State-Corrected Predictive Preference Learning for Multimodal Robot Navigation on Uneven Terrain

Rushikesh Jadhav, Aryan Kumar Singh, Srinjoy Ganguly, Ravi Prakash, Vivek Kumar

***Abstract*— Multimodal interactive imitation learning for mobile robot navigation on uneven and unstructured terrain remains a largely open problem. While interactive imitation learning (IIL) has advanced rapidly in structured settings such as on-road driving and tabletop manipulation, these methods typically operate on compact state vectors or single modality inputs. Unstructured terrain introduces continuous variation in surface geometry and ground irregularity, demanding rich multimodal perception. Geometric structure from LiDAR, visual appearance from camera, and proprioceptive data from IMU each address a distinct perceptual gap essential for safe policies on uneven terrain. In this work, we adapt interactive imitation learning methods to mobile robot navigation on uneven terrain with fused LiDAR, camera, and IMU perception. Building on Predictive Preference Learning (Cai et al., 2025), we propose State-Corrected PPL (SC-PPL). In standard PPL, each expert intervention is bootstrapped across predicted future states by pairing every future observation with the same expert action taken at the current state. This assumption degrades over the prediction horizon because the appropriate corrective action at a distant future state may differ substantially from the one issued at the point of intervention — a problem amplified on uneven terrain, where small changes in position can demand very different steering and speed responses. SC-PPL addresses this by computing the appropriate expert action independently at each simulated future state rather than broadcasting the current correction forward. The resulting preference pairs carry corrective labels that remain valid at each future state, enabling preferences without label degradation. Experiments with a Clearpath Jackal on uneven terrain in Isaac Sim show that SC-PPL achieves higher navigation success compared to other IIL frameworks.**

I. INTRODUCTION

Autonomous navigation of wheeled robots on uneven and unstructured terrain is essential for applications such as search and rescue, agricultural monitoring, planetary exploration, and infrastructure inspection. Unlike structured environments such as paved roads or indoor floors, uneven terrain presents continuously changing elevation profiles, irregular surface geometry, and unpredictable ground contact conditions. These properties make it difficult to hand-craft reward functions or collect sufficiently diverse offline demonstrations, as the space of possible terrain configurations a robot may encounter is vast and the consequences of navigation errors — rollover, entrapment, or mechanical damage — are severe.

Reinforcement learning (RL) methods can in principle learn navigation policies through trial and error, but they require extensive environment interaction and risk driving the robot into dangerous states during exploration. Imitation learning (IL) avoids unsafe exploration by training on expert demonstrations, but offline demonstration datasets inevitably lack coverage of the safety-critical states the robot encounters during deployment, leading to compounding errors from distributional shift [3], [4]. Interactive imitation learning (IIL) addresses this by placing a human expert in the training loop to provide online corrective demonstrations when the agent makes mistakes [2], [5], [6], [7]. Methods such as HG-DAgger [2], Ensemble DAgger [5], and Thrifty-DAgger [6] have shown that even sparse expert corrections during training can substantially improve policy performance over pure offline IL.

However, existing IIL methods have been developed and evaluated primarily in structured domains — on-road autonomous driving [8], [9], [10] and tabletop robotic manipulation [7] where the environment is relatively predictable and a single sensor modality or a compact state vector often suffices. Applying IIL to uneven terrain navigation introduces challenges that these prior settings do not address. First, uneven terrain introduces continuous variation in surface geometry and ground irregularity, demanding multimodal perception that jointly captures geometric structure (LiDAR), visual appearance (camera), and body-terrain interaction dynamics (IMU) to support both the learning agent and the supervising expert. Second, the cost of expert supervision is amplified: on unpredictable terrain, the expert must continuously anticipate how the terrain ahead will affect the robot's stability and traction, increasing cognitive load. Third, and most critically, the correct action on uneven terrain can change rapidly over short distances — a steering command that is safe on a gentle slope may be hazardous just meters ahead where the gradient steepens or the surface transitions.

This third challenge is particularly relevant to Predictive Preference Learning from Human Interventions (PPL) [1], a recent IIL method that propagates each expert intervention into future states by constructing contrastive preference labels over a predicted trajectory. PPL predicts the agent's future states for a horizon of L steps, and for each predicted state, stores a preference pair in which the expert's corrective action is preferred over the agent's rejected action. This bootstrapping enables the policy to learn corrective behavior not only at the intervention state but also in the safety-critical regions the agent is expected to visit next. However, PPL assumes that the expert action taken at the current intervention state remains the appropriate correction at all L future states. On structured roads, where the environment changes slowly, this assumption holds reasonably well over short horizons. On uneven terrain, it breaks down: the terrain geometry and surface conditions at a state five steps ahead may demand a fundamentally different response than what

was appropriate at the point of intervention. As a result, the preference labels degrade in quality over the horizon, limiting the practical length of L and the effectiveness of the method.

In this work, we adapt interactive imitation learning methods to wheeled robot navigation on uneven terrain with fused LiDAR, camera, and IMU perception. Building on PPL, we propose State-Corrected PPL (SC-PPL), which eliminates the action broadcasting assumption by computing the appropriate expert action independently at each predicted future state. During an intervention, the agent's rejected action is simulated forward through the physics engine for L steps. At each resulting future state, the expert planner evaluates the local terrain and computes the correct action for that specific state. The preference buffer is then populated with pairs in which the positive action is locally valid rather than propagated from a distant intervention point. This simple modification directly improves the quality of preference labels across the entire horizon. We evaluate SC-PPL alongside Ensemble DAgger and standard PPL all operating on the same multimodal observation space on uneven terrain in NVIDIA Isaac Sim using a Clearpath Jackal robot.

II. Related Work

A. Interactive Imitation Learning

Interactive imitation learning (IIL) incorporates an expert into the training loop to provide online corrective demonstrations, mitigating the distributional shift that afflicts offline imitation learning [3], [4]. Early methods such as DAgger [4] aggregate expert-labeled data at states visited by the novice policy. Human-Gated DAgger (HG-DAgger) [2] extends this by allowing the expert to decide when to intervene, reducing the labeling burden. Ensemble DAgger [5] uses Bayesian disagreement among an ensemble of policies to automatically request expert input when the agent is uncertain, while Thrifty-DAgger [6] introduces budget-aware novelty and risk gating to further limit expert queries.

Methods like EGPO [8] and Proxy Value Propagation (PVP) [9] attempt to minimize human involvement by designing proxy penalty functions for unsafe behavior. Despite this, the human expert is still required to maintain continuous oversight to catch potential failures before they occur. Furthermore, because these approaches do not exploit the agent's predicted future trajectories to identify undesirable outcomes in advance, they inherently require repeated corrective demonstrations whenever the agent encounters similar scenarios.

B. Preference-Based Reinforcement Learning

Preference-based RL learns reward functions or policies from preference rankings over trajectory pairs [13], [14]. Reinforcement Learning from Human Feedback (RLHF) trains an explicit reward model from offline preference data and uses it to guide policy optimization [15], [16]. More recently, Direct Preference Optimization (DPO) [17] and Contrastive Preference Optimization (CPO) [18] bypass reward model training and directly optimize the policy to satisfy preference labels through a classification loss. Related variants include IPO [19] and SimPO [20].

Applying these methods to real-time robot control is challenging because they require extensive labeling of preference data, which is costly and noisy [14], [12]. PPL [1] addresses this by automatically converting expert interventions into contrastive preference pairs over predicted future states, eliminating the need for explicit trajectory ranking.

C. Robot Navigation on Uneven Terrain

Navigation on uneven and unstructured terrain has been approached from multiple directions. Classical methods learn terrain cost functions from expert demonstrations for path planning on rough terrain. More recent work uses reinforcement learning with procedurally generated terrain and domain randomization to train locomotion and navigation policies in simulation. On the perception and localization side, LiDAR-inertial SLAM methods such as LeGO-LOAM [23] and LIO-SAM [24] have enabled robust mapping on variable and unstructured ground, while broader multimodal fusion of LiDAR, camera, and IMU has advanced traversability estimation [21] and state estimation on uneven terrain [22]. In contrast to map-based pipelines, our work takes a mapless approach, directly learning a reactive navigation policy from multimodal sensory input through interactive expert feedback.

III. Problem Formulation

We model the navigation task as a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, d_0 \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function, $r : \mathcal{S} \times \mathcal{A} \to [R_{\min}, R_{\max}]$ is the reward function, $\gamma \in (0, 1)$ is the discount factor, and d_0 is the initial state distribution. We consider the reward-free setting where the agent has no access to r.

Observation Space. Each state $s \in \mathcal{S}$ is represented by the concatenation of three sensor modalities and a goal vector:

$$\mathbf{o} = [\mathbf{o}_{\text{cam}};\ \mathbf{o}_{\text{lidar}};\ \mathbf{o}_{\text{imu}};\ \mathbf{g}]. \tag{1}$$

The camera observation $\mathbf{o}_{\text{cam}} \in \mathbb{R}^{3\times64\times64}$ captures terrain appearance. The LiDAR scan $\mathbf{o}_{\text{lidar}} \in [0, 1]^{675}$ provides normalized range measurements. The IMU reading $\mathbf{o}_{\text{imu}} \in \mathbb{R}^6$ encodes angular velocities and linear accelerations. The goal vector $\mathbf{g} \in \mathbb{R}^2$ encodes relative distance and heading to the goalpoint.

Action Space. The action $a = (v, \omega) \in \mathcal{A} \subset \mathbb{R}^2$ specifies linear and angular velocity commands, where $v \in [-v_{\max}, v_{\max}]$ and $\omega \in [-\omega_{\max}, \omega_{\max}]$. These are mapped to the left and right wheel angular velocities of the skid-steer platform using a differential-drive kinematic approximation:

$$\omega_l = \frac{v - \omega \cdot d/2}{r_w}, \qquad \omega_r = \frac{v + \omega \cdot d/2}{r_w}, \tag{2}$$

where d is the effective wheel track width and r_w is the wheel radius.

Interactive Imitation Learning Setting. The training loop includes an expert policy $\pi_h(a \mid s)$ that can intervene

when the novice policy $\pi_n(a \mid s)$ produces unsafe or undesirable actions. The intervention decision is modeled as a deterministic indicator function $I(s,a) : \mathcal{S} \times \mathcal{A} \to \{0,1\}$, where $I(s,a) = 1$ indicates that the expert takes control from the agent.

With the notations above, the agent's actual trajectories during training are sampled from the following shared behavior policy:

$$\pi_b(a \mid s) = \pi_n(a \mid s)(1 - I(s,a)) + \pi_h(a \mid s)G(s), \quad (3)$$

where $G(s) = \int_{a' \in \mathcal{A}} I(s,a')\pi_n(a' \mid s)da'$ is the marginal probability of the agent taking an action that will be rejected and intervened upon by the expert.

Trajectory Prediction. The agent has access to a trajectory prediction model $f : \mathcal{S} \times \mathcal{A} \times \mathbb{N} \to \mathcal{S}^H$ that, given the current state s and agent action a_n, produces a sequence of predicted future states:

$$f(s, a_n, H) = (\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_H), \quad (4)$$

where each predicted state $\tilde{s}_i$ contains the full multimodal observation $\tilde{s}_i = [o^{(i)}_{\text{cam}}; o^{(i)}_{\text{lidar}}; o^{(i)}_{\text{imu}}; g^{(i)}]$. We use the physics engine to simulate these future states.

Preference Alignment. Following CPO [18], given a preference dataset $\mathcal{D}_{\text{pref}}$ where each entry (s, a^+, a^-) indicates that action a^+ is preferred over a^- at state s, the policy π_θ is trained using:

$$\mathcal{L}_{\text{pref}}(\pi_\theta) = -\mathbb{E}_{(s,a^+,a^-)\sim\mathcal{D}_{\text{pref}}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(a^+|s)}{\pi_\theta(a^-|s)}\right)\right] \quad (5)$$

where $\sigma(\cdot)$ is the sigmoid function and $\beta > 0$ is a hyperparameter.

IV. Method

A. Predictive Preference Learning (PPL)

PPL [1] augments behavioral cloning with contrastive preference learning over predicted future states. At each state s, PPL performs the following:

1) The novice policy π_n suggests an action a_n at the current state s. Instead of executing a_n immediately, we query the trajectory prediction model $f(s, a_n, L)$ to obtain a predicted rollout $(\tilde{s}_1, \ldots, \tilde{s}_L)$ with full multimodal observations at each predicted state. These represent the trajectory the agent would have followed if a_n were applied for L steps.
2) For each predicted state $\tilde{s}_i$, a preference tuple $(\tilde{s}_i, a^+ = a_h, a^- = a_n)$ is added to the preference buffer $\mathcal{D}_{\text{pref}}$. The positive action a^+ is the expert's corrective action taken at state s, broadcast to all L future states. The negative action a^- is the policy's action at the state s
3) The pair (s, a_h) is added to the demonstration buffer $\mathcal{D}_h$.

The policy is trained with the combined loss:

$$\mathcal{L}(\pi_\theta) = \mathcal{L}_{\text{pref}}(\pi_\theta) + \mathcal{L}_{\text{BC}}(\pi_\theta), \quad (6)$$

Algorithm 1 Predictive Preference Learning (PPL)

Require: Preference horizon L, expert π_h, trajectory prediction model f
for each step t **do**
  Agent proposes action $a_n \sim \pi_n(s_t)$
  if intervention triggered **then**
    Expert computes action $a_h \sim \pi_h(s_t)$
    Add (s_t, a_h) to $\mathcal{D}_h$
    Predict future states $(\tilde{s}_1, \ldots, \tilde{s}_L) = f(s_t, a_n, L)$
    for $i = 1$ to L **do**
      Add $(\tilde{s}_i,\ a^+ = a_h,\ a^- = a_n)$ to $\mathcal{D}_{\text{pref}}$
    end for
    Execute a_h
  else
    Execute a_n
  end if
  Train π_n with $\mathcal{L} = \mathcal{L}_{\text{pref}} + \mathcal{L}_{\text{BC}}$
end for

where the behavioral cloning loss is defined as:

$$\mathcal{L}_{\text{BC}}(\pi_\theta) = -\mathbb{E}_{(s,a_h)\sim\mathcal{D}_h}\left[\log \pi_\theta(a_h \mid s)\right]. \quad (7)$$

B. State-Corrected PPL (SC-PPL)

SC-PPL modifies the preference label construction in PPL by replacing the broadcast expert action with a state-specific expert action at each predicted future state. Since the trajectory prediction model already produces the full state at each future step $\tilde{s}_i$, we can query the expert planner at $\tilde{s}_i$ to obtain the locally correct action $a^{(i)}_h$ rather than reusing the action a_h from the intervention state s.

At each state s, SC-PPL performs the following:

1) The novice policy π_n suggests an action a_n at the current state s. Instead of executing a_n immediately, we query the trajectory prediction model $f(s, a_n, L)$ to obtain a predicted rollout $(\tilde{s}_1, \ldots, \tilde{s}_L)$ with full multimodal observations at each predicted state. These represent the trajectory the agent would have followed if a_n were applied for L steps.
2) At each predicted state $\tilde{s}_i$, the expert planner computes a state-specific corrective action $a^{(i)}_h \sim \pi_h(\tilde{s}_i)$ based on the local terrain conditions at $\tilde{s}_i$. The tuple $(\tilde{s}_i, a^+ = a^{(i)}_h, a^- = a_n)$ is recorded in $\mathcal{D}_{\text{pref}}$. Unlike standard PPL, the positive action a^+ is now locally valid at each future state rather than broadcasting from the intervention point.
3) The pair (s, a_h) is added to the demonstration buffer $\mathcal{D}_h$.

The training objective remains identical to Eq. 6. But now the positive actions in $\mathcal{D}_{\text{pref}}$ are state-specific rather than broadcast.

Effect on the performance bound. In the framework of Theorem 4.1 from [1], SC-PPL directly reduces the preference label error δ_{pref}. Since each positive action $a^{(i)}_h$ is computed at the corresponding state $\tilde{s}_i$ rather than imported

Algorithm 2 State-Corrected PPL (SC-PPL)

Require: Preference horizon L, expert π_h, trajectory prediction model f
  for each step t **do**
    Agent proposes action $a_n \sim \pi_n(s_t)$
    if intervention triggered **then**
      Expert computes action $a_h \sim \pi_h(s_t)$
      Add (s_t, a_h) to $\mathcal{D}_h$
      Predict future states $(\tilde{s}_1, \ldots, \tilde{s}_L) = f(s_t, a_n, L)$
      for $i = 1$ to L **do**
        $a_h^{(i)} = \pi_h(\tilde{s}_i)$ ▷ **State-specific expert action**
        Add $(\tilde{s}_i,\ a^+ = a_h^{(i)},\ a^- = a_n)$ to $\mathcal{D}_{\text{pref}}$
      end for
      Execute a_h
    else
      Execute a_n
    end if
    Train π_n with $\mathcal{L} = \mathcal{L}_{\text{pref}} + \mathcal{L}_{\text{BC}}$
  end for

from a distant state s, the preference-pair distribution $\rho_{\text{pref}}^{\tilde{s}_i}$ more closely approximates the ideal distribution $\rho_{\text{ideal}}^{\tilde{s}_i}$. This reduction results in better preference labels.

V. Experimental Setup

All experiments are conducted in NVIDIA Isaac Sim using a Clearpath Jackal skid-steer robot equipped with a front-facing camera (64×64 resolution), a planar LiDAR scanner (675 points, 270° field of view), and a 6-axis IMU. We use uneven terrain with varying elevation profiles, slope gradients, and surface irregularity (Fig. 2). The robot has maximum linear and angular velocities of $v_{\max} = 2.0$ m/s and $\omega_{\max} = 2.0$ rad/s. The multimodal policy network encodes each modality independently and concatenates them into a shared feature vector. All compared methods share this common architecture and are trained for the same number of environment steps on an NVIDIA RTX A6000 GPU (Table I). For PPL and SC-PPL, we set $L = 2$, $\beta = 0.1$. For our ensemble baselines, Ensemble DAgger [5] and Mixture of Experts (MoE) [25], we maintain an ensemble size of $K = 3$ policies. In the MoE formulation, a learned gating network dynamically weights the outputs of the K expert policies based on the current multimodal observation to produce the final continuous action.

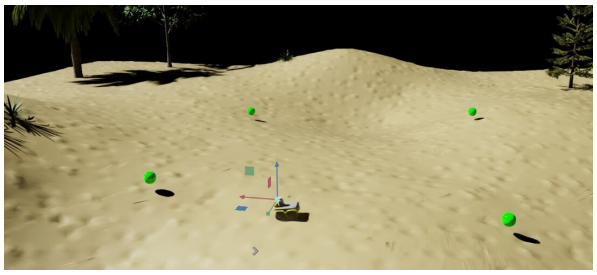

Fig. 1. Clearpath Jackal robot navigating on uneven terrain in NVIDIA Isaac Sim. Goalpoints are shown as green spheres.

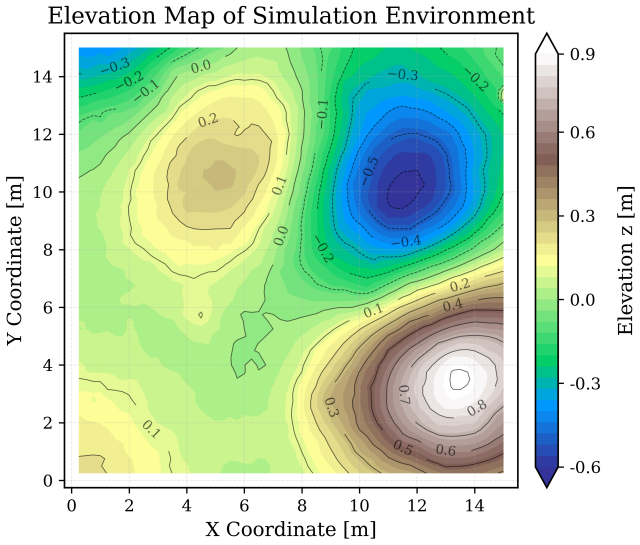

Fig. 2. Elevation map of the simulated environment, illustrating the continuous variation in surface geometry and elevation gradients.

A. Expert Policy

We use an automated expert policy to enable reproducible, large-scale experiments. The expert is a terrain-aware pure pursuit controller that follows globally planned paths on the slope-weighted cost map. The global path is computed using A* search with slope-based costs.

In SC-PPL, this same expert is queried at each simulated future state $\tilde{s}_i$ to compute the state-specific action $a_h^{(i)}$. Since the global path and cost map are precomputed, this re-evaluation adds negligible computational overhead.

B. Evaluation Metrics

Each policy is evaluated on held-out routes traversing procedurally generated terrain with diverse elevation profiles and slope gradients. To assess both the task performance and the quality of the learned navigation policies, we report the following metrics:

- **Task Performance**: We measure the *Success Rate* (percentage of routes completed without rollover, entrapment, or timeout) and *Route Completion* (average fraction of waypoints reached before termination).
- **Attitude RMS**: The root mean square of the robot's roll and pitch angles over the trajectory, quantifying the overall platform stability on uneven terrain. Let ϕ_t and θ_t be the robot's roll and pitch angles (in radians), respectively, at timestep t. The Attitude RMS over an episode of duration N is defined as $\sqrt{\frac{1}{N}\sum_{t=1}^{N}(\phi_t^2 + \theta_t^2)}$. Lower values indicate a smoother and safer traversal with less aggressive tilting.
- **Cross-Track Error (CTE)**: The mean shortest distance in 3D space between the robot's executed trajectory and the expert's path. Let $\mathbf{p}_t = (x_t, y_t, z_t) \in \mathbb{R}^3$ be the robot's Cartesian position at timestep t, and let $\mathcal{P}_{\text{ref}}$ be the continuous set of 3D points defining the expert's path over the uneven terrain. The cross-track error at

timestep t is the perpendicular Euclidean distance to the closest point on the reference path, defined as $e_{\text{cte}}(t) = \min_{\mathbf{p}_{\text{ref}} \in \mathcal{P}_{\text{ref}}} \|\mathbf{p}_t - \mathbf{p}_{\text{ref}}\|_2$. We report the mean CTE over the episode duration N, calculated as $\frac{1}{N}\sum_{t=1}^{N} e_{\text{cte}}(t)$.

- **Velocity Jitter (V-Jitter and W-Jitter)**: The step-to-step variation in the commanded linear (v) and angular (ω) velocities, measured in m/s^2 and rad/s^2, respectively. Lower jitter indicates smoother, more hardware-friendly control outputs

VI. Results

To evaluate the effectiveness of State-Corrected PPL (SC-PPL), we compare its performance against Behavioral Cloning (BC), Mixture of Experts (MOE), Ensemble DAgger (ED), and standard PPL. Our evaluation focuses on task success and robot stability.

A. Navigation Success

SC-PPL achieves the highest navigation success rate (88%), outperforming PPL (72%), MOE (55%), ED (43%), and BC (0%).

B. Path tracking and Stability

SC-PPL demonstrates the tightest adherence to the expert path, achieving the lowest mean Cross-Track Error (CTE) of 1.157 m. In comparison, standard PPL struggles to maintain this level of accuracy, exhibiting a nearly doubled mean CTE of 2.014. While ED achieves better stability (Attitude RMS of 0.113 rad) and smoother control (V-Jitter of 0.031 m/s^2), it suffers from a high failure rate. Ultimately, SC-PPL provides the best overall balance of success rate and stability.

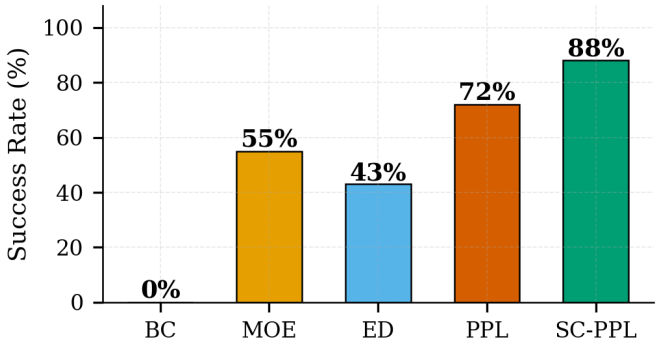

Fig. 3. Navigation success rate comparison across evaluated methods, illustrating the relative performance of SC-PPL alongside standard PPL and other baselines.

VII. Conclusion

In this work, we adapted interactive imitation learning methods for navigation on uneven terrain with fused LiDAR, camera, and IMU perception. We extended prior architectures such as Ensemble DAgger and PPL to leverage these multimodal features. We proposed State-Corrected PPL (SC-PPL), which replaces the action broadcasting assumption of standard PPL with state-specific expert action computation at each predicted future state. This modification directly improves the quality of preference labels in the preference buffer, enabling preference horizons without label degradation—a property that is particularly important on uneven terrain, where the correct action can change rapidly with position.

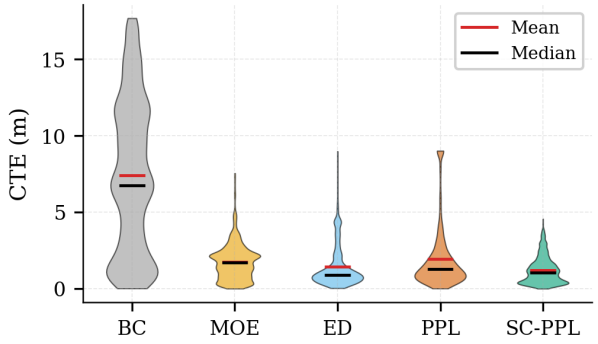

Fig. 4. Distribution of Cross-Track Error (CTE) for all tested policies, comparing adherence to the expert path across methods.

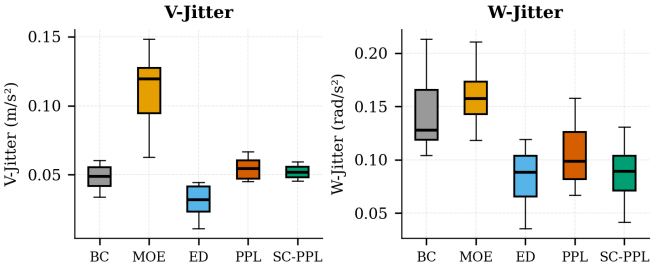

Fig. 5. Variation in linear (V-Jitter) and angular (W-Jitter) velocity commands. Lower jitter values indicate smoother and more hardware-friendly control.

Limitations. Our experiments are conducted entirely in simulation using an automated expert planner. The expert relies on access to the simulator's global cost map, which may not be available in real-world deployments. Extending SC-PPL to physical robots would require either a learned expert policy or a human operator capable of providing corrective actions, along with a real-time terrain model to support state-specific action computation at simulated future states. Additionally, the physics-engine forward simulation used for trajectory prediction incurs computational overhead that may limit real-time applicability on resource-constrained platforms. Deploying SC-PPL on physical hardware and evaluating its performance with human operators remain important directions for future work.

References

[1] H. Cai, Z. Peng, and B. Zhou, "Predictive Preference Learning from Human Interventions," in *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2025.

[2] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "HG-DAgger: Interactive imitation learning with human experts," in *2019 IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 8077–8083.

[3] S. Ross and J. A. Bagnell, "Efficient reductions for imitation learning," in *Proc. 13th Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2010, pp. 661–668.

[4] S. Ross, G. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2011, pp. 627–635.

TABLE I

COMPARISON OF METHODS WITH TRAINING/TESTING STATISTICS. THE OVERALL INTERVENTION RATE IS GIVEN TOGETHER WITH THE HUMAN DATA USAGE.

Method	**Human-in-the-Loop**	**Training**		**Testing**	
		Expert Data Percentage	**Total Data Usage**	**Success Rate**	**Route Completion**
BC	×	100	20K	0.00 ± 0.00	0.15 ± 0.02
Ensemble-DAgger	✓	54.9	20K	0.43 ± 0.11	0.72 ± 0.02
Mixture of Experts	✓	56.3	20K	0.55 ± 0.04	0.65 ± 0.06
PPL	✓	**32.1**	20K	0.72 ± 0.07	0.83 ± 0.09
SC-PPL	✓	43.9	20K	**0.88** ± 0.05	**0.91** ± 0.03

TABLE II

COMPARISON OF NAVIGATION QUALITY ACROSS ALL TEST EPISODES. THE TABLE REPORTS MEAN CROSS TRACK ERROR, AVERAGE DISTANCE, AND MOTION SMOOTHNESS METRICS (ATT. RMS, JITTER, ROLL, PITCH) FOR EACH METHOD.

Method	**Mean CTE** (m)	**Avg Dist.** (m)	**Att. RMS** (rad)	**V-Jitter** (m/s^2)	**W-Jitter** (rad/s^2)	**Max Roll** (rad)	**Max Pitch** (rad)
BC	7.323 ± 2.903	71.2 ± 52.0	0.165 ± 0.042	0.048 ± 0.008	0.142 ± 0.033	0.370 ± 0.155	0.255 ± 0.089
MOE	1.722 ± 0.464	59.2 ± 22.1	0.138 ± 0.041	0.112 ± 0.025	0.160 ± 0.045	0.234 ± 0.052	0.279 ± 0.068
ED	1.460 ± 0.699	55.3 ± 16.1	**0.113 ± 0.038**	**0.031 ± 0.011**	**0.084 ± 0.026**	**0.215 ± 0.093**	**0.245 ± 0.106**
PPL	2.014 ± 1.452	**44.8 ± 14.0**	0.144 ± 0.020	0.054 ± 0.008	0.106 ± 0.030	0.297 ± 0.163	0.255 ± 0.040
SC-PPL	**1.157 ± 0.328**	45.2 ± 14.9	0.132 ± 0.027	0.050 ± 0.008	0.088 ± 0.025	0.229 ± 0.056	0.258 ± 0.063

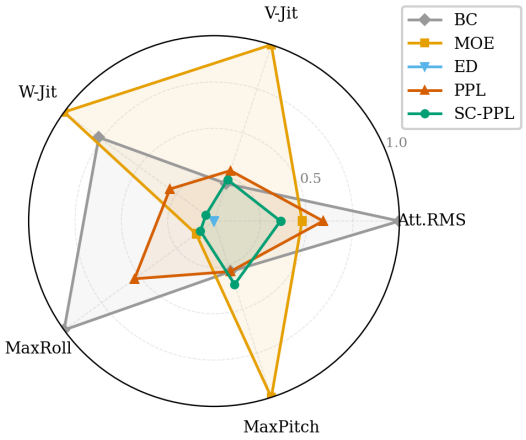

Fig. 6. Radar chart comparing platform stability and action smoothness. Values are normalized with smaller areas (closer to the center) indicating superior performance.

[5] K. Menda, K. Driggs-Campbell, and M. J. Kochenderfer, "EnsembleDAgger: A Bayesian Approach to Safe Imitation Learning," in *2019 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 5041–5048.

[6] R. Hoque et al., "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning," in *Conf. on Robot Learning (CoRL)*, 2021.

[7] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese, "Human-in-the-Loop Imitation Learning using Remote Teleoperation," *arXiv preprint arXiv:2012.06733*, 2020.

[8] Z. Peng et al., "Safe Driving via Expert Guided Policy Optimization," in *Conf. on Robot Learning (CoRL)*, 2021.

[9] Z. Peng et al., "Learning from Active Human Involvement through Proxy Value Propagation," in *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 36, 2024.

[10] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Learning from Interventions," in *Robotics: Science and Systems (RSS)*, 2020.

[11] Q. Li et al., "Human-AI Copilot Optimization," in *Int. Conf. on Learn. Represent. (ICLR)*, 2022.

[12] J. Hejna, R. Rafailov, H. Sikchi, C. Finn, S. Niekum, W. B. Knox, and D. Sadigh, "Contrastive preference learning: learning from human feedback without rl," *arXiv preprint arXiv:2310.13639*, 2023.

[13] P. F. Christiano et al., "Deep Reinforcement Learning from Human Preferences," in *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2017.

[14] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active Preference-Based Learning of Reward Functions," in *Robotics: Science and Systems (RSS)*, 2017.

[15] L. Ouyang et al., "Training language models to follow instructions with human feedback," in *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2022.

[16] N. Stiennon et al., "Learning to summarize from human feedback," in *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2020.

[17] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 36, pp. 53728–53741, 2023.

[18] H. Xu et al., "Contrastive Preference Optimization," *arXiv preprint*, 2024.

[19] M. G. Azar et al., "A General Theoretical Paradigm to Understand Learning from Human Preferences," in *Int. Conf. on Artif. Intell. Stat. (AISTATS)*, 2024.

[20] Y. Meng et al., "SimPO: Simple Preference Optimization with a Reference-Free Reward," *arXiv preprint*, 2024.

[21] M. Waibel et al., "Terrain Traversability Estimation," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2013.

[22] D. Wisth et al., "VILENS: Visual, Inertial, Lidar, and Leg Odometry," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 309–326, 2023.

[23] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 4758–4765.

[24] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 5135–5142.

[25] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.