Traversing Narrow Paths: A Two-Stage Reinforcement Learning Framework for Robust and Safe Humanoid Walking

Tianchen Huang, Runchen Xu, Yu Wang, Wei Gao and Shiwu Zhang

Abstract— Traversing narrow paths is challenging for humanoid robots due to the sparse and safety-critical footholds required. Purely template-based or end-to-end reinforcement learning-based methods suffer from such harsh terrains. This paper proposes a two–stage training framework for such narrow path traversing tasks, coupling a template-based foothold planner with a low-level foothold tracker from Stage-I training and a lightweight perception aided foothold modifier from Stage-II training. With the curriculum setup from flat ground to narrow paths across stages, the resulted controller in turn learns to robustly track and safely modify foothold targets to ensure precise foot placement over narrow paths. This framework preserves the interpretability from the physics-based template and takes advantage of the generalization capability from reinforcement learning, resulting in easy sim-to-real transfer. The learned policies outperform purely template-based or reinforcement learning-based baselines in terms of success rate, centerline adherence and safety margins. Validation on a Unitree G1 humanoid robot yields successful traversal of a $0.2\,m$–wide and $3\,m$–long beam for 20 trials without any failure.

I. INTRODUCTION

Safe and accurate footholds are critical for humanoid robots traversing narrow paths, where the path width for feasible footholds shrink to one foot level and even modest perception or control delays can precipitate failure due to vanished recovery margins. Therefore, successful narrow path traversal hinges on efficient terrain perception, precise foothold selection and robust locomotion control.

Within this context, existing approaches mainly follow two paradigms. On the one hand, model-based foothold generation methods take advantage of compact template models to provide guidance for where to place the swing foot to yield balanced locomotion control [1]. On the other hand, model-free Reinforcement Learning (RL) methods learn end-to-end foothold selection and locomotion control from data, *e.g.*, an attention-based terrain map encoder trained jointly with the control policy has realized generalized legged locomotion [2]. When given sparse foothold options, learning-based methods typically introduce curriculum schedules and lightweight exteroception to cope with sparse rewards. A representative example is *BeamDojo*, which employs a two-stage RL pipeline with a sampling-based foothold reward and onboard LiDAR terrain height mapping to achieve hardware-validated traversal over narrow beams and stepping stones [3].

The authors are with the Institute of Humanoid Robots, Department of Precision Machinery and Precision Instrumentation, University of Science and Technology of China, Hefei, Anhui 230026, China. weigao@ustc.edu.cn; wangyuustc@ustc.edu.cn
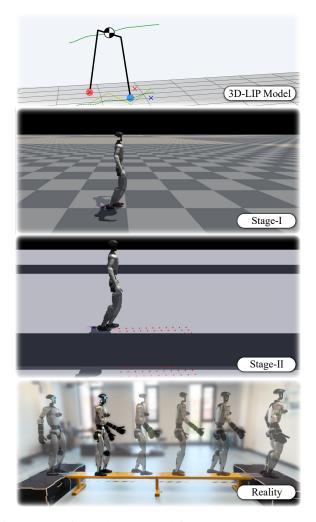
Fig. 1: Overview. The proposed framework uses 3D-LIPM as foothold planner, low-level policy from Stage-I training as foothold tracker and high-level policy from Stage-II training as foothold modifier for robust and safe narrow path walking. Successful experimental validation on the Unitree G1 humanoid robot is performed.

Despite the progress, narrow path traversal remains challenging. Purely end–to–end RL methods can overfit simulator–specific assumptions, facing unsafe exploration and suffering from limited foothold interpretability in safety–critical tasks. Conversely, purely template–based methods are vulnerable to modeling discrepancy, contact uncertainty and control system latency, resulting in inaccurate foot placement on limited support areas. These limitations have motivated

a category of synthesized methods, which retain a physics-based prior as an interpretable foothold planner and allocate residual learning as a safety–relevant modifier. Thus, the residual learning methods can augment the nominal controller with data–driven refinements, improving robustness without discarding insights from physical models [4].

Following this flavor, this paper proposes a lightweight and efficient two–stage reinforcement learning framework for robust and safe narrow path traversal. Stage-I is trained in simulation on flat ground: a Linear Inverted Pendulum model (LIPM) based foothold planner is utilized during training such that the low-level RL-based tracker can robustly follow each foothold and realize stable contact scheduling. Stage-II is trained in simulation on narrow paths: a high–level RL-based modifier generates a body–frame residual for the swing foot only, refining the foothold generated by the planner to ensure safe and precise foot placements on narrow paths. This setup helps preserve the interpretability from physics-based template models. Additionally, in the proposed method, sensing is kept minimal and consistent across the simulated and the physical robots to ease deployment, avoiding heavyweight vision pipelines with only necessary information for safe foot placement.

Overall, this paper makes two key contributions:

1) *Physics-guided foot placement learning with a two-stage training curriculum for narrow path traversal.* The LIPM-based foothold is refined by the bounded body-frame residual to achieve robust and safe foothold selection for the swing leg. Stage-I learns a robust foothold tracker via intentionally added target disturbance through training on flat ground, and Stage-II optimizes terrain-aware objectives for a safe foothold modifier on narrow paths.

2) *Minimal sensing requirement and experimental proof on a physical humanoid robot.* Using only compact anterior terrain height sampling maps and onboard IMU/joint signals with consistent representation in both simulation and experiments, a Unitree G1 humanoid robot has been able to reliably traverse a narrow beam of $0.2\,m$ wide and $3\,m$ long, outperforming methods based on either template models or Reinforcement Learning purely.

II. RELATED WORK

A. Physics-Based Foothold Planning for Locomotion Control

Bipedal robots from early years achieve balanced walking via reduced-order models and analytic stability measures. Preview control of Zero-Moment Point (ZMP) based on the Linear Inverted Pendulum model enables locomotion pattern generation that respects balance constraints [5]. N-step capturability has provided a theoretical framework for feasible stabilization within a finite number of steps [6]. The extrapolated Center of Mass (CoM) concept connects CoM state to required foot placement for balance control [7], and has led to the idea of Instantaneous Capture Point (ICP) for selecting "when and where to step" for push recovery [1].

These physics-grounded methods remain influential because they yield interpretable balance rules and compact foothold representations.

With model-based foothold planners, Model Predictive Control (MPC) has been extensively used to explicitly optimize future footholds under CoM dynamics, enabling online adaptation to external disturbances. The LIPM-MPC formulations take advantage of linear MPC to adjust future footholds online, for either balance control or velocity tracking [8]–[10]. Recent variants can plan both step position and orientation [11], [12], or leverage reduced-order models such as DCM/ALIP, for improved locomotion performance [13], [14]. Additionally, adapting step duration can also markedly improve landing accuracy for restricted footholds. By modulating swing duration online, precise contacts can be realized even when feasible regions are small [15].

On the other hand, recent work also combines template-based foothold planner with model-free RL, using physics-based guidance to shape the action space during training [4]. Hardware-validated improvements have been reported with RL policies tracking template-based footholds. However, this approach so far is only tailored to flat-ground locomotion. When sparse-support tasks such as beam traversal are confronted, recent advancements have proposed to incorporate exteroceptive terrain perception for foothold adjustment within purely RL framework [2], [3]. Nevertheless, an efficient narrow path traversing control framework with high success rate and careful safety consideration based on interpretable physical models is still lacking. Therefore, this paper utilizes the Linear Inverted Pendulum model, and confines learning to a lightweight residual learning.

B. Residual Learning with Staged Curriculum and Local Terrain Perception

Residual learning approaches augment a nominal controller with a learned correction, improving performance while preserving attributes of the nominal controller. In manipulation, Residual Policy Learning improves nondifferentiable policies by learning an additional residual policy [16]. Similarly, Residual Reinforcement Learning demonstrates that decomposing a controller into a physics-based model and a learned residual policy yields data-efficient behaviors on harware [17].

However, for the narrow path traversing tasks considered in this paper, the supervision for residual learning can easily collapse due to sparse and high-risk foothold rewards. The failure sensitivity of the training process makes residual learning brittle without additional framework structure or training curricula. In simulation, *ALLSTEPS* has showed that curriculum-driven RL can master stepping-stone locomotion and highlighted the importance of staged learning for contact-constrained locomotion tasks [18]. Most pertinent to our settings, *BeamDojo* tackles humanoid locomotion with sparse footholds using a two-stage RL pipeline based on LiDAR-based terrain height sampling maps, resulting in successful locomotion on beams and stepping stones in both simulated and real worlds [3]. Therefore, the proposed for-

mulation in this paper shares the staged training philosophy but differs by using the explicit LIPM as foothold planner and the lightweight residual learning as foothold modifier to realize more robust and safer locomotion behaviors.

As mentioned, to determine adequate footholds through residual learning, local terrain height sampling maps have emerged as a necessary part of the framework. Robust quadruped controllers trained in simulation have succeeded in natural environments by consuming compact height maps rather than heavy vision stacks. Efficient terrain height mapping pipelines via Graphics Processing Units further enable real-time robot-centric terrain height maps for locomotion control [19]. Recent progress integrates such local terrain height maps to achieve field robustness [20]. High-agility behavior demonstrations (*e.g.*, ANYmal Parkour) further validate the viability of compact terrain perception on hardware [21]. This paper follows this trend by sampling a minimal anterior terrain height map, enabling the proposed controller to refine footholds for narrow path traversal without heavy-weight perception.

III. METHOD

A. Framework Overview

To enable safe and repeatable humanoid traversal over narrow paths like beams, where contacts are sparse and any failure can be catastrophic, this paper focuses on a lightweight and physics-based architecture that separates *where to step* from *how to realize the step*, as shown in Fig. 2. Given the robot's states, a 3D-LIPM first plans the next swing-leg foothold target u_{init}. The high-level modifier then predicts a body-frame residual $\Delta u = (\Delta x, \Delta y, \Delta \psi)$ to refine the initial foothold as the final target $u_{\text{final}} = u_{\text{init}} \oplus \Delta u$, where $\oplus$ denotes composition in the task space. The low-level tracker finally issues desired joint positions to a Proportional-Derivative (PD) controller to ensure u_{final}. This framework confines learning to a safety-relevant role while keeps the nominal stepping physics explicit and interpretable.

The foothold modifier and tracker are trained through RL with distinct objectives. *Stage-I* trains the robust tracker via intentionally added foothold disturbances: a small and random zero-mean offset is added to the LIPM-based foothold target at every step, so the policy learns reliable foothold tracking and clean swing–stance transition under a scheduled right/left stance alternation, without any manual centerline locking. During Stage-I rollouts, the tracking policy runs at $100\,Hz$ and outputs desired joint positions, which are executed by the joint PD controller at $1\,kHz$. Note that the high-level modifier is not used in this stage. *Stage-II* trains the foothold modifier to refine the initially planned footholds on narrow support: the modifier predicts a body-frame residual for the swing foot only. The modifier is event-driven, queried once when the step transition occurs. Its output is held constant between consecutive events. During Stage-II rollouts, the operational frequencies remain the same: the tracker runs at $100\,Hz$ with joint PD control at $1\,kHz$, while the modifier updates upon step transition

events. The two policies thus run at different rates but are synchronized by the shared gait event.

To ease deployment, besides the onboard IMU signals and joint states, the control framework consumes only compact perception information. The same representation is used in both simulation and physical experiments. No heavy-weight vision stack is required. This results in an lightweight and interpretable framework for efficient training and sim-to-real transfer.

B. Stage-I Training for Robust Foothold Tracker

Stage-I trains a robust low-level foothold tracker on flat ground that can adapt to small random foothold disturbances, preparing for the modified foothold on narrow paths by the residual from Stage-II policy. However, before the Stage-I training can be carried out, a model-based foothold planner has to be established first.

1) LIPM-based foothold planner: Assuming a constant CoM height z_0, the LIP model yields $\omega_0 = \sqrt{g/z_0}$, and then the Instantaneous Capture Point as $\xi_x = x + \dot{x}/\omega_0$ and $\xi_y = y + \dot{y}/\omega_0$. When step transition occurs, the planner proposes the next foothold target $u_{\text{init}} = \Pi(\xi, v_{\text{cmd}}, \dot{\psi}_{\text{cmd}}, \text{phase})$. This keeps nominal stepping physics explicit without task-specific hard constraints. Here (x, y) and $(\dot{x}, \dot{y})$ denote the current CoM position and velocity, and $\Pi(\cdot)$ maps the ICP, the commanded velocity $(v_{\text{cmd}}, \dot{\psi}_{\text{cmd}})$, and the gait phase to the foothold target u_{init}.

2) Foothold tracker training on flat ground: Stage-I trains the low-level policy to track footholds and remain reliable under small target disturbances. During training, a bounded perturbation is injected as $\tilde{u}_{\text{init}} = u_{\text{init}} + \varepsilon$, where $\varepsilon = (\delta x, \delta y, \delta \psi) \subset \mathbb{R}^3$ is defined in the body frame as

$$\begin{aligned} \varepsilon \sim \mathcal{D} &= \text{Unif}\big([-\delta_x, \delta_x] \times [-\delta_y, \delta_y] \times [-\delta_\psi, \delta_\psi]\big), \\ \delta_x &= \delta_y = 0.05 \text{ m}, \qquad \delta_\psi = 20° \ (\approx 0.349 \text{ rad}) \end{aligned} \quad (1)$$

Note that the perturbation is applied only to the swing foot and is held constant until touchdown. Overall, the policy observes proprioception information (IMU signals and joint states) and current step phase, and outputs joint position commands to the joint PD controller. The reward function emphasizes accurate foothold realization and stable contact scheduling, with light regularization for smoothness and safety. The reward terms are listed in Table IV in the appendices, of which the key ones are summarized below.

(i) step_tracking: We reward correct stance leg alternation and precise swing leg foot placement at touchdown as

$$\begin{aligned} r_{\text{sched}} = w_{\text{alt}} \left(\mathbb{I}_R - \mathbb{I}_L\right) s \\ + w_{\text{pos}} \phi_1\big(\|\mathbf{p}_{\text{sw}} - \mathbf{p}_{\text{tgt}}\|\,; a_p\big) \\ + w_{\text{yaw}} \phi_1\big(|\psi_{\text{sw}} - \psi_{\text{tgt}}|\,; a_\psi\big) \end{aligned} \quad (2)$$

where $\mathbb{I}_R, \mathbb{I}_L \in \{0, 1\}$ are right and left foot contact indicators at touchdown, $s \in \{-1, +1\}$ is the sign for stance leg alternation based on step phase, $\mathbf{p}_{\text{sw}}$ and $\mathbf{p}_{\text{tgt}}$ are the actual and desired foothold positions, ψ_{sw} and ψ_{tgt} are the actual and desired foothold orientations, and ϕ_1 is defined in Table VI in the appendices. The corresponding scalings used in this term are $w_{\text{alt}} = 1$, $w_{\text{pos}} = 5$, $a_p = 1$, $w_{\text{yaw}} = 0.5$ and $a_\psi = 1$.
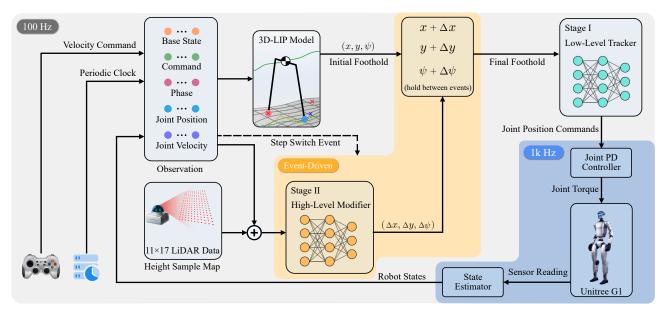
Fig. 2: The proposed framework for humanoid robot traversing narrow paths. A two-stage training curriculum is designed for the low-level foothold tracker and high-level foothold modifier. Different background colors indicate different operational frequencies.

(ii) tracking_lin_vel_world: We penalize the error between commanded and measured base linear velocities in the world frame (normalized by the command magnitude), fostering faithful velocity following.

(iii) base_heading & base_z_orientation: We align the base heading angle to the commanded value, and penalize base tilt indicated by projected gravity, stabilizing the base orientation for clean foot placement.

(iv) joint_regularization: We add a soft regularizer on hip and waist yaw angles and leg abduction/adduction angles to keep them near neutral, avoiding extreme poses during locomotion.

C. Stage-II Training for Safe Foothold Modifier

Stage-II trains a high-level exteroception-based foothold modifier on narrow paths that refines the template-based foothold target from Stage-I. The objective is to prioritize safe and precise foot placement under narrow support, without any manual lateral locking.

1) Anterior terrain height map: To keep sensing lightweight and deployment-friendly, the foothold modifier consumes a compact anterior terrain height map aligned with the body frame. The body frame is defined with x pointing forward and y pointing leftward. The terrain height map is sampled within a fixed area in front of the robot and flattened into a vector in the order of from near to far and from left to right. The dimension of this fixed area is designed to be $x \in [0.1, 1.1]\,m$ and $y \in [-0.8, 0.8]\,m$. The sampling resolution is uniformly $0.1\,m$ along both axes, resulting in 11×17 grid points. In simulation, height of these grid points are queried from the terrain height field. In physical experiments, a LiDAR-based mapper produces the same map to ensure consistency.

2) Foothold modifier training on narrow paths: At each step transition, the foothold modifier outputs a body-frame residual $\Delta u = (\Delta x, \Delta y, \Delta \psi) \in \mathbb{R}^3$ for the swing foot only. The final foothold target for foot $i \in \{\mathrm{L}, \mathrm{R}\}$ is calculated as

$$u_{\text{final}}^{(i)} = \begin{cases} u_{\text{init}}^{(i)} \oplus \text{sat}_S(\Delta u), & \text{if } i \text{ is swing}, \\ u_{\text{init}}^{(i)}, & \text{if } i \text{ is stance}, \end{cases} \quad (3)$$

$$\text{sat}_S(\Delta u) = \text{clip}(\Delta u, -S, S)$$

where $\oplus$ denotes pose composition in the foot task space and $S = (s_x, s_y, s_\psi)$ sets component-wise foothold adjustment bounds.

To explicitly reflect locomotion safety and preserve the interpretability from Stage-I policy, the reward terms for Stage-II training are designed as listed in Table V, of which the key ones are summarized below.

(i) foothold_safety: We encourage foothold targets that are within the narrow path and locally flat. We penalize situations include (i) "abyss", where terrain height is below a safe threshold, and (ii) lack of local flatness around the target foothold, expressed as

$$r_{\text{footstep\_safety}} = -\mathbb{I}\{ h(\mathbf{u}_{\text{final}}) < u_{\text{th}} \} \\ - \text{Var}\{ h(\mathbf{u}) : \mathbf{u} \in \mathcal{N}(\mathbf{u}_{\text{final}}) \}. \quad (4)$$

where $h(\mathbf{u})$ outputs the terrain height at location $\mathbf{u}$, $\mathbf{u}_{\text{final}}$ is the foothold target after modification, $\mathcal{N}(\mathbf{u}_{\text{final}})$ represents a small grid patch with $\mathbf{u}_{\text{final}}$ at its center for assessing local flatness, and u_{th} is a safety threshold for identifying valid foothold regions.

(ii) beam_balance: We encourage centerline adherence by applying a Gaussian shaping to the foothold's lateral deviation, resulting in higher reward for smaller deviation (Fig. 3).
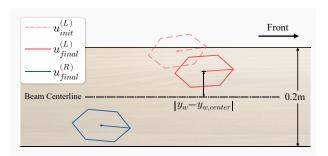
Fig. 3: Demonstration of foothold modification in the Stage-II training. The dashed polygon shows the initial foothold target from the foothold planner for the left leg, $u_{\text{init}}^{(L)}$, while the solid polygons denote the final foothold targets from the foothold modifier for both legs, $u_{\text{final}}^{(L)}$ (red) and $u_{\text{final}}^{(R)}$ (blue). A smooth Gaussian penalty is given to the foothold's lateral offset from the beam centerline, $|y_w - y_{w,\text{center}}|$, to encourage staying near the centerline.

(iii) forward_progress: We reward forward movement along the narrow path only and discourages the other way.

(iv) face_forward: We encourage alignment between foot orientation and forward direction by applying a shaping function to the foothold's yaw angle, so that a smaller yaw angle yields a larger reward.

(v) feet_proximity: We penalize excessively small distance between the feet along the direction of narrow path to avoid leg interference.

(vi) action_magnitude & action_smoothness: We penalize the magnitude of foothold residual and its step-to-step variation to keep the refinement minimal and smooth.

During training, the observation of the high-level modifier policy concatenates: (i) the proprioception informaiton including the IMU signals and joint states, (ii) the step phase features, (iii) current foothold target from the template-based planner, and (iv) the flattened anterior terrain height map. Notably, even though no centerline locking or lateral offset is manually added, safe and precise footholds emerge from the compact terrain perception based locomotion control.

IV. EXPERIMENTS

A. Setup

The proposed approach is evaluated first in simulation and then on a Unitree G1 humanoid robot. Following Section III, the settings are kept the same between simulation and experiments to ensure apples-to-apples comparison. Both the low-level and high-level policies are exported as TorchScript and executed on the onboard computer in Unitree G1. Standard safety guards, including torque limits, fall/edge detectors and foot–to-foot clearance checks, are enabled during all trials.

As with narrow paths, straight beams with width of $\{0.15, 0.20, 0.25\}\,m$ and length between 3–$5\,m$ are used during training in simulation. In physical experiments, the narrow path is set to be a wooden beam of $0.2\,m$ wide and $3\,m$ long placed on level ground. For each setting, 20 independent trials were run with a standardized initial pose

and command. For both simulated and physical robot, a trial is deemed successful if the robot reaches the beam end within a time/step limit, with all footholds' centers remain within the beam boundaries and no falls, while a trial is considered a failure if any foothold's center exceeds the beam edge, the torso violates the attitude limits, or a protective stop is triggered by excessive joint position or velocity.

B. Simulation Evaluation

The proposed approach is evaluated in simulation against two baselines and through one ablation study.

Baselines: (i) **No-Modifier**: This baseline tracks the foothold from the LIPM-based planner with no residual refinement. Consequently, the controller is the same as the Stage-I policy in the proposed full method. This baseline tests the benefit of the high-level modifier. (ii) **RL-Only**: This baseline further ditches the reduced-order model based foothold planner and utilize an end-to-end RL framework for locomotion control. The learned policy receives exactly the same observation as the proposed method but directly outputs desired joint positions at the $100\,Hz$ control rate, which are then realized by the joint PD controller at $1\,kHz$. The training uses the same reward function as that in the Stage-II training of the proposed method, except the terms that refer to the template-based foothold target are omitted. Besides, training budgets, evaluation episode counts and random seeds are kept the same to yield a fair comparison.

Ablation study: The effect of the small random perturbations added to the LIPM-based foothold targets in the Stage-I training are studied through ablation. The resulted policy is trained identically to the proposed method except without these perturbations. This ablation study tests whether target disturbance during training yields a foothold tracker that better tolerates target variations at step transitions, thus improving foot-placement RMSE, centerline following and success rate on narrow path traversal tasks.

The evaluation in simulation uses three metrics: success rate (%), centerline deviation (m), and foot-placement RMSE (m). Each metric is evaluated over 20 episodes. The results for the baseline comparisons and the ablation study are summarized in Tables I and II, respectively. It can be seen that, across different path widths and lengths, the proposed method can improve success rate and reduce centerline deviation compared to the other two baselines. As with the foot-placement RMSE, it is computed with respect to the commanded foothold target u_{cmd}. For the *No-Modifier* baseline, perturbation from the modifier's residual is disabled at test time, so that $u_{\text{cmd}} = u_{\text{init}}$ and the tracker can follow clean foothold targets with small RMSE. However, the commanded foothold target in our proposed method is calculated as $u_{\text{cmd}} = u_{\text{final}} = u_{\text{init}} \oplus \Delta u$, with perturbation from the nonzero residual Δu. Therefore, realizing these targets leads to a slightly larger foot-placement RMSE. On the other hand, results from the ablation study indicate that removing Stage-I foothold disturbances leads to poorer tracking performance at step transition and larger foot-placement errors.

TABLE I: Baseline comparison results from walking on beams in simulation (mean ± std over 20 runs).

Method	Success rate (%)	Centerline dev. (m)	FP-RMSE (m)
No-Modifier	15	0.04690 ± 0.00057	0.01962 ± 0.00079
RL-Only	0	0.18192 ± 0.07075	—
Ours	**100**	**0.01639 ± 0.00117**	**0.02633 ± 0.00083**

TABLE II: Ablation study results from walking on a $0.20\,m$-wide beam in simulation (mean ± std over 20 runs).

Configuration	Success (%)	Centerline dev. (m)	FP-RMSE (m)
w/o Stage-I disturbances	50	0.09696 ± 0.07876	0.05467 ± 0.05467
Ours	**100**	**0.01639 ± 0.00117**	**0.02633 ± 0.00083**

It is worth noting that one dominant failure mode can be observed in the process of training. When pronounced heading angle oscillation occurs on the robot, the feasible foothold set in the local terrain height map shrinks. Consequently, the initial foothold target from the planner policy can drift toward the path edge and drive the residual from the modifier policy to reach its bounds ($\|r\| \to S$), yet still unable to pull the final foothold target back onto the narrow path. This leads to a failure with off-path footholds.

C. Hardware Validation

As mentioned in the setup, the full proposed method is validated on a Unitree G1 humanoid robot. 20 independent trials were carried out. During each trial, a body-centric anterior terrain height map with the same dimension and resolution as in training is constructed online using the onboard LiDAR stream, as shown in Fig. 4. The pipeline for building the height map is: (i) transform raw data points into the robot's body frame using the IMU's gravity perception and crop the data to the fixed region of interest (ROI), (ii) grid the ROI at $0.1\,m$ resolution, use an adaptive binning square for each grid point (the adaptive binning square starts with $0.1\,m$ side length and expand up to $0.3\,m$ by $0.05\,m$ until nonempty), and take the maximum height $z_{\max}$ inside the square as the estimated height, (iii) clamp the heights to $[0.7, 1.4]\,m$ to mitigate sparse returns and sensor bias, and (iv) flatten the 11×17 height map to a vector and publish at a fixed rate synchronized with the modifier queries.

Two task-level metrics are used to evaluate the robot's experimental performance on the $3\,m \times 0.2\,m$ beam. Besides the success rate used in simulation evaluation, traversal rate is selected and defined as the fraction of the beam length completed before failure. The traversal rate is calculated as $r_i = \min(1, d_i/L_{\text{beam}})$ per trial and then averaged over all trials. For a head-to-head comparison, the state-of-the-art results from **BeamDojo** [3] are referred to as the baseline for the experiments. It can be seen from Table III that the proposed method in this paper obtains higher success rate and traversal rate along the beam.

During bring-up and pilot runs in experiments, height-estimation bias near the path boundary was occasionally observed, which could nudge the foothold target toward the
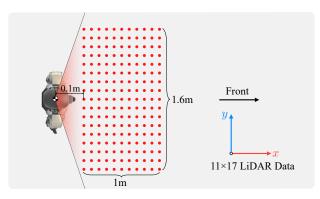

Fig. 4: Top view of the anterior terrain height map based on the robot's LiDAR sensor. The map spans $x \in [0.1, 1.1]\,m$ and $y \in [-0.8, 0.8]\,m$ with a resolution of $0.1\,m$ uniformly, providing a real-time terrain representation for foothold modification.

TABLE III: Hardware comparison results from walking on a $3\,m \times 0.2\,m$ beam in reality (mean ± std). Ours: N=20 trials; BeamDojo: N=5 trials as reported.

Setting	Success rate (%)	Traversal rate (%)
BeamDojo [3] (G1, real beam)	80	88.16
Ours (G1, real beam)	**100**	**100**

boundary. To mitigate this defect, several measures were performed, including applying robust per-grid statistics (median with outlier rejection), mild temporal smoothing of the height map and conservative residual bounds. Consequently, the evaluation runs reported in Table III did not exhibit this failure any more.

V. CONCLUSION

This paper proposes a two-stage reinforcement learning framework for robust and safe humanoid walking control when traversing narrow paths. The key insights indicated by the results include (i) foothold target disturbances added in Stage-I is crucial to reducing touchdown errors and off–beam steps and (ii) a small and smooth foothold residual added in Stage-II to the LIPM-based foothold improves success rate, centerline adherence and safety margins when traversing narrow paths. Besides, a compact anterior terrain height map is sufficient for foothold decisions and sim–to–real transfer simplification. No heavy vision pipelines are needed.

In the future, we will (i) extend the proposed framework beyond straight beams to more sparse-support terrains (e.g., stepping stones, gaps, curved beams), and (ii) expand the foothold representation to 3D with a nominal vertical profile z^* to enable traversing stairs and uneven terrains.

REFERENCES

[1] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," *IEEE*, 2007.

[2] J. He, C. Zhang, F. Jenelten, R. Grandia, M. BÄcher, and M. Hutter, "Attention-based map encoding for learning generalized legged locomotion," 2025. [Online]. Available: https://arxiv.org/abs/2506.09588

[3] H. Wang, Z. Wang, J. Ren, Q. Ben, T. Huang, W. Zhang, and J. Pang, "Beamdojo: Learning agile humanoid locomotion on sparse footholds," 2025. [Online]. Available: https://arxiv.org/abs/2502.10363

[4] H. J. Lee, S. Hong, and S. Kim, "Integrating model-based footstep planning with model-free reinforcement learning for dynamic legged locomotion," 2024. [Online]. Available: https://arxiv.org/abs/2408.02662

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, 2003, pp. 1620–1626 vol.2.

[6] T. Koolen, T. D. Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[7] A. L. Hof, "The 'extrapolated center of mass' concept suggests a simple control of balance in walking." *Hum Mov*, vol. 27, no. 1, pp. 112–125, 2008.

[8] P. B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," *IEEE*, 2006.

[9] H. Diedam, D. Dimitrov, P. B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2009.

[10] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic foot step placement," *Advanced Robotics*, vol. 24, pp. 719–737, 04 2010.

[11] Y. Ding, C. Khazoom, M. Chignoli, and S. Kim, "Orientation-aware model predictive control with footstep adaptation for dynamic humanoid walking," 2022. [Online]. Available: https://arxiv.org/abs/2205.15443

[12] M. Dai, X. Xiong, and A. Ames, "Bipedal walking on constrained footholds: Momentum regulation via vertical com control," 2021. [Online]. Available: https://arxiv.org/abs/2104.10367

[13] R. J. Griffin and A. Leonessa, "Model predictive control for stable walking using the divergent component of motion with footstep location and yaw adaptation," *International Journal of Humanoid Robotics*, vol. 16, no. 05, pp. 279–286, 2019.

[14] G. Gibson, O. Dosunmu-Ogunbi, Y. Gong, and J. Grizzle, "Terrain-adaptive, alip-based bipedal locomotion controller via model predictive control and virtual constraints," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2022, p. 6724–6731. [Online]. Available: http://dx.doi.org/10.1109/IROS47612.2022.9981969

[15] Z. Xiang, V. Paredes, G. A. Castillo, and A. Hereid, "Adaptive step duration for precise foot placement: Achieving robust bipedal locomotion on terrains with restricted footholds," 2024. [Online]. Available: https://arxiv.org/abs/2403.17136

[16] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling, "Residual policy learning," 2019. [Online]. Available: https://arxiv.org/abs/1812.06298

[17] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2019, p. 6023–6029. [Online]. Available: https://doi.org/10.1109/ICRA.2019.8794127

[18] Z. Xie, H. Y. Ling, N. H. Kim, and M. van de Panne, "Allsteps: Curriculum-driven learning of stepping stone skills," 2020. [Online]. Available: https://arxiv.org/abs/2005.04323

[19] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation mapping for locomotion and navigation using gpu," 2022. [Online]. Available: https://arxiv.org/abs/2204.12876

[20] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, Jan. 2022. [Online]. Available: http://dx.doi.org/10.1126/scirobotics.abk2822

[21] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.adi7566

APPENDIX

A. Reward Functions

The reward functions used during the two-stage training process are shown in Tables IV and V. The corresponding symbols and their definitions are provided in Tables VI.

TABLE IV: Stage-I rewards.

Term	Weight	Equation
step_tracking	3.0	$(\mathbb{I}_R - \mathbb{I}_L)\, s \,\times\, \phi_1(\|\delta_p\|_2;\ a_p=1)$ $\times\ \phi_1(|\delta_\psi|;\ a_r=1)$
tracking_lin_vel_world	4.0	$\phi_2\big(\| (v^c_{xy} - v_{xy}) \odot (1 + |v^c_{xy}|)^{-1} \|_2;\ a_v=1\big)$
base_heading	3.0	$\phi_1(|\mathrm{wrap}(\psi^c - \psi)|;\ a_\psi=\frac{\pi}{2})$
base_z_orientation	1.0	$\phi_2(\|g_{xy}\|_2;\ a_g=0.2)$
base_height	1.0	$\phi_2(h - h^*;\ a_h=1)$
joint_regularization	1.0	$\frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}\phi_2(q_j;\ a_q=1)$
lin_vel_z	1×10^{-1}	$-v_z^2$
ang_vel_xy	1×10^{-2}	$-\|\omega_{xy}\|_2^2$
dof_vel	1×10^{-3}	$-\|\dot{q}\|_2^2$
torques	1×10^{-4}	$-\|\tau\|_2^2$
actuation_rate	1×10^{-3}	$-\|a_t - a_{t-1}\|_2^2/\Delta t^2$
actuation_rate2	1×10^{-4}	$-\|a_t - 2a_{t-1} + a_{t-2}\|_2^2/\Delta t^2$
dof_pos_limits	10	$-\sum_j [(\ell_j - q_j)_+ + (q_j - u_j)_+]$
torque_limits	1×10^{-2}	$-\sum_j (|\tau_j| - 0.8\,\tau_j^{\max})_+$

TABLE V: Stage-II rewards.

Term	Weight	Equation
foothold_safety	1.0	$-5 \sum_{f\in\{L,R\}} \mathbb{I}\{ h(\mathbf{p}_t^f) < -0.20 \}\, m_{\mathrm{swing}}^f$
beam_balance	1.0	$\exp\big(-(|y - y_c|/\sigma_y)^2\big) - 1$
feet_proximity	0.1	$-\dfrac{(d_{\min} - |x_R - x_L|)_+}{d_{\min}}$
forward_progress	1.0	$\max(0,\ x_t - x_{t-1})$
face_forward	0.1	$\max(0,\ 1 - |\mathrm{wrap}(\psi)|/\pi)$
contact_schedule	1.0	$(\mathbb{I}_R - \mathbb{I}_L)\, s \,\times\, \phi_1(\|\delta_p\|_2;\ a_p=1)$ $\times\ \phi_1(|\delta_\psi|;\ a_r=1)$
tracking_lin_vel_world	2.0	$\phi_2\big(\| (v^c_{xy} - v_{xy}) \odot (1 + |v^c_{xy}|)^{-1} \|_2;\ a_v=1\big)$
base_heading	0.2	$\phi_1(|\mathrm{wrap}(\psi^c - \psi)|;\ a_\psi=\frac{\pi}{2})$
base_z_orientation	0.5	$\phi_2(\|g_{xy}\|_2;\ a_g=0.2)$
base_height	0.2	$\phi_2(h - h^*;\ a_h=1)$
action_magnitude	0.01	$-\|\mathbf{r}_t\|_2^2$
action_smoothness	0.01	$-\|\mathbf{r}_t - \mathbf{r}_{t-1}\|_2^2$

TABLE VI: Used symbols and constants for reward tables.

Symbol	Definition / Value
$\phi_1(e;a)$	$\exp(-|e|/(a\,\sigma))$
$\phi_2(e;a)$	$\exp(-(e/a)^2/\sigma)$
σ	tracking shape scale = 0.25
h^*	base height target = 0.78 m
$\mathcal{J}$	joints $\{0, 1, 5, 6\}$ (hip/waist yaw/ab-ad soft centering)
$\delta_p,\ \delta_\psi$	swing-foot pos./yaw error at touchdown (to target)
s	gait schedule sign; $\mathbb{I}_R$, $\mathbb{I}_L$: contact indicators
y_c	beam centerline; $\sigma_y = 0.1$ m
$d_{\min}$	inter-foot distance threshold along x, = 0.1 m
$\mathbf{r}_t$	footstep residual $(\Delta x, \Delta y, \Delta \psi)$
$(\cdot)_+$	$\max(\cdot, 0)$; Δt: control step

B. Domain Randomization

The parameter settings for domain randomization are provided in Table VII.

C. Hyperparameter

The hyperparameter values used in the two-stage training process can be found in Tables VIII and IX.

TABLE VII: Domain Randomization Setting

Term	Value
Observations	
angular velocity noise	$\mathcal{U}(-0.2,\ 0.2)$ rad/s
projected gravity noise	$\mathcal{U}(-0.05,\ 0.05)$
joint position noise	$\mathcal{U}(-0.01,\ 0.01)$ rad
joint velocity noise	$\mathcal{U}(-0.01,\ 0.01)$ rad/s
height measurement noise	$\mathcal{U}(-0.10,\ 0.10)$ m
Humanoid Physical Properties	
payload mass (added mass)	$\mathcal{U}(-1.0,\ 1.0)$ kg
external push (interval / max vel)	every 2.5 s, $\|\mathbf{v}_{xy}\| \leq 0.5$ m/s
Terrain Dynamics	
friction coefficient	$\mathcal{U}(0.5,\ 1.25)$
restitution	fixed $= 0$
Elevation Map	
window/grid (sim & real)	fixed ROI, fixed grid; no DR
measurement noise (map)	$\mathcal{U}(-0.10,\ 0.10)$ m

TABLE VIII: Stage-I Training Hyperparameters

Term	Value
Rollout / Runner	
parallel envs	4096
steps per env	24
rollout size / update	4096×24 samples
max iterations	5000
save interval	100
episode length	5 s
policy / algo class	`ActorCritic` / `PPO`
PPO / Optimization	
learning rate	1×10^{-5} (`schedule=adaptive`)
num learning epochs	5
num mini-batches	4
clip range	0.2
entropy coef	0.01
value loss coef	1.0 (clipped value: `True`)
discount γ	0.99
GAE λ	0.95
desired KL	0.01
max grad norm	1.0

TABLE IX: Stage-II Training Hyperparameters

Term	Value
Rollout / Runner	
parallel envs	1024
steps per env	*inherited (not overridden)*
max iterations	10000
save interval	100
policy init noise	1.0
action space	residual $(\Delta x,\ \Delta y,\ \Delta \psi)$, dim $= 3$
control decimation	10
PPO / Optimization	
learning rate	1×10^{-5} (`schedule=adaptive`)
num learning epochs	5
num mini-batches	4
clip range	0.2
entropy coef	0.01
discount γ	0.99
GAE λ	0.95
desired KL	0.01
max grad norm	1.0