No reward? No problem: Training Maze Agents with Preferences

Giovanni Filomeno
Supervisor: Vihang Patil

Motivation:

Defining reward functions for complex navigation tasks is difficult and often misleading. Preference-based learning as an alternative is explored, highlighting its failure modes and improvements.

Challenge:

Can DPO learn navigation behavior in maze environments? What are its limits, and how can we overcome them?

Key Contributions:

- A custom maze environment for testing preference-based methods
- Identification of a specific failure mode in DPO when preference signals are misaligned
- A scoring function and preference shaping method to recover performance in those cases

Introduction:

- In many real-world settings, designing a dense and reliable reward function is impractical.
- Preference-based learning offers an alternative: agents learn which behaviors are better via pairwise comparisons.
- Direct Preference Optimization (DPO) enables scalable learning from such comparisons, without needing explicit rewards.
- However, DPO can fail in sparse or deceptive environments.
- This failure mode has been studied in a custom maze setting and propose a scoring strategy to make preference-based learning effective even without hand-crafted rewards.

Experimental Setup & Key Insights:

- Environment: A custom 2D maze environment was developed with winding “S-shaped” corridors to expose weaknesses in naïve and greedy strategies.
- Point Generation: Thousands of random states were sampled within the maze to build a diverse dataset for learning preferences.
- Preference Function: Pairwise preference scores were computed using a handcrafted metric combining distance to goal, proximity to walls, presence of dead ends, and actual path length.
- Learning via DPO: A neural network was trained to model preferences by assigning higher scores to better states, effectively shaping a reward-free value function through ranking loss.
- Sensitivity Analysis (Future Work): The final policy was highly sensitive to the choice of scoring function. Minor changes in dead-end penalties or path length weights produced drastically different behaviors, highlighting the importance of careful preference design.

Algorithm: Preference-Based Maze Training

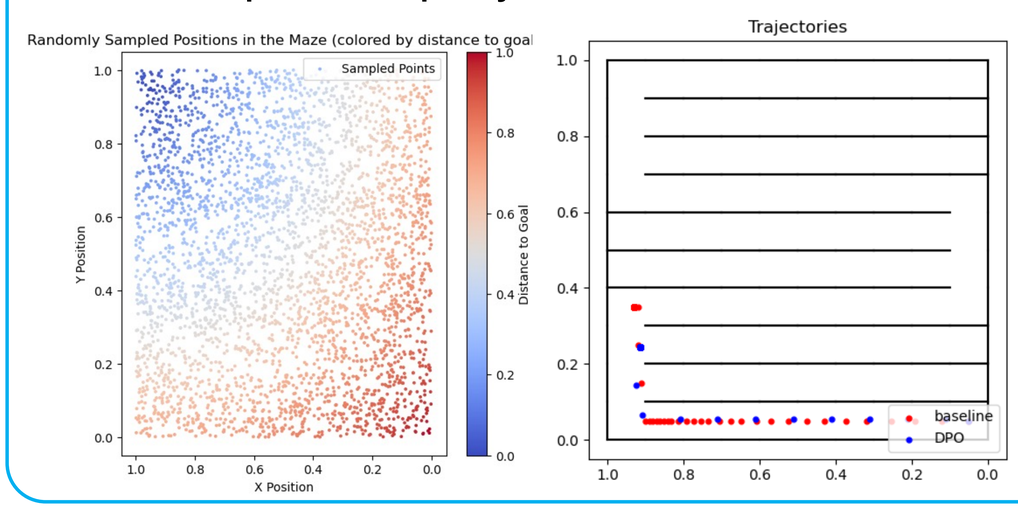
Input: Maze M, number of samples N
Output: Policy $\pi(\text{state}) \approx \nabla f_\theta(\text{state})$

1. Sample N random positions $\{s_1, \ldots, s_n\}$ from M
2. For each pair (s_i, s_j), compute a preference score based on:
   - Distance to goal
   - Proximity to walls
   - Presence of dead ends
3. Train f_θ using DPO:

$$P(s_i \succ s_j) \propto \frac{\exp(f_\theta(s_i))}{\exp(f_\theta(s_i)) + \exp(f_\theta(s_j))}$$

4. Deploy π by following the gradient $\nabla f_\theta(\text{state})$

Calibration of preference: poorly calibrated score leads to fail



Results: DPO can escape from Maze