Appendix

Table 1. Hyperparameters for CERL

Hyperparameter	Value
Population size k	10
Roll-out size b	10
Target weight τ	$5e^{-3}$
Actor Learning Rate	$1e^{-3}$
Critic Learning Rate	$1e^{-3}$
Replay Buffer	$1e^{6}$
Batch Size	256
Mutation Probability mut_{prob}	0.9
Mutation Fraction mut_{frac}	0.1
Mutation Strength $mut_{strength}$	0.1
Super Mutation Probability $supermut_{prob}$	0.05
Reset Mutation Probability $resetmut_{prob}$	0.05
Number of elites e	$k/5$
Lamarckian Transfer Period ω	5
Value Learning α	0.2
Normalized Observation	None
Gradient Clipping	None
Exploration Policy	$\mathcal{N}(0, \sigma)$
Exploration Noise σ	0.1

This section details the hyperparameters used for Collaborative Evolutionary Reinforcement Learning (CERL) across all benchmarks.

- **Optimizer = Adam**
  Adam optimizer was used to update both the actor and critic networks for all learners.
- **Population size k = 10**
  This parameter controls the number of different individual actors (policies) that are present in the evolutionary population.
- **Roll-out size b = 10**
  This parameter controls the number of roll-out workers (each running an episode of the task) that compose the computational resource available to the resource-manager.

  Note: The two parameters above (population size k and roll-out size b) collectively modulates the proportion of exploration carried out through noise in the actor's *parameter* space and its *action* space.
- **Target weight $\tau = 5e^{-3}$**
  This parameter controls the magnitude of the soft update between the actors and critic networks, and their target counterparts.
- **Actor Learning Rate = $1e^{-3}$**
  This parameter controls the learning rate of the actor network.
- **Critic Learning Rate = $1e^{-3}$**
  This parameter controls the learning rate of the critic network.
- **Replay Buffer Size = $1e^{6}$**
  This parameter controls the size of the replay buffer. After the buffer is filled, the oldest experiences are deleted in order to make room for new ones.
- **Batch Size = 256**
  This parameters controls the batch size used to compute the gradients.
- **Actor Neural Architecture = $[400, 300]$**
  The actor network consists of two hidden layers, each with 400 and 300 nodes, respectively. Exponential Linear Units (ELU) was used as the activation function. Layer normalization was used before each layer.
- **Critic Neural Architecture = $[400, 300]$**
  The actor network consists of two hidden layers, each with 400 and 300 nodes, respectively. Exponential Linear Units (ELU) was used as the activation function. Layer normalization was used before each layer.
- **Number of Elites $e = k/5$**
  The number of elites was set to be 20% of the population size (k). This parameter controls the fraction of the population that are categorized as elites. Since an elite individual (actor) is shielded from the mutation step and preserved as it is, the elite fraction modulates the degree of exploration/exploitation within the evolutionary population. In general, tasks with more stochastic dynamics (correlating with more contact points) have a higher variance in fitness values. A higher elite fraction in these tasks helps in reducing the probability of losing good actors due to high variance in fitness, promoting stable learning.
- **Mutation Probability $mut_{prob} = 0.9$**
  This parameter represents the probability that an actor goes through a mutation operation between generation.
- **Mutation Fraction $mut_{frac} = 0.1$**
  This parameter controls the fraction of the weights in a chosen actor (neural network) that are mutated, once the actor is chosen for mutation.
- **Mutation Strength $mut_{strength} = 0.1$**
  This parameter controls the standard deviation of the Gaussian operation that comprises mutation.
- **Super Mutation Probability $supermut_{prob} = 0.05$**
  This parameter controls the probability that a super mutation (larger mutation) happens in place of a standard mutation.

- **Reset Mutation Probability** $resetmut_{prob} = 0.05$
  This parameter controls the probability a neural weight is instead reset between $\mathcal{N}(0, 1)$ rather than being mutated.

- **Value learning rate** $\alpha = 0.2$
  This parameter controls the learning rate used to update the value of a learner after receiving a fitness value for its roll-out.

- **Lamarckian Transfer Period** $\omega = 5$
  This parameter controls the frequency of information flow between the portfolio of learners and the evolutionary population. A higher ω generally allows more time for expansive exploration by the evolutionary population while a lower ω can allow for a more narrower search. The parameter controls how frequently the exploration in action space (portfolio of gradient-based learners) shares information with the exploration in the parameter space (actors in the evolutionary population).

- **Exploration Noise** $\sigma = 0.1$
  This parameter controls the standard deviation of the Gaussian operation that comprise the noise added to the actor's actions during exploration by the learners (learner roll-outs).