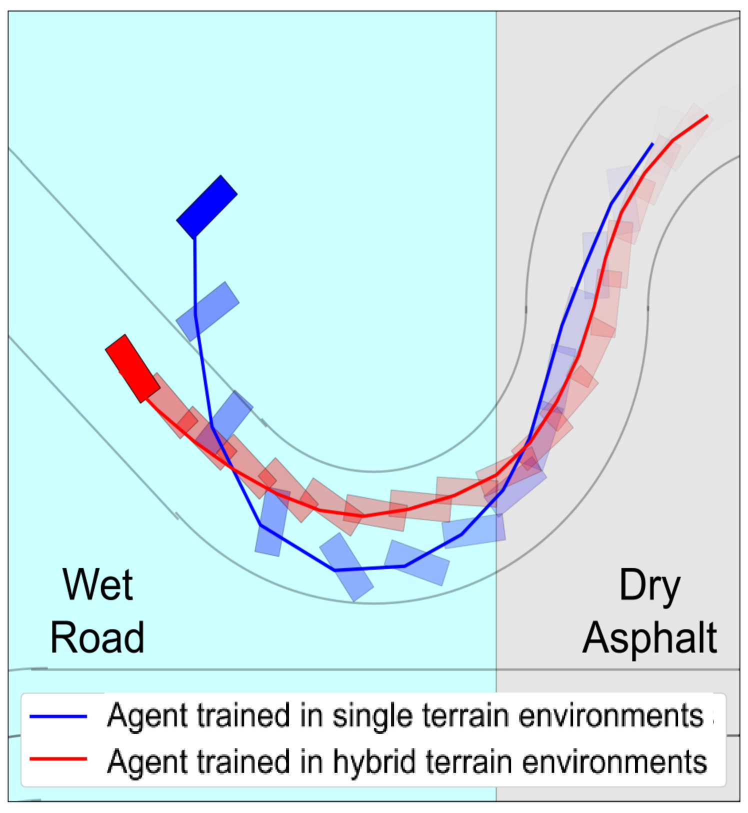
Learning Adaptive Diffusion Policies for Hybrid Dynamical Systems

Leroy D'Souza*, Akash Karthikeyan*, Yash Vardhan Pant, Sebastian Fischmeister
Department of Electrical and Computer Engineering, University of Waterloo

Motivation

Motivating example: Autonomous racing in multi-terrain tracks.



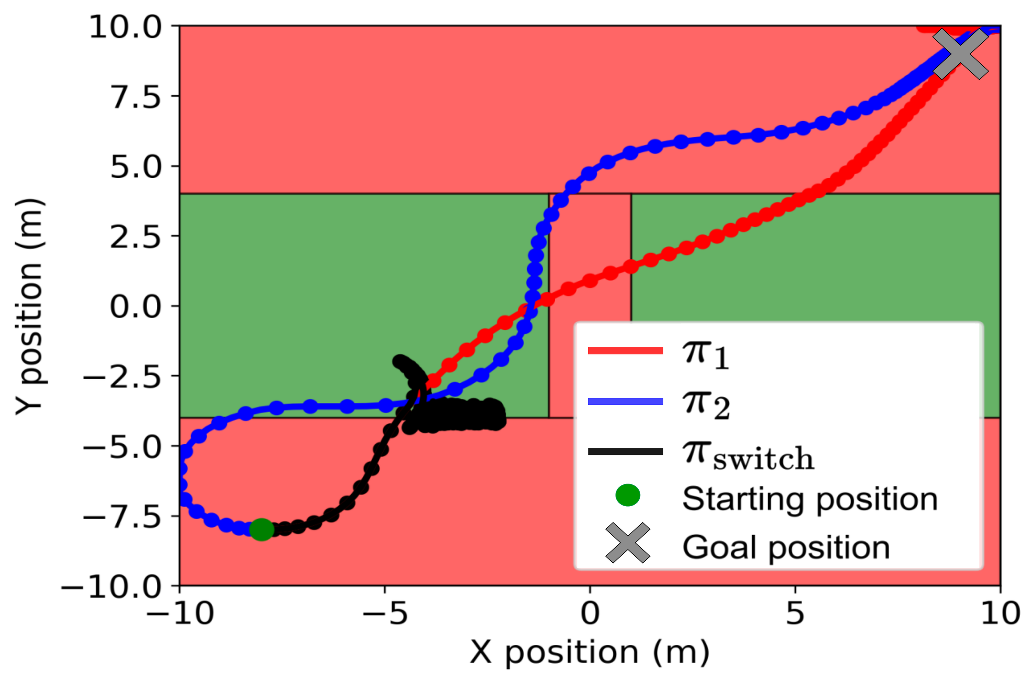
- Different terrains have different dynamics models.
- Moving between different terrains induces **sudden switches** in the dynamics function yielding a **hybrid system**.

Hybrid system model:

$$s_{t+1} = \sum_{m=1}^{M_E} \delta^E(m, s_t) f^m(s_t, a_t)$$

Mode switching indicator — Mode-specific dynamics

Can a policy that switches between mode-specific policies work?



$$\pi_{\text{switch}}(a_t|s_t) = \sum_{m=1}^{M_E} \delta^E(m, s_t) \pi_m(a_t|s_t)$$

Mode-specific policies

- Policies for each individual mode can yield very different behaviors.
- Switching between conflicting policies can yield stagnant trajectories.

In general, switching policies fail to account for the effects of switches between modes.

Problem Overview:

Can we learn policies that understand the importance of anticipating sudden switches in transition dynamics functions?

Method

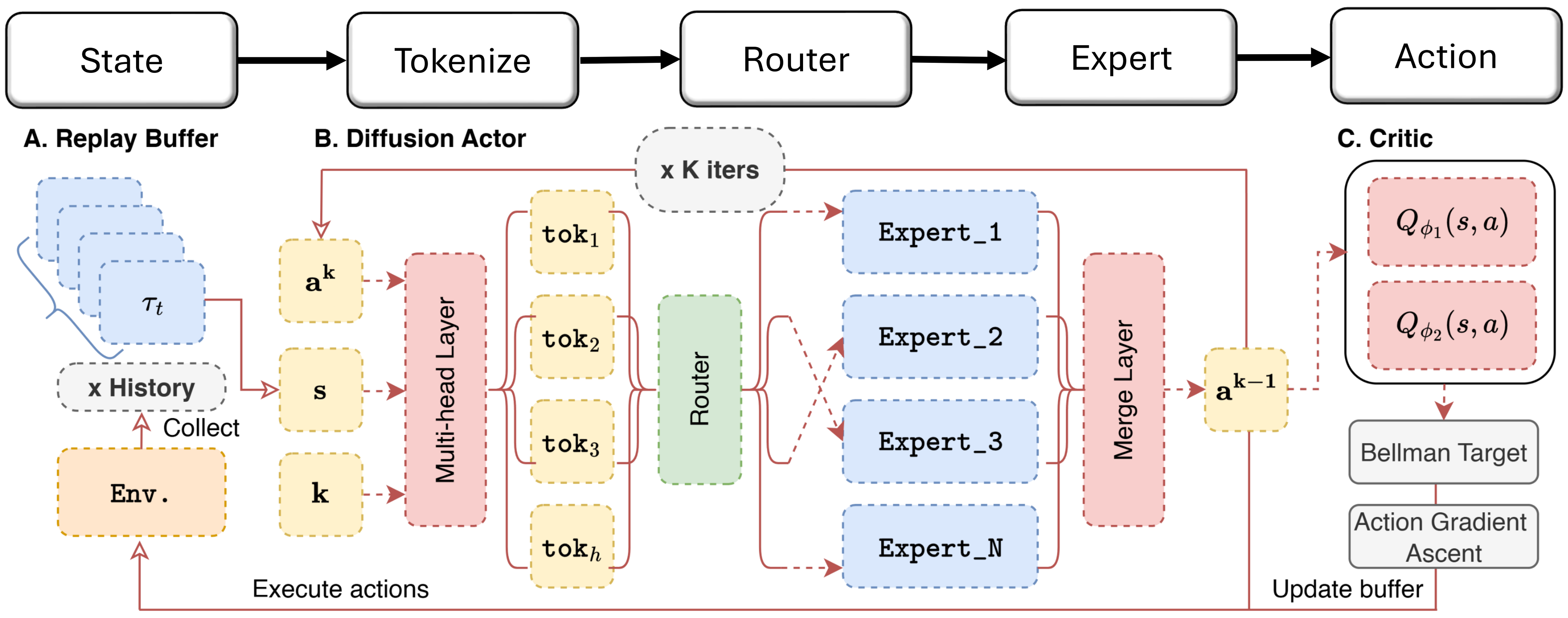
We propose MoE-Diff which learns an energy-diffusion policy using a Mixture-of-Experts (MoE) actor with an interpretable router.
We train MoE-Diff end-to-end via iterative energy minimization, composing and interpolating modes to adapt in *a priori* unknown environments.

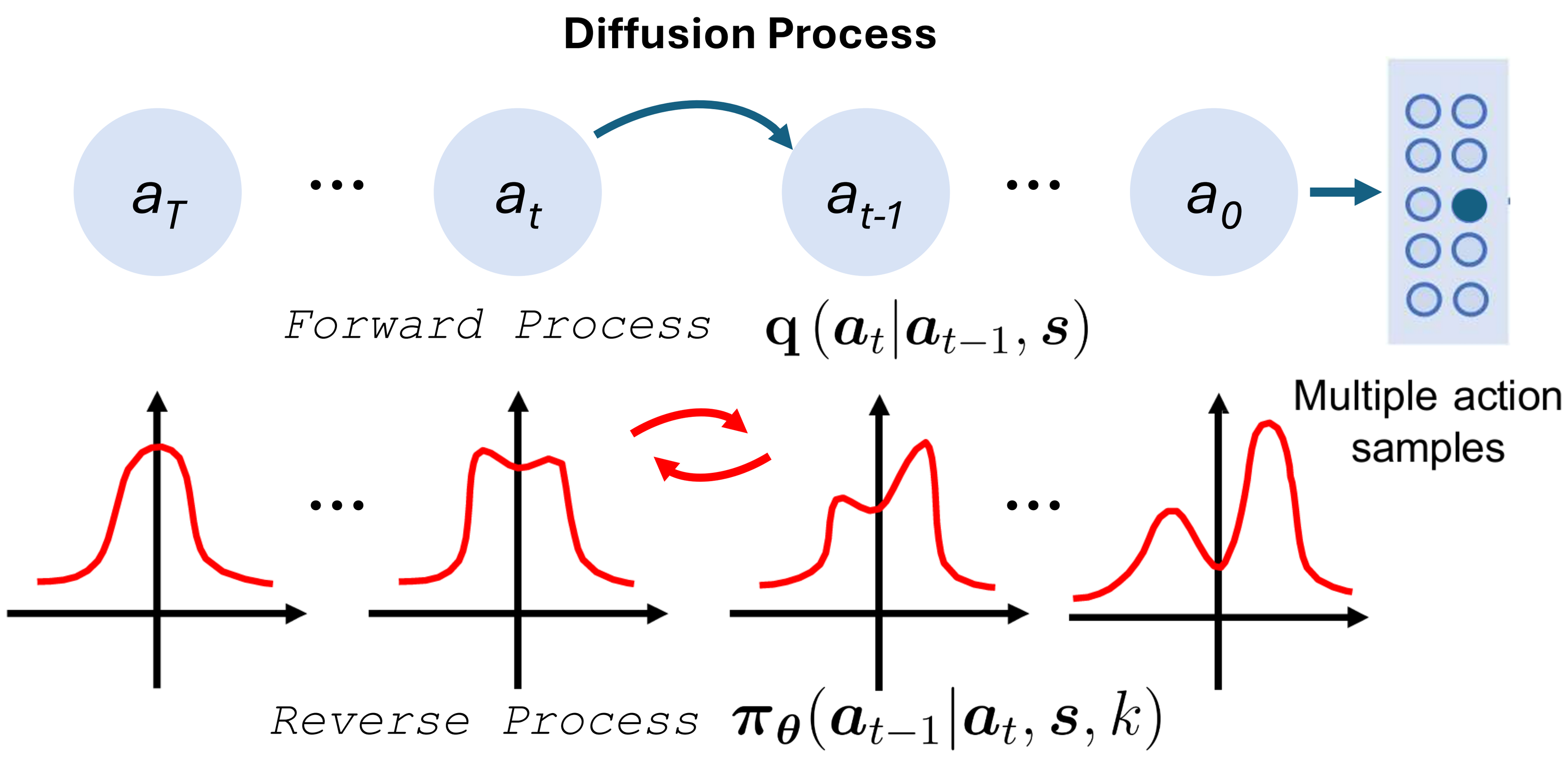
MoE-Diff Policy

A. **Replay Buffer** : Stores past observations and actions from environment interaction.

B. **Diffusion Actor (MHMoE[Wu et. al, 2017])** : Starts from a noisy action sample and iteratively denoises it into a final action, conditioned on the state and diffusion timestep; the final action is executed in the environment.

C. **Critic (Double Q[Hasselt, 2010])** : Estimates values and provides the action gradient to refine actions during policy improvement; refined actions are appended back to the buffer.





DDPM Loss

$$\|\epsilon - \pi_\theta(\sqrt{\bar{\alpha}_k} a_0 + \sqrt{1-\bar{\alpha}_k}\epsilon, s, k)\|^2$$

Contrastive Loss

$$-\log\left(\frac{e^{-\pi_\theta(a^*, s, k)}}{e^{-\pi_\theta(a^*, s, k)} + e^{-\pi_\theta(a^-, s, k)}}\right)$$

Training Regularizer

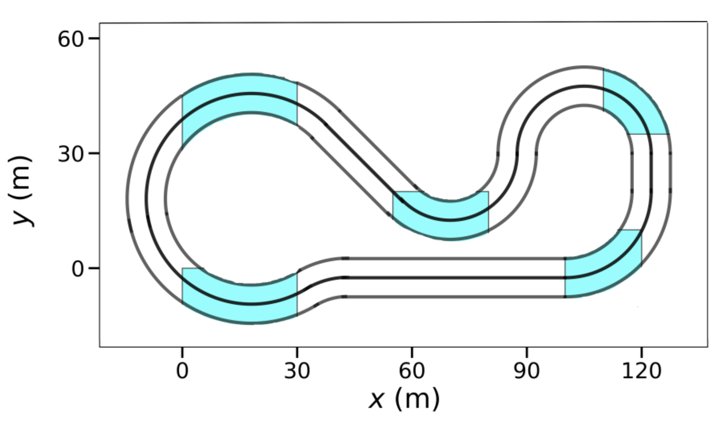
$$\lambda_z \cdot \mathbb{E}_{s\sim D}\left[\left(\log \sum_{p=1}^{N} \exp(R_\theta(s))\right)^2\right]$$

Action Gradient Ascent

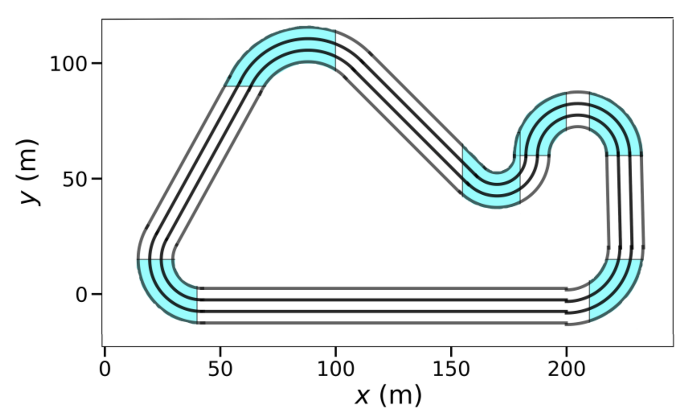
$$\tilde{a}^{n+1} \leftarrow a^n + \eta \nabla_a \left(\min_{j=1,2} Q_{\psi_j}(s, a)\right)\Big|_{a=a^n}$$

Results

Generalization to previously unseen environments



Training track layout

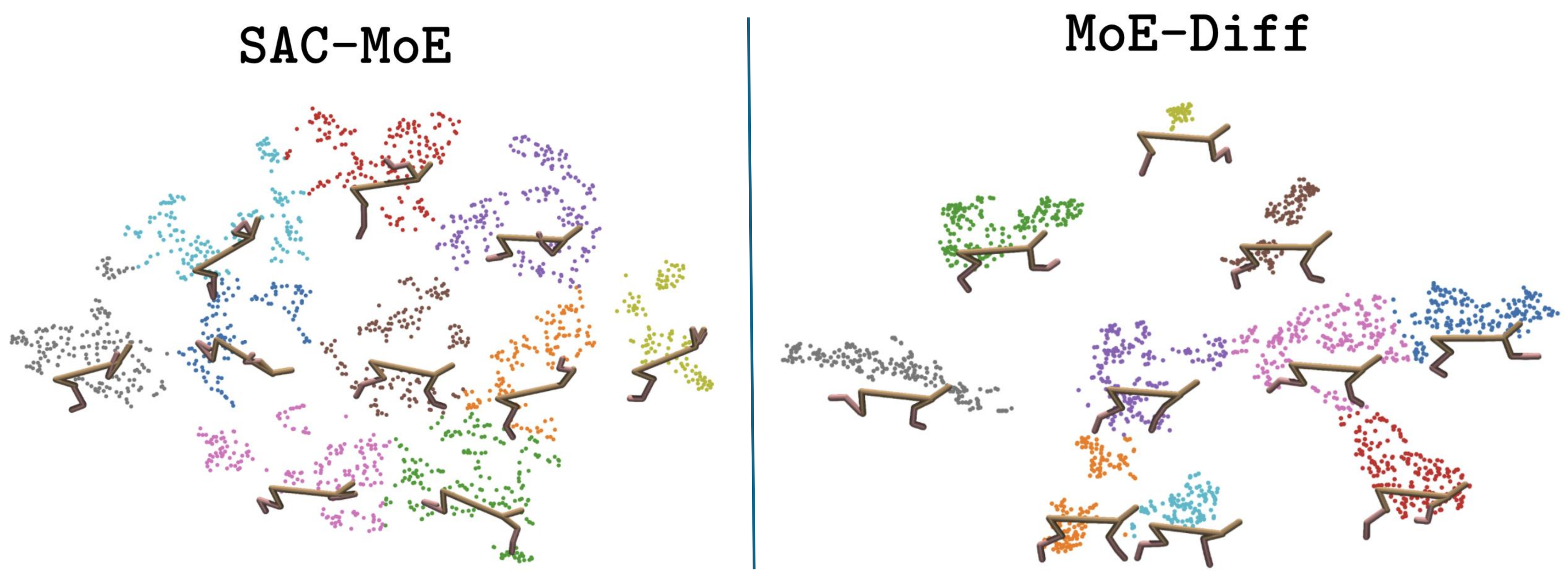


Evaluation track layout

Friction value(s)	SAC	UP-True	DiPo ⊕ $\mathcal{C}$	MoE-Diff (Ours)
Single friction tracks				
0.5	104.9 (80.3)	111.2 (69.0)	158.1 (83.8)	**224.4 (69.1)**
0.4	58.8 (54.4)	105.9 (66.9)	116.3 (76.4)	**162.3 (85.6)**
0.3	26.2 (17.1)	**81.1 (55.2)**	44.8 (36.3)	75.0 (52.5)
Hybrid (mode-switching) friction tracks				
{1.0, 0.5}	176.7 (100.9)	142.7 (82.5)	160.9 (89.0)	**232.8 (58.4)**
{0.3, 0.5}	49.6 (44.5)	100.2 (66.3)	120.1 (82.4)	**233.6 (51.6)**
{0.25, 0.4}	29.5 (37.3)	95.1 (60.6)	88.8 (71.4)	**144.2 (78.2)**

MoE-Diff generalizes to racetrack layouts and dynamics modes that were not observed during training.

Qualitative Analysis of Router Gating



t-SNE visualization of state embeddings with k-means cluster averages, showing distinct poses linked to different expert activations in the MoE architecture.

Future Work

- Automatic curricula for sampling hybrid training environments to **prevent instability** in policy learning and **improve sample efficiency**.
- Improving **behavior diversity** by training in reward-free settings with entropy maximization.