ICRA2026 Late Breaking Results Poster presented at
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Knowledge-Based Locomotion Policy for Quadruped Robots under Incomplete Terrain Observation

Taehyeong Kim and Sangmoon Lee*

***Abstract*— Body-mounted LiDAR sensors suffer from systematic blind spots during stair locomotion, creating a partial observability problem that single-step terrain snapshots cannot resolve. We address this with a recurrent locomotion policy for the Unitree Go2 that builds implicit knowledge of stair geometry through a GRU-based recurrent encoder over pointcloud and proprioceptive inputs, enabling robust stair ascent and descent even under occluded LiDAR conditions. Ablation experiments show that masking pointcloud input at inference time causes catastrophic failure on stair terrain and severe performance degradation overall, confirming that implicit stair knowledge is a critical cue for step negotiation rather than a merely complementary signal.**

I. INTRODUCTION

Body-mounted LiDAR sensors suffer from systematic blind spots during stair locomotion: as the robot pitches during ascent or descent, the frontal stair surface can fall outside the sensor field of view. Existing exteroceptive locomotion policies typically assume complete terrain observation [3], and their performance degrades when this assumption is violated at deployment. To address this partial observability problem, we learn a recurrent locomotion policy that builds implicit knowledge of stair geometry from pointcloud and proprioceptive inputs, enabling robust stair traversal even under occluded LiDAR conditions.

Contributions

- *A knowledge-based recurrent formulation under partial observability* that encodes implicit stair geometry knowledge through a GRU over exteroceptive and proprioceptive inputs, enabling robust traversal under LiDAR blind spots.
- *A realistic body-mounted LiDAR simulation* that captures mounting-induced visibility degradation during stair locomotion.
- *An adaptive pointcloud curriculum* that progressively transfers training from reference terrain sensing to body-mounted LiDAR observation.
- *A proprioceptive auxiliary supervision objective* that improves recurrent state learning under degraded exteroceptive conditions.

Taehyeong Kim is a M.S. student with the Department of Electronic Engineering, Kyungpook National University, Daegu 41566, Republic of Korea rablk@knu.ac.kr

Sangmoon Lee is with the Department of Electronic Engineering, Kyungpook National University, Daegu 41566, Republic of Korea. He is the corresponding author. moony@knu.ac.kr

II. METHOD

Fig. 1 illustrates the proposed stair locomotion policy. The body-mounted LiDAR in simulation follows the Go2 Mid-360 mounting geometry and produces 96 ray hits arranged in a 12×8 pattern. These hits are represented as robot-frame (x, y, z) coordinates, forming a 288-dimensional pointcloud observation. Because LiDAR occlusion makes any single scan an incomplete snapshot of the terrain, the policy uses a GRU to accumulate exteroceptive and proprioceptive cues over time, forming a knowledge representation that persists across partial observations.

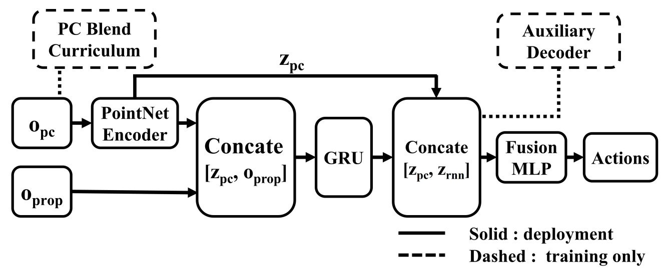


Fig. 1: Knowledge-Based Recurrent Locomotion Policy

The policy encodes $\mathbf{o}_{\mathrm{pc}}$ into a 64-dimensional terrain feature $\mathbf{z}_{\mathrm{pc}}$ using a PointNet-based encoder, and concatenates it with $\mathbf{o}_{\mathrm{prop}}$ before feeding into a GRU to produce a 256-dimensional recurrent feature $\mathbf{z}_{\mathrm{rnn}}$. The recurrent feature and the terrain feature are then fused by an MLP to generate 12-dimensional joint actions.

To improve robustness under incomplete terrain sensing, training starts from reference point clouds and gradually shifts toward body-mounted LiDAR observations through a PC Blend Curriculum. The transition is controlled by feature similarity between the reference and LiDAR point clouds rather than by a fixed schedule. An auxiliary decoder additionally predicts base linear velocity from proprioceptive sequences, encouraging the recurrent module to preserve motion-state information even when exteroceptive cues are degraded. Terrain difficulty and commanded speed are expanded progressively as policy performance improves.

III. RESULTS

This section quantitatively evaluates the proposed pointcloud based recurrent stair locomotion policy. We consider three policy settings: (1) **Baseline E (normal)**, which uses both proprioception and pointcloud input, (2) **Baseline E (blind PC)**, which uses the same trained Baseline E policy but masks the pointcloud input to zero at inference time, and

TABLE I: Success Rate and Stumble Rate on StairAscend.

Method	Flat	Tiny	Medium	Hard	Stumble (%)
Baseline E (normal)	**100.0**	**98.4**	**100.0**	**100.0**	**0.79**
Baseline E (blind PC)	39.2	1.6	0.0	0.0	0.96
Baseline A	100.0	87.2	100.0	99.2	4.59

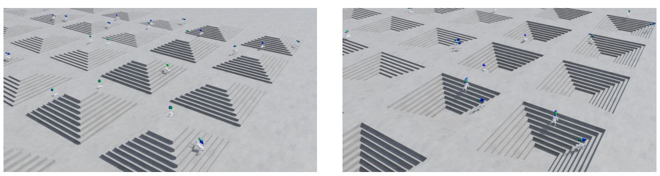
TABLE II: Success Rate and Stumble Rate on StairDescend.

Method	Flat	Tiny	Medium	Hard	Stumble (%)
Baseline E (normal)	**100.0**	**100.0**	**98.4**	**86.4**	**0.08**
Baseline E (blind PC)	100.0	0.0	0.0	0.0	0.15
Baseline A	100.0	100.0	100.0	75.2	0.47

(3) **Baseline A**, a proprioception-only MLP policy trained without pointcloud input.

A. *Experimental Setup*

Evaluation is conducted in the Isaac Lab play environment with 50 parallel environments spanning all terrain difficulty groups: flat, tiny, medium, and hard. For stair terrains, these groups correspond to different step-height ranges: tiny (0.02–0.06 m), medium (0.05–0.15 m), and hard (0.10–0.25 m). The evaluation tasks are `Unitree-Go2-StairAscend` and `Unitree-Go2-StairDescend`. A fixed forward velocity command of 0.5 m/s is applied throughout evaluation, and 500 episodes are collected per policy setting.



(a) StairDescend environment (b) StairAscend environment

Fig. 2: Evaluation environments in Isaac Lab. Left: StairDescend. Right: StairAscend.

B. *Success Criterion*

An episode is considered successful if the robot either reaches timeout or exits the tile boundary without failure, while also satisfying task-dependent progress conditions where applicable. Here, t_o denotes time_out, r_e denotes reached_tile_edge, and p_o denotes progress_ok.

For flat terrain, success is defined as

$$\mathrm{success}_{\mathrm{flat}} = t_o \vee r_e. \tag{1}$$

For stair ascent, success additionally requires sufficient upward height progression:

$$\mathrm{success}_{\mathrm{ascend}} = (t_o \vee r_e) \wedge p_o \wedge \Delta z \geq H_{\mathrm{target}}. \tag{2}$$

For stair descent, success is similarly defined as

$$\mathrm{success}_{\mathrm{descend}} = (t_o \vee r_e) \wedge p_o \wedge \Delta z \leq -H_{\mathrm{target}}. \tag{3}$$

Here, p_o requires the robot to move at least $1.5\,\mathrm{m}$ from its spawn origin, and H_{target} is defined as 80% of the estimated stair height of the assigned terrain.

C. *Quantitative Results*

Table I and Table II summarize the success rates and stumble rates on the StairAscend and StairDescend tasks.

StairAscend. Baseline E (normal) achieves near-perfect success across all terrain types, while Baseline A degrades on tiny stairs (87.2%) and shows a nearly 6× higher stumble rate (4.59% vs. 0.79%). The large stumble gap suggests that without pointcloud input, the policy has greater difficulty estimating step geometry and timing foot clearance correctly, leading to more frequent foot catches even on shallow steps. When pointcloud input is masked at inference time (blind PC), performance collapses on stair terrain (0.0% on medium and hard) and degrades even on flat terrain (39.2%), confirming that the recurrent module cannot compensate for missing exteroceptive features through proprioception alone.

StairDescend. On hard terrain, Baseline E outperforms Baseline A by 11.2 percentage points (86.4% vs. 75.2%) and achieves a 6× lower stumble rate (0.08% vs. 0.47%), indicating that pointcloud input helps the policy anticipate large step drops and maintain stable footholds during descent. Blind PC collapses to 0% on all stair terrain while retaining 100% on flat, showing that proprioception alone is sufficient for level-ground locomotion but cannot provide the terrain preview needed to control descent on steps.

IV. CONCLUSION

This work presented a knowledge-based recurrent formulation for quadrupedal stair locomotion under partial LiDAR observability, combining pointcloud terrain features with GRU-based proprioceptive history. The blind PC ablation demonstrates that the learned policy actively encodes exteroceptive terrain structure into its recurrent state: removing pointcloud input at inference time causes catastrophic failure on stair terrain and severe performance degradation overall. These results suggest that the recurrent encoder acquires implicit knowledge of stair geometry during training: this knowledge is actively encoded in the recurrent state and cannot be recovered from proprioception alone, confirming its role as a critical cue for step negotiation. Future work will validate this framework on the real Unitree Go2 and evaluate sim-to-real transfer under physical sensor occlusion.

REFERENCES

[1] M. Mittal *et al.*, "Isaac Lab: A GPU-accelerated simulation framework for multi-modal robot learning," arXiv:2511.04831, 2025.

[2] J. Schulman *et al.*, "Proximal policy optimization algorithms," arXiv:1707.06347, 2017.

[3] N. Rudin *et al.*, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proc. Conf. Robot Learning (CoRL)*, 2022.

[4] C. R. Qi *et al.*, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.