A. Motivation and Goals

- Distill knowledge from locally well-behaved agents into a single globally well-behaved agent.
- Start with a population of agents and gradually merge policies over rounds of a genetically-inspired iterative algorithm.

B. GPO Algorithm

1. population ← \( \pi_1, \ldots, \pi_n \)
2. repeat
3. population ← MUTATE(population)
4. parents_set ← SELECT(population, fitness)
5. children ← empty set
6. for tuple(\( \pi_i, \pi_j \)) in parents_set do
7. \( \pi_i \) ← CROSSOVER(\( \pi_i, \pi_j \))
8. add \( \pi_i \) to children
9. end for
10. population ← children
11. until \( k \) steps of genetic optimization

C. Crossover Operator

\[
\begin{align*}
\pi_{\text{child}}(a) &= \pi_{\text{parent}}(a) \cdot \pi_{\text{child}}(a) + \pi_{\text{parent}}(a) \cdot \pi_{\text{child}}(a) \\
&= \pi_{\text{parent}}(a) \cdot \left( 1 - \pi_{\text{child}}(a) \right) + \pi_{\text{child}}(a) \cdot \left( 1 - \pi_{\text{parent}}(a) \right)
\end{align*}
\]

D. Contrast with Parameter Crossover

- MUTATE perturbs the parameters of the neural network policy. Instead of random perturbations, we use standard policy-gradient algorithms (PPO, A2C) to move the parameters in the direction of the noisy gradients approximated from sampled trajectories.
- Data Sharing: When mutating multiple policies in parallel, a policy \( \pi_i \) can also use data samples from other similar policies for off-policy learning. For example, with the PPO objective, the modified gradient for \( \pi_i \) is

\[
\nabla_{\theta_i} L_{\text{PPO}}(\theta_i) = \mathbb{E}_{x \sim \tau}[\nabla_{\theta_i} \log(\pi_{\text{child}}(a|s)) R(x)] - \mathbb{E}_{x \sim \tau}[KL(\pi_{\text{child}}(a|s), \pi_{\text{parent}}(a|s))]
\]

where \( \tau \equiv \{ j | KL(\pi_i, \pi_j) < \epsilon \} \) before the start of current round of mutation contains similar policies to \( \pi_i \) (including \( \pi_i \)).

SELECT chooses policies-pairs \( \{ \pi_x, \pi_y \} \) with high fitness for the crossover step. Different fitness functions are possible:

- Performance fitness as sum of expected returns of both policies, i.e.
  \[
  f(\pi_x, \pi_y) \equiv E_{\tau \sim \pi_x} [R(\tau)] + E_{\tau \sim \pi_y} [R(\tau)]
  \]
- Diversity fitness as KL-divergence between policies, i.e.
  \[
  f(\pi_x, \pi_y) \equiv KL(\pi_x, \pi_y)
  \]

E. MUTATE and SELECT Operators

- E. MUTATE and SELECT Operators

\[
\begin{align*}
\nabla_{\theta_i} L_{\text{PPO}}(\theta_i) &= \mathbb{E}_{x \sim \tau}[\nabla_{\theta_i} \log(\pi_{\text{child}}(a|s)) R(x)] - \mathbb{E}_{x \sim \tau}[KL(\pi_{\text{child}}(a|s), \pi_{\text{parent}}(a|s))]
\end{align*}
\]

- E. MUTATE and SELECT Operators

\[
\begin{align*}
\nabla_{\theta_i} L_{\text{PPO}}(\theta_i) &= \mathbb{E}_{x \sim \tau}[\nabla_{\theta_i} \log(\pi_{\text{child}}(a|s)) R(x)] - \mathbb{E}_{x \sim \tau}[KL(\pi_{\text{child}}(a|s), \pi_{\text{parent}}(a|s))]
\end{align*}
\]

\[
\begin{align*}
\nabla_{\theta_i} L_{\text{PPO}}(\theta_i) &= \mathbb{E}_{x \sim \tau}[\nabla_{\theta_i} \log(\pi_{\text{child}}(a|s)) R(x)] - \mathbb{E}_{x \sim \tau}[KL(\pi_{\text{child}}(a|s), \pi_{\text{parent}}(a|s))]
\end{align*}
\]

- E. MUTATE and SELECT Operators

\[
\begin{align*}
\nabla_{\theta_i} L_{\text{PPO}}(\theta_i) &= \mathbb{E}_{x \sim \tau}[\nabla_{\theta_i} \log(\pi_{\text{child}}(a|s)) R(x)] - \mathbb{E}_{x \sim \tau}[KL(\pi_{\text{child}}(a|s), \pi_{\text{parent}}(a|s))]
\end{align*}
\]

- E. MUTATE and SELECT Operators

\[
\begin{align*}
\nabla_{\theta_i} L_{\text{PPO}}(\theta_i) &= \mathbb{E}_{x \sim \tau}[\nabla_{\theta_i} \log(\pi_{\text{child}}(a|s)) R(x)] - \mathbb{E}_{x \sim \tau}[KL(\pi_{\text{child}}(a|s), \pi_{\text{parent}}(a|s))]
\end{align*}
\]