Hamza Emin Hacıoğlu 2786978
Burak Muammer Yıldız 2529451

Unsupervised Hierarchical Skill Discovery

The goal of this project is to investigate unsupervised skill discovery in hierarchical reinforcement learning by training agents to learn diverse and reusable behaviors without external rewards. Using **DIAYN** and **VALOR** as core methods, the project aims to examine how distinct skills emerge in environments such as **MiniGrid** and **DMControl**, and to characterize the structure, variability, and consistency of these skills. The broader objective is to understand the properties of these unsupervised skills and evaluate their potential usefulness in later tasks that require structured or temporally extended behaviors.

ALGORITHMS

- **DIAYN (Diversity Is All You Need)**: Learns distinct skills by maximizing mutual information between states and a latent skill variable. A discriminator predicts the skill from states, guiding a latent-conditioned policy trained via Soft Actor-Critic.

- **VALOR**: Infers latent options from short trajectory segments using a variational objective, producing temporally consistent behaviors (e.g., specific movement patterns).

- **Baseline**: Random-skill policy, sampling a latent code at each episode to compare learned skills against untrained behavior

ENVIRONMENTS

- **DMControl**: Continuous control tasks (e.g., Cheetah, Walker) for testing diverse locomotion and stable continuous behaviors.
- **MiniGrid**: Grid-based, discrete environments (e.g., MiniGrid-Empty, MiniGrid-FourRooms) to observe navigation and exploration behaviors.

METRICS

- **Skill Discriminability** – Accuracy of the discriminator in identifying skills from states or trajectories. Higher accuracy indicates clearer skill separation.

- **Skill analysis (qualitative):** visualize rollouts of each skill in an empty environment, and plot skill usage frequencies per task.

- **Mutual Information (I(S; Z))** – Estimates how strongly skills influence visited states; higher values indicate more structured skills.

- **Temporal Consistency** – Checks if skills produce coherent, consistent behavior over time.

- **Forward transfer:** learning curves (return vs. steps) and sample complexity to reach a fixed performance threshold on each task..

RESOURCES

- **Hardware**: 1× GPU (5080)

- **Software**: Python 3.10, PyTorch, Gymnasium + MiniGrid, dm_control