

Main Components:

1. **policy(state, info, eval_mode, params)**

A handcrafted policy that:

- Locates the lane in a 13x13 grid.
- Steers towards the closest lane center.
- Adjusts acceleration based on how straight the road looks.
- If eval_mode=True, loads optimized parameters from cmaes_params.json (currently unused in logic).

2. **fitness(params)**

Runs the policy on **3 tracks**, returns **negative total distance covered** as the fitness value (CMA-ES minimizes this, hence the negative).

3. **call_cma()**

Runs CMA-ES for a specified number of generations and population size. Trains the parameters to **maximize total distance covered**.

4. **main block:**

- --train: Runs call_cma() and saves best policy parameters to cmaes_params.json.
- --eval: Loads the trained policy and runs it on all tracks, printing metrics.
- --render: Optionally records videos of the runs.

Notes:

- The policy is **heuristic-based**, with room to actually use params (currently they're loaded in eval but not used).
- cmaes_params.json is the output of training and input for evaluation.