing hierarchical reinforcement learning algorithm (Kulkarni et al., 2016; Vezhnevets et al., 2017).



Figure 2: A sample screen from the ATARI 2600 game called “Montezuma’s Revenge” adopted from (Kulkarni et al., 2016)

1.3 Path Planning for Autonomous Robot Navigation

Navigation is one of the most important robotic tasks. One classic approach consists of three steps: build a map of the environment through, e.g., Simultaneous Localization and Mapping (SLAM), plan a path using the map, and finally execute the path. High-fidelity geometric maps make it easier for path planning and execution. However, building such maps is time-consuming (Gao et al., 2017). Further, even small environmental changes may render them partially invalid. Alternatively, the optical flow approach does not use maps at all and relies on the visual perception of the local environment for navigation. While this avoids the difficulty of building and maintaining accurate geometric maps, it is difficult to achieve effective goal-directed global navigation in complex geometric environments without maps. Our approach uses 2-D floor plans available for indoor environments. We use robotics simulator such as Gebezo and Robot Operating System (ROS) to simulate environments for robotics navigation tasks.

1.4 Bipedal Locomotion and Reaching Tasks

Learning physics-based locomotion skills is a difficult problem. With this simulation we aim to learn a variety of environment-aware locomotion skills with a limited amount of prior knowledge. We investigate the task of reaching and also locomotion using two-level hierarchical control framework. Results are demonstrated on a simulated 3D biped (Peng et al., 2017; Geijtenbeek et al., 2013). For animating virtual characters, physics based simulation is used by (Geijtenbeek et al., 2013) (Figure 3). The subgoal discovery algorithm can be applied to the experience space of locomotion task and learning to reach for these animated creatures in hierarchical reinforcement learning framework.

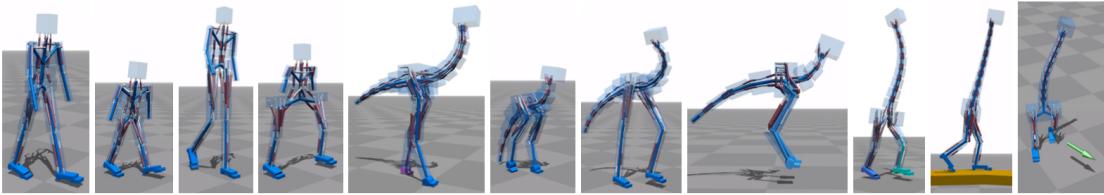


Figure 3: Physics-based simulation of locomotion for a variety of creatures driven by 3D muscle-based control adopted from (Geijtenbeek et al., 2013)

1.5 Autonoumous Driving

The capability of autonomous robotic systems to successfully navigate through difficult environments continues to advance (Silver et al., 2010; El Sallab et al., 2017; Dosovitskiy et al., 2017). Driving is usually a navigation task from an initial state to a destination (goal) given the map of the environment. One the most difficult form is navigation in densely populated urban environments due to complex multi-agent dynamics at traffic intersections; the necessity to track and respond to the motion of tens or hundreds of other actors (Dosovitskiy et al., 2017). Training and validation of sensorimotor control models for urban driving in the physical world is beyond the reach of most research groups. An alternative is to train and validate driving strategies in simulation. There are several open source driving simulators that we will use to test our sub-goal discovery and hierarchical reinforcement learning algorithms in order to approach the complex problem of autonomous driving.

TORCS is a highly portable multi platform car racing simulation. It is used as ordinary car racing game, as AI racing game and as research platform (Bernhard Wymann, 2014) (Figure 4).



Figure 4: Example shots from (Bernhard Wymann, 2014). Left: A third-person view. Right: Perception from the driver's perspective shielded.

CARLA is another open-source simulator for autonomous driving research (Dosovitskiy et al., 2017). CARLA has been built for flexibility and realism in the rendering and physics simulation. It is implemented as an open-source layer over Unreal Engine 4 (UE4). CARLA has been developed from the ground up to support development, training, and validation of autonomous urban driving systems. The simulation platform supports flexible specification of sensor suites and environmental conditions. We use CARLA to study the performance of subgoal discovery algorithms for hierarchical reinforcement learning applied to autonomous driving. We use CARLA to stage controlled goal-directed navigation scenarios of increasing difficulty. We manipulate the complexity of the route that must be traversed, the presence of traffic, and the environmental conditions (Figures 5 and 6).

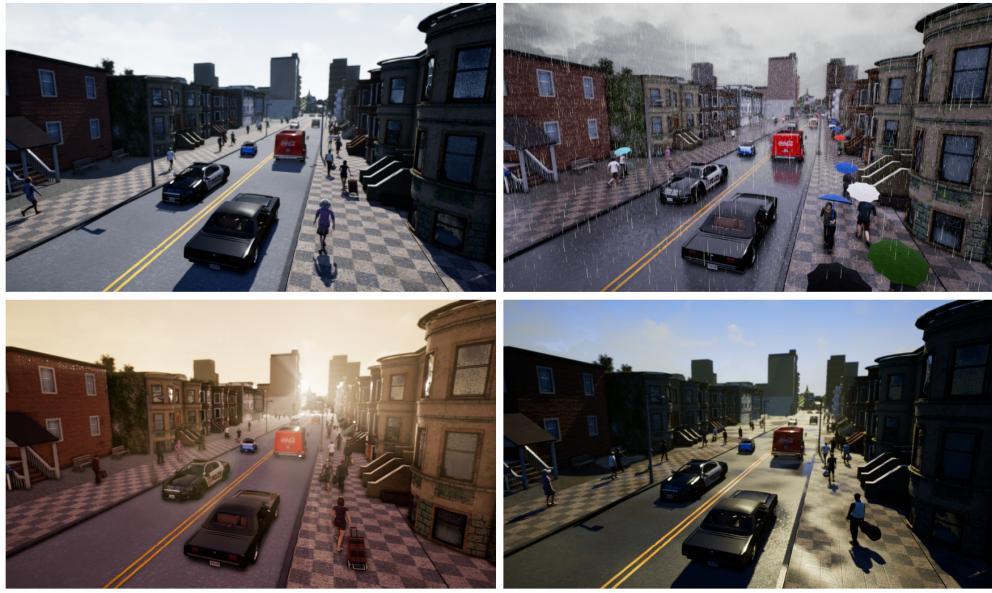


Figure 5: A third-person view in four weather conditions. Clockwise from top left: clear day, daytime rain, daytime shortly after rain, and clear sunset adopted from (Dosovitskiy et al., 2017)

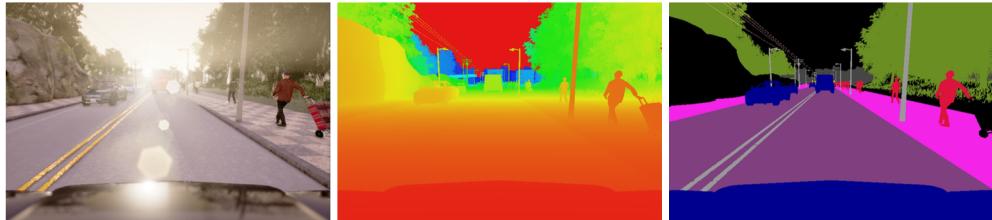


Figure 6: Three of the sensing modalities provided by CARLA. From left to right: normal vision camera, ground-truth depth, and ground-truth semantic segmentation. Depth and semantic segmentation are pseudo-sensors that support experiments that control for the role of perception. Additional sensor models can be plugged in via the API adopted from (Dosovitskiy et al., 2017)

References

- Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.
- Bernhard Wymann, Eric Espié, C. G. C. D. R. C. A. S. (2014). TORCS, The Open Racing Car Simulator. <http://www.torcs.org>.
- Botvinick, M. M., Niv, Y., and Barto, A. C. (2009). Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 113(3):262 – 280.
- Dosovitskiy, A., Ros, G., Codevilla, F., López, A., and Koltun, V. (2017). CARLA: an open urban driving simulator. *CoRR*, abs/1711.03938.
- El Sallab, A., Abdou, M., Perot, E., and Yogamani, S. (2017). Deep Reinforcement Learning framework for Autonomous Driving. *ArXiv e-prints*.
- Gao, W., Hsu, D., Lee, W. S., Shen, S., and Subramanian, K. (2017). Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation. *CoRR*, abs/1710.05627.
- Geijtenbeek, T., van de Panne, M., and van der Stappen, A. F. (2013). Flexible muscle-based locomotion for bipedal creatures. *ACM Trans. Graph.*, 32(6):206:1–206:11.
- Kulkarni, T. D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. B. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *CoRR*, abs/1604.06057.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. (2013). Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602.
- Peng, X. B., Berseth, G., Yin, K., and Van De Panne, M. (2017). Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.*, 36(4):41:1–41:13.
- Silver, D., Bagnell, J. A., and Stentz, A. (2010). Learning from demonstration for autonomous navigation in complex unstructured terrain. *Int. J. Rob. Res.*, 29(12):1565–1592.
- Sutton, R. S. and Barto, A. G. (1998). *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition.
- Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. (2017). Feudal networks for hierarchical reinforcement learning. *CoRR*, abs/1703.01161.