COLLISION-FREE IS NOT ENOUGH: PASSAGE-CONDITIONED NAVIGATION FOR WHEELED-LEGGED ROBOTS

Anonymous authors
Paper under double-blind review

ABSTRACT

A collision-free detour can still leave a robot unable to negotiate the next stair or ledge. For wheeled-legged robots, negotiating terrain depends on approach geometry, speed, and gait as well as obstacle clearance. We introduce passage-conditioned navigation: a persistent reference combines demonstrated geometry and gait events with configured rules, while observations and execution feedback guide local adaptation with a fixed locomotion library. On a teaching-compiled route, event-conditioned handling reduces mean lateral entry error from 13.43 to 0.56 cm across 98 deterministic kinematic rollouts. In matched contact-simulation perturbation cases across six segments of a separate, frozen mapping-derived reference, rejoining reduces mean terminal route deviation from 1.14 m to 0.11 m and raises ordered completion from 0% to 59.4%. Across 2,800 contact-simulation trials, the live-sensing reference achieves 62.6% completion, exceeding two graph-based route alternatives by 18.6 and 19.0 percentage points with the same executor. An outdoor deployment traverses successive terrain transitions using factory gait controllers; the navigation stack also runs on an Apple M1 laptop. A geometry stress test exposes limits to transfer. Together, these results show how stored passage conditions and reference-based recovery support local adaptation, and where controller and observation interactions still constrain reliability.

1 INTRODUCTION

When a wheeled-legged robot detours around an obstacle, avoiding collision is only part of the problem: the new approach may leave too little distance to slow down, require a gait change before the controller is ready, or bring the robot obliquely to the next step. These interactions are common on inspection routes that connect stairs and ledges with loose surfaces, slopes, and paved paths. A successful passage therefore contains experience about how to enter and traverse a transition, as well as the sequence of positions that leads through it.

Reusing that experience creates a tension. Repeating the taught track cannot accommodate every local change, while repairing its geometry can discard conditions that made the next passage possible. We ask: *how can a robot adapt to current local changes while retaining the passage conditions supplied by past experience?* The question connects teach-repeat-replan and integrated wheeled-legged navigation (Gao et al., 2019; Lee et al., 2024). We address it through a route-side representation and its interaction with an existing locomotion library.

The compiled reference retains a route corridor, selected approach geometry, speed limits, and gait intervals. It persists across executions. A separate runtime state retains recent observations, progress, the selected lateral transition, and gait handoff status. These two time scales are distinct from spatial scope: a route is spatially global, but its stored conditions can constrain a single local transition. The planner uses the reference to generate candidate curves and current evidence to assess them. Execution logic then coordinates speed, selected ledge-entry checks, and confirmed gait changes (Figure 3).

Our contribution is a passage-conditioned reference and runtime interface that keep corridor, approach, speed, and gait information available during local adaptation. The compiler separates demonstration-derived fields from configured rules, making their runtime use explicit. We evalu-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

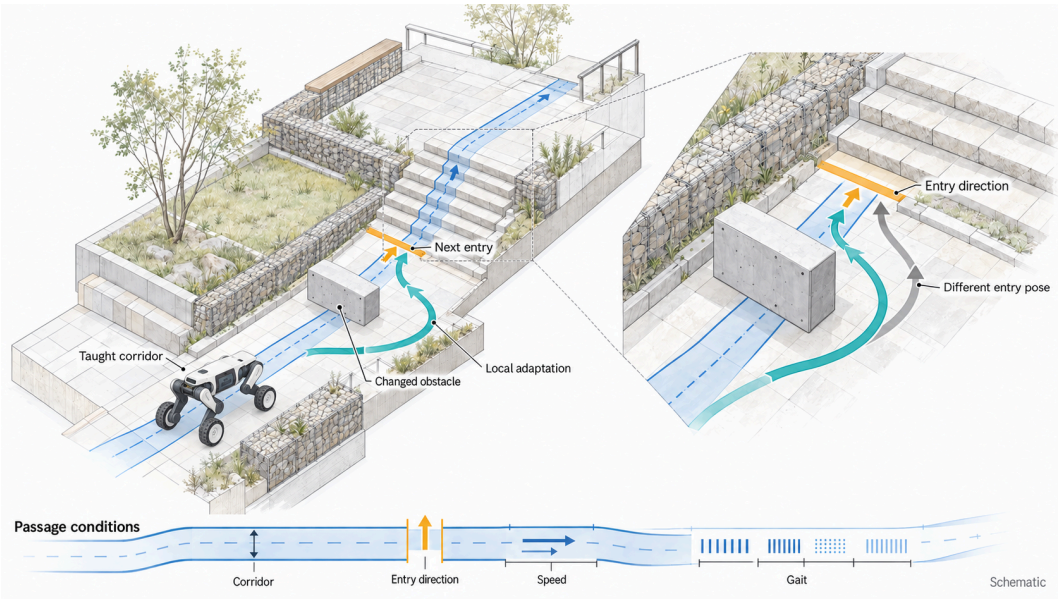

Figure 1: **A route may change while the next passage still imposes conditions.** A new obstacle interrupts the taught corridor. The teal detour approaches the next transition in the illustrated direction; the gray alternative reaches it with a different entry pose. The reference retains corridor, entry, speed, and gait information. Terrain and candidate paths are schematic.

ate this interface through recorded compilation, controlled ledge approaches, matched component removals, and route-rejoining experiments with shared locomotion actors.

The evaluation combines recorded teaching, outdoor traversal footage, and controlled simulation to connect stored conditions with measurable execution behavior. We examine terrain-prior selection and fallback entry rules alongside recovery.

2 Related Work

Teaching, route reuse, and replanning. Teach-Repeat-Replan converts imperfect demonstrated flight trajectories into smooth, kinodynamically feasible references and replans around unmapped or moving obstacles; its experiments include indoor and outdoor flight (Gao et al., 2019). LiDAR teach-and-repeat in difficult ground environments (Baril et al., 2022), adaptive route following (Sehn et al., 2022), and trajectory-based predictive control (Rosolia & Borrelli, 2020) also show that route reuse and adaptation are established ideas. In particular, Sehn et al. (2022) plan in a curvilinear space and penalize deviation from the taught path to retain prior terrain knowledge. Building on route-relative planning, we focus on the compilation and consumption of selected entry geometry and gait intervals for wheeled-legged terrain transitions.

Wheeled-legged planning and locomotion. Whole-body MPC with online gait generation (Bjelonic et al., 2021) and motion libraries with online MPC (Bjelonic et al., 2022) coordinate mobility and feedback. Lee et al. (2024) combine learned locomotion, mobility-aware local navigation, and large-scale planning, with kilometer-scale urban deployments. Their navigation and locomotion policies are learned and tightly coupled. We study an algorithmic compiler and runtime interface that retain task-specific passage conditions while reusing an existing gait library. This interface makes route preparation and execution decisions explicit.

Path and policy conditioning. Path-conditioned learned planning (Haro et al., 2026) and policy-dependent traversability (Yan et al., 2026) account for reference guidance and controller-dependent terrain capability. Our focus is the transfer of demonstrated route geometry and gait events into a

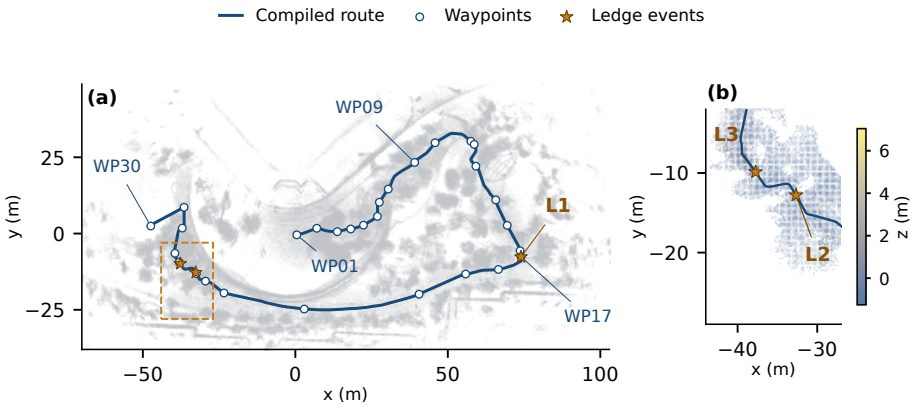

Figure 2: **A field map with an event-associated task reference.** (a) The compiled route and all 29 waypoint marks in a route-centred view of the field map; selected waypoint identifiers are labelled. (b) Ground-cloud detail around two retained ledge events, coloured by map height. The three event positions are stored in the route bundle. Points show mapped geometry and the line shows the compiled reference.

reference used for local curve repair, entry checks, and confirmed mode transitions. Our controlled comparisons use the same locomotion actors and evaluation protocol across route alternatives.

3 Task and Observation Interfaces

Inputs are a point-cloud-derived map M_0, ordered goal regions W, an existing locomotion library $\mathcal{G}$, and K real teaching episodes $\mathcal{D}_H^{(K)}$. A reference $\Pi^0 = (\gamma^0(s), v^0(s), g^0(s))$ organizes geometry, speed limits, and gait intervals by route coordinate s. The high-level expert procedure consists of explicit rules and geometric operations. Online mode selection schedules controllers from the fixed library.

The teaching compiler directly consumes pose trails, waypoint marks, gait events, and map geometry. Additional sensor topics may also be archived. Speed limits are assigned during compilation and configuration; the teaching input supplies route geometry and gait events. The contact-based simulation branch loads a frozen route derived from a mapping trajectory and map annotations. Execution reads the stored route and terrain fields. Physical execution consumes localized pose, LiDAR-derived observations, IMU attitude, and controller feedback. Demonstration selection, corridor widths, approach extents, and geometric thresholds are configured by the operator.

Field goals use body intersection with a $0.25\,\mathrm{m}$ waypoint disk; simulation requires the robot centre to reach each goal within $0.20\,\mathrm{m}$ in order. The compiler's $0.10\,\mathrm{m}$ waypoint-fit tolerance is a separate geometric check. Goal coordinates and ordering are frozen within each evaluation (Appendix H).

Figure 2 locates the ordered task and the retained passage events in the same coordinate frame as the field point cloud. This spatial association allows the executor to activate local conditions using progress along a persistent route.

4 Method: Experience-Constrained Local Adaptation

Figure 3 separates the persistent reference from current observations and execution state. Terrain-evidence selection acts on the planner's observations; route and passage conditions remain inputs to candidate planning and coordination.

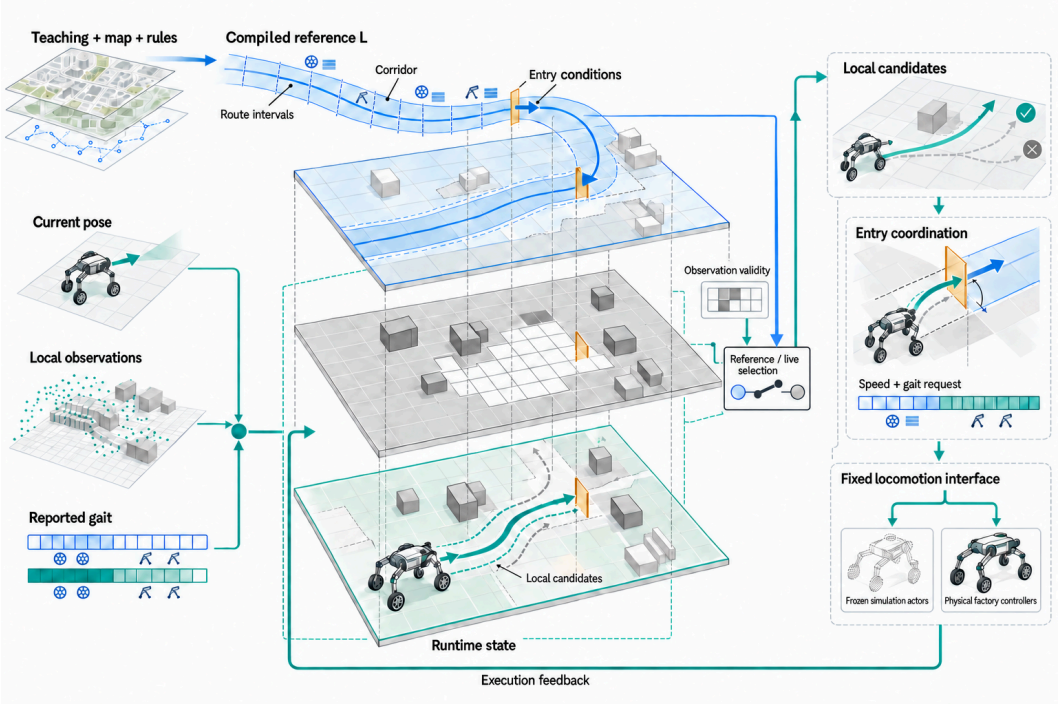

Figure 3: **Persistent passage conditions meet current execution state.** Teaching, map information, and configured rules produce L. The aligned layers depict reference geometry, local terrain observations, and candidate motion in the same region. The selector chooses terrain evidence while retaining the route's corridor, entry, speed, and gait fields for planning and execution coordination. Feedback updates runtime state S_t; updates to L occur through explicit recompilation and reload. Frozen simulation actors and physical factory controllers are separate interfaces.

4.1 Compile a persistent reference

The compiler returns a reference package

$$L = \text{Compile}_\alpha(M_0, W, \mathcal{D}_H^{(K)}) = (M_0, W, \Pi^0, \mathcal{K}_E), \tag{1}$$

where α contains explicit rule settings and $\mathcal{K}_E$ stores corridor, gait, approach, and switch conditions. The locomotion library $\mathcal{G}$ is paired with this package at validation and execution time. Demonstration and splice selection, corridor widths, and approach extents are configured through α.

The compiler first aligns passing waypoint marks with the selected pose trails and uses ordered waypoint anchors to delimit correction intervals. Where candidate corrections cover the same interval, a turning-based rule selects the replacement trail. The assembled polyline is resampled in cumulative planar arc length, $s_i = s_{i-1} + \|p_i - p_{i-1}\|_2$, and aligned to waypoint anchors. Corridor, body-sweep, and clearance checks then govern shortcuts and rounded joins; unsuccessful candidates can retain the taught geometry.

Platform-gait trail portions supply candidate ledge crossings. Local planar displacement defines their approach direction; clustered detections produce a crossing estimate that can be adjusted using the ground-cloud profile. A configured straight crossing lane is blended into the route only when the shift, waypoint-access, and body-sweep checks permit it. Gait-demo votes are projected to the assembled line, combined with map-derived steepness and configured platform extents, and converted into arc-aligned intervals. The resulting package writes centerline, gait, speed limit, detour permission, and corridor width for each segment, together with crossing and approach fields. Crossing position and direction are trail-derived with optional map adjustment; approach extents, tolerances, speed caps, and retry rules are configured. This shared route coordinate is what lets a later local decision refer to a terrain transition recorded during teaching.

A route-prior observation from a recorded passage can relax an overly conservative map classification under the recorded speed and approach conditions. The contact-based simulation experiments load a separately prepared reference through the route interface and evaluate runtime behavior on that reference. Acceptance checks reject a candidate; hazard and skew diagnostics record additional geometric conditions. We represent acceptance checks as $\mathcal{K}_{\mathrm{hard}}$ and supplementary hazard diagnostics as $\mathcal{K}_{\mathrm{warn}}$. The field build runs a separate body-sweep clearance check on each generated bundle before its kinematic route check.

4.2 Use short-lived state to repair a route

Let o_t denote actually consumed local observations, $\hat{x}_t$ the localized pose, e_t execution reports, and S_t recent state. The runtime interfaces are

$$S_{t+1} = \mathrm{Update}(S_t, o_t, \hat{x}_t, e_t), \tag{2}$$

$$(\gamma_t, v_t^{\mathrm{req}}, g_t^{\mathrm{req}}) = \mathrm{Adapt}(L, S_{t+1}), \tag{3}$$

$$u_t = \mathrm{Execute}(\gamma_t, v_t^{\mathrm{req}}, g_t^{\mathrm{req}}, \hat{x}_t, g_t^{\mathrm{confirmed}}). \tag{4}$$

S_t includes bounded local-observation history, route progress, tracking error, selected lateral-transition state, and handoff/blocked timers. The observation deque is cleared by age or discontinuities; the plan anchor survives between control ticks to preserve a selected transition. A stale observation stops the follower. These runtime variables are bounded. The persistent bundle L remains fixed during execution and changes through explicit recompilation and reload.

On the sensor-live route-follower path, pose is projected into route-relative coordinates (s, d). Candidate curves are generated inside the current corridor with quintic lateral transitions and a return toward the next goal. Footprint sweeps query the local grid. For a candidate target offset d_j, the planner uses the cost

$$J_j = w_o\rho_j + w_f(1 - f_j) + w_l|d_j| + w_s|d_j - d_{\mathrm{prev}}| - w_p f_j + w_b\mathbf{1}[d_j d_{\mathrm{prev}} < 0], \tag{5}$$

where ρ_j is obstacle proximity and f_j is the free-length fraction. The planner rejects candidates with insufficient executable prefix length. A switching margin can retain the current candidate; the retained candidate's transition anchor persists between ticks to avoid continually restarting the lateral motion. Bounded A* is a fallback where detours are permitted. The persistent route defines the candidate coordinates and permitted corridor. Local evidence changes candidate validity, cost, and speed scaling.

4.3 Coordinate speed, gait, and confirmed execution

The follower bounds commands by segment speed limits, applies local clearance-dependent speed scaling, and requests the segment's gait. Heading control and entry braking further affect commands. Gait request and confirmation are distinct: the hardware-style handoff requests zero motion, waits for measured stillness, issues a switch, and waits for acknowledgement. The fixed simulation actors also consume the previous action and gait phase (Appendix J). Comparisons retain the same handoff procedure; the simulation adapter models acknowledgement explicitly. Gait-interval hysteresis retains the current interval under small backward changes in projected progress. Implementation thresholds are listed in Appendix H.

Preserve a ledge approach. A platform-gait interval retains a crossing position p_j and direction ψ_j. These define a local approach frame:

$$\begin{bmatrix}\ell_{j,t}\\ d_{j,t}\end{bmatrix} = R(-\psi_j)(p_t - p_j), \qquad \Delta\psi_{j,t} = \mathrm{wrap}(\psi_t - \psi_j). \tag{6}$$

At $\ell_{j,t} = -0.7\,\mathrm{m}$, the field configuration checks the nominal geometric condition $|d_{j,t}| \le 0.10\,\mathrm{m}$ and $|\Delta\psi_{j,t}| \le 10^\circ$. The condition is evaluated in the stored crossing frame, so displacement or turning after a local route change can trigger a new approach. It can pivot or reverse $0.3\,\mathrm{m}$ for another approach. After two retries, it permits crossing within a relaxed lateral bound of $0.25\,\mathrm{m}$ without rechecking heading, and otherwise holds. A late arrival beyond the check point can also clear the check. Entry conditions are enforced by these execution branches after local planning, with the stated retry and late-arrival exceptions. The approach-speed limit is configured separately from the command used in the platform gait.



Figure 4: **One physical deployment across successive terrain transitions.** Nine unretouched frames from a single outdoor run show the successive terrain transitions encountered by the wheeled-legged robot.

Select the terrain evidence used by the planner. On live-sensing segments, the planner consumes the local LiDAR-derived grid and scan. The field configuration substitutes route-profile observations on selected stairs/platform intervals and around ledge crossings. This lets the robot traverse familiar terrain that would be rejected by the flat-ground grid, while placing obstacle response in those intervals under the route prior. The implementation records this observation selection separately from route tracking and gait selection.

5 Evaluation

We evaluate how passage information is compiled, whether its runtime use changes approach geometry, and how local adaptation affects task completion. A kinematic diagnostic consumes the teaching-compiled reference; contact simulation tests recovery and route preparation on a separate reconstructed course. Physical footage provides a deployment case study across successive terrain transitions.

Simulation protocol. The MuJoCo navigation core uses frozen `J3100.onnx` and `stairs_1150.onnx` actors, distinct from the physical factory gait stack. Simulation references are adapted from a mapping-trajectory draft on a reconstructed course; the field reference below is compiled from teaching. Matched comparisons share actors, goals, sensors, and condition parameters, with observations depending on the resulting trajectory. Nominal and pose-noise conditions use, respectively, no injected noise and Gaussian standard deviations of 0.03 m in planar position and 1° in yaw. Earlier development batches also place a physical box absent from the preparation map but visible to the sensor model. Success requires software completion and visiting all goal centres in order within 0.20 m. Falls, blocked states, and timeouts remain in the reported outcomes.

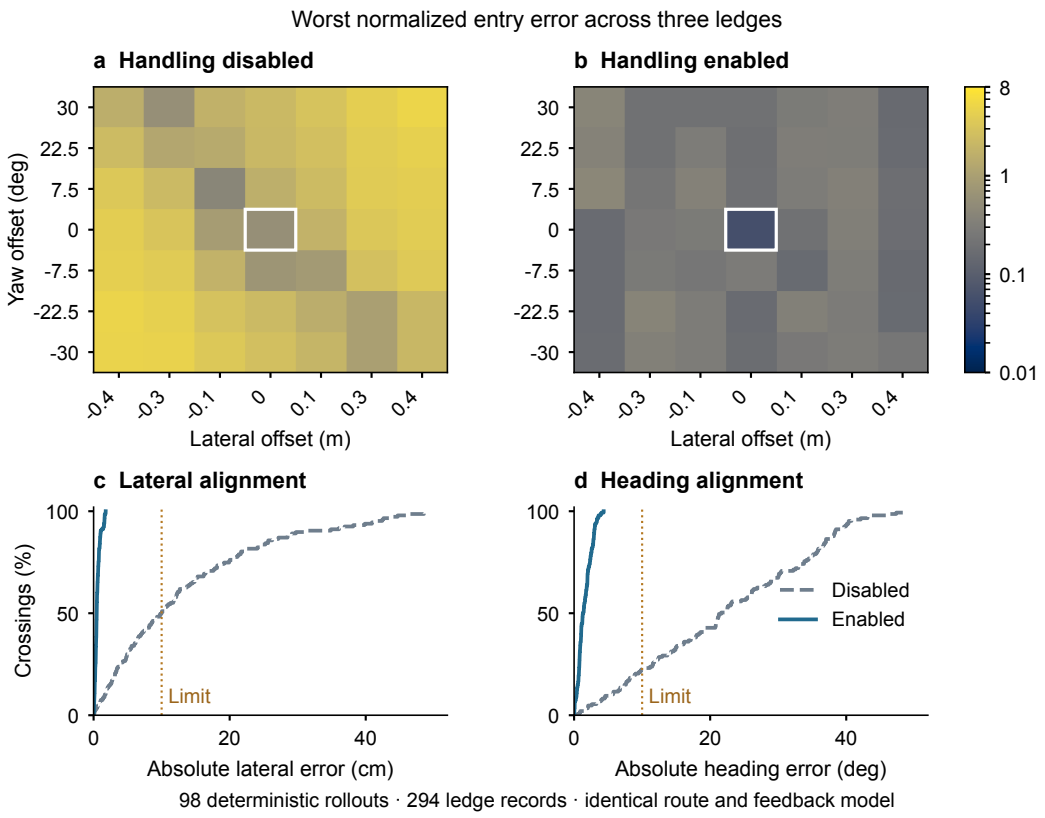

Figure 5: **Stored events organize precise approaches.** Top: the worst normalized error among the three ledges in each offset cell, on a shared logarithmic colour scale. For each crossing, $E = \max(|d|/(0.10\,\mathrm{m}), |\Delta\psi|/(10^\circ))$; $E \leq 1$ meets both configured limits. White boxes mark zero-offset controls. Bottom: cumulative distributions of absolute crossing errors over all records. The kinematic model uses delayed commands and synthetic gait feedback.

5.1 Teaching compilation

The field reference can be traced to three complete pose trails, two gait-labelled trails drawn from those recordings, and eleven candidate local corrections. Five corrections are selected as splices in the final bundle. After splice selection and geometric processing, the compiled reference contains 270.9 m of centerline, 29 waypoint entries, and 28 segments. These quantities describe the resulting bundle, whose task definition omits one waypoint without a passing mark.

The recorded-to-compiled correspondence in Figure 7 links local passage geometry and gait events to stored crossing and interval fields. Crossing position and direction are trail-derived with map-assisted placement; approach extents, tolerances, and speed caps are configured. The following diagnostic tests how consuming these fields affects approach behavior.

Runtime use of compiled conditions. We execute the field navigation core on this reference in a deterministic kinematic diagnostic. Ledge handling is crossed with lateral offsets $\{-0.4, -0.3, -0.1, 0, 0.1, 0.3, 0.4\}$ m and yaw offsets $\{-30, -22.5, -7.5, 0, 7.5, 22.5, 30\}^\circ$. The resulting 98 rollouts visit three stored ledges each, yielding 294 crossing records. Reference geometry, gait plan, flat input to the core, and command/feedback models remain shared. Enabling handling activates event-associated approach tracking, speed limits, retries, and a route-prior window, testing their coupled runtime effect.

Mean absolute lateral error falls from 13.43 cm to 0.56 cm, and mean absolute heading error from 22.30° to 1.55° when handling is enabled (Figure 5). All prescribed offsets are injected and all crossings are recorded. Both configured geometric limits are met in 100% of enabled ledge records

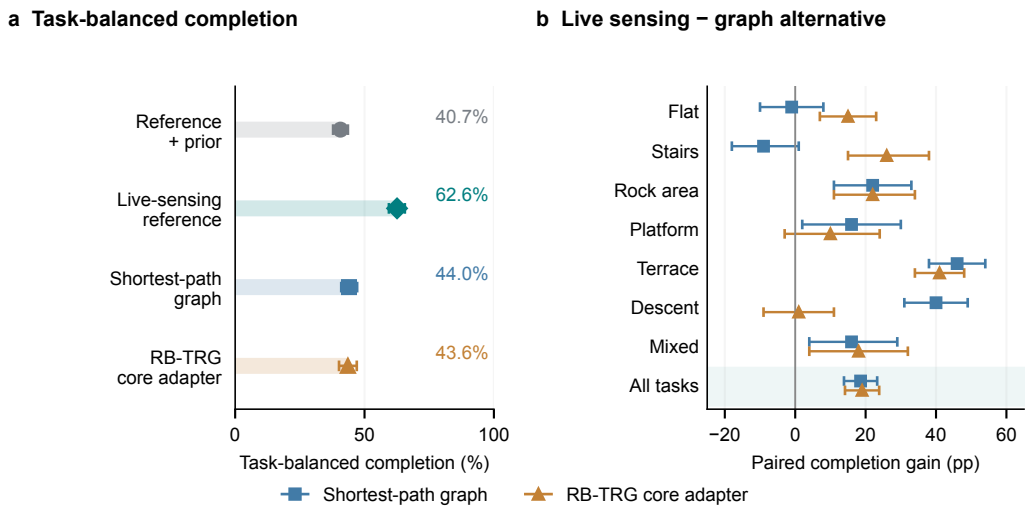

Figure 6: **Live sensing improves aggregate completion on the fixed task suite.** (a) Task-balanced completion over 700 trials per configuration; tasks carry weight $1/7$ and the two conditions weight $1/2$ each. (b) Paired differences against both graph alternatives, by task and pooled. The 95% intervals use 20,000 percentile seed-block bootstrap draws, preserving both conditions and all task strata within each seed. They describe seed variability on one reconstructed course. Appendix L gives the task protocol, corrected rock-area start, and complete completion/fall counts.

and 22.4% of disabled records (147 records per setting). At these marked events, the stored heading also agrees with the local centerline tangent: the contribution of event information is to associate that geometry with the tracking and execution rules for each passage. The experiment measures approach geometry under a kinematic feedback model. Appendix E gives the crossing association, initial diagnostic grid, and model assumptions.

5.2 Route preparation and observation selection

We compare four configurations with a shared executor. The reference configuration retains the prepared centerline and selected terrain-observation prior; its live-sensing variant removes that override. Two variants replace the centerline using a shared layered graph inside the same frozen corridor: shortest-path planning and a body-aware risk graph adapted from the public RB-TRG planning core (Guo et al., 2026). Both retain live sensing, speed limits, gait intervals, and entry settings. The experimental factors are centerline preparation and terrain-observation selection, with the other passage fields shared across variants.

Fifty paired seeds under both conditions on seven fixed subroutes produce 2,800 valid trials spanning flat terrain, stairs, a rock-area approach, platforms, terraces, descent, and a mixed sequence. With equal task and condition weights, the live-sensing reference achieves 62.6% completion, versus 40.7% with the terrain prior, 44.0% with shortest-path planning, and 43.6% with the RB-TRG adapter (Figure 6). Its paired gains over the graph alternatives are 18.6 percentage points (95% interval 13.9–23.3) and 19.0 points (14.1–23.9), respectively. Recorded falls are 24.6% for live sensing and 46.3% for the RB-TRG adapter. Task structure matters: shortest-path planning reaches 76% on stairs versus 67% for live sensing, while the terrain prior reaches 58% in the rock area versus 30% for live sensing.

A separate 640-trial geometry stress test rescales scene height by $\{0.8, 1.0, 1.1, 1.2\}$, including a same-host identity control, on four tasks with ten pose-noise seeds. Live sensing and the RB-TRG adapter each complete 27.5%, versus 38.8% for shortest-path planning and 25.6% for the prior-enabled reference (Appendix M). The fixed-course advantage therefore does not transfer to this cohort. Earlier 324-trial development batches show cross-environment sensitivity, and a 48-cell entry-factor diagnostic shows an interaction between observation selection and gait-request timing

(Appendices A.1 and A.2). These comparisons identify conditions under which stored context and current sensing help fixed actors negotiate transitions.

5.3 Rejoining a persistent reference

We keep the frozen centerline and live sensing fixed while crossing rejoining on/off with entry conditioning on/off. The latter removes only the 0.6 m gait-entry lead and entry deceleration; speed caps, gait labels, actors, and handoff logic remain shared. Across eight segments, three fresh seeds, both factors, and matched position-displaced/undisplaced conditions, we test 192 configurations. A $1.2\,\mathrm{m}$ lateral displacement is attempted at the smaller of 8 m progress and 40% of segment length, subject to ground validity.

The paired analysis requires every factor combination in a route–seed set to receive the displacement. Six segments contribute 32 rejoining-off/on pairs. Rejoining reduces mean terminal distance to the frozen centerline from 1.14 m to 0.11 m and raises ordered completion from 0% to 59.4%. Every rejoining-enabled run returns inside the original corridor, although some subsequently fail to complete. Within these runs, entry conditioning changes completion from 56.3% to 62.5% (16 executions per setting). Appendix I retains all 192 outcomes, trigger availability, paired distances, and an earlier nominal replay showing recovery before the next gait request. Together they separate recovery of route progress from coordination of the next transition; entry braking was inactive in the lateral-force diagnostic (Appendix C).

5.4 Physical deployment across terrain transitions

The system placed third in the All-terrain Patrol task of the Embodied Future track at the 2026 Global Open-source AI Challenge (GOAI). Among that task's finalists, it was the only entry to navigate the designated terrain sequence without manual intervention. The competition received 2,999 valid preliminary submissions across all tracks (GOAI Organizing Committee, 2026). The physical deployment uses LiDAR localization, perception, and factory gait controllers (Figure 4; Appendix F). The navigation stack and control-model inference also ran on an Apple M1 laptop. This deployment integrates the navigation stack with existing locomotion controllers; the simulation experiments use separate ONNX actors.

6 Discussion

The passage-conditioned reference makes information implicit in a taught trajectory available to local adaptation without retraining locomotion policies. Teaching compilation supplies approach and gait fields; the kinematic diagnostic links their use to ledge-entry geometry. On a separate mapping-derived reference, contact simulation shows recovery through rejoining.

The kinematic diagnostic evaluates coupled handling functions; contact simulations evaluate flat and stair actors on a separately prepared reference. A controlled physical comparison against the same demonstrated geometry remains an open experiment for assessing the complete passage representation. Rejoining drives the displacement recovery effect; entry conditioning changes completion from 56.3% to 62.5% within matched rejoining-enabled cases. Removing entry braking improves the earlier batch. Live sensing leads the fixed 50-seed comparison, but geometry changes and earlier execution-environment batches reverse the live-sensing completion advantage.

REPRODUCIBILITY AND AI USE

The supplementary release consists of navigation and evaluation source code, count tables, development and recovery run summaries, and the kinematic protocol and outcomes, including failed outcomes and the setup-invalid ledger.

Language models assisted manuscript organization and editing, literature and code cross-checks, experimental diagnosis, and implementation of evaluation adapters, instrumentation, and plotting scripts. We take responsibility for the research claims and the resulting text, code, and analyses. Conceptual method and terrain illustrations used an image-generation tool; empirical frames retain their original image content, with geometric annotations added as overlays. Simulation figures render recorded states and measurements.

REFERENCES

Dominic Baril, Simon-Pierre Deschênes, Olivier Gamache, Maxime Vaidis, Damien LaRocque, Johann Laconte, Vladimír Kubelka, Philippe Giguère, and François Pomerleau. Kilometer-scale autonomous navigation in subarctic forests: Challenges and lessons learned. *Field Robotics*, 2: 1628–1660, 2022. doi: 10.55417/fr.2022050. URL https://doi.org/10.55417/fr.2022050. Official WILN repository: https://github.com/norlab-ulaval/wiln.

Marko Bjelonic, Ruben Grandia, Oliver Harley, Cla Galliard, Samuel Zimmermann, and Marco Hutter. Whole-body mpc and online gait sequence generation for wheeled-legged robots. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8388–8395. IEEE, September 2021. doi: 10.1109/iros51168.2021.9636371. URL http://dx.doi.org/10.1109/IROS51168.2021.9636371.

Marko Bjelonic, Ruben Grandia, Moritz Geilinger, Oliver Harley, Vivian S Medeiros, Vuk Pajovic, Edo Jelavic, Stelian Coros, and Marco Hutter. Offline motion libraries and online mpc for advanced mobility skills. *The International Journal of Robotics Research*, 41(9-10):903–924, June 2022. ISSN 1741-3176. doi: 10.1177/02783649221102473. URL http://dx.doi.org/10.1177/02783649221102473.

Fei Gao, Luqi Wang, Boyu Zhou, Luxin Han, Jie Pan, and Shaojie Shen. Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments, 2019. URL https://arxiv.org/abs/1907.00520.

GOAI Organizing Committee. Global open-source ai challenge: 70 finalist projects announced, September 2026. URL https://www.goaihz.com/news/%E6%80%BB%E5%86%B3%E8%B5%9B70%E5%BC%BA%E5%90%8D%E5%8D%95. Official announcement, September 10, 2026.

Zhiqiao Guo, Bichi Zhang, and Sören Schwertfeger. Robot-body-aware traversal risk graph planning for wheeled-legged robots in complex terrain, 2026. URL https://arxiv.org/abs/2608.16433.

Mateo Haro, Julia Richter, Fan Yang, Cesar Cadena, and Marco Hutter. Path-conditioned reinforcement learning-based local planning for long-range navigation, 2026. URL https://arxiv.org/abs/2603.13888.

Joonho Lee, Marko Bjelonic, Alexander Reske, Lorenz Wellhausen, Takahiro Miki, and Marco Hutter. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Science Robotics*, 9(89):eadi9641, 2024. doi: 10.1126/scirobotics.adi9641. URL https://arxiv.org/abs/2405.01792.

Ugo Rosolia and Francesco Borrelli. Learning how to autonomously race a car: A predictive control approach. *IEEE Transactions on Control Systems Technology*, 28(6):2713–2719, November 2020. ISSN 2374-0159. doi: 10.1109/tcst.2019.2948135. URL http://dx.doi.org/10.1109/TCST.2019.2948135.

Jordy Sehn, Yuchen Wu, and Timothy D. Barfoot. Along similar lines: Local obstacle avoidance for long-term autonomous path following, 2022. URL https://arxiv.org/abs/2211.02047.

Chengzhen Yan, Yiming Jiang, Qingchen Liu, and Jiahu Qin. Learning policy-dependent traversability for terrain-aware quadruped navigation. *IEEE Transactions on Automation Science and Engineering*, 23:10325–10336, 2026. ISSN 1558-3783. doi: 10.1109/tase.2026.3696520. URL http://dx.doi.org/10.1109/TASE.2026.3696520.

A Component removals and execution sensitivity

We connected the navigation core to existing MuJoCo scenes and frozen `J3100.onnx` and `stairs_1150.onnx` actors. The simulation adapter is separate from the factory locomotion stack used in the physical deployment. We test three route segments under nominal conditions, pose perturbation, and an inserted obstacle, using seeds 11, 23, and 37. Every case–condition–seed cell contains all five configurations, for 135 attempts. These are development experiments on the existing reconstructed course.

The configurations remove local detours, entry braking, route readjustment, or the adapter's terrain-observation override. Disabling detours removes lateral detour permission and the A* fallback while retaining live observations and readjustment. Disabling readjustment changes the route-rejoining procedure. Disabling the terrain override retains the live-input path in this adapter. Each variant probes its corresponding execution interface, with the remaining reference fields held fixed.

The complete trial-level development outcomes accompany the source data in `component_trial_records.csv`.

Removed component	Completion	Falls	Obstacle contact	Non-wheel contact
local detour	-11.1	+0.0	+7.4	+7.4
entry braking	+7.4	-11.1	+0.0	-3.7
readjustment	-3.7	+3.7	-14.8	-11.1
terrain override	+18.5	-11.1	+0.0	-14.8

Table 1: **Component-removal effects in the development batch.** Entries are percentage-point changes from the current core on matched cells, with equal weights for three routes and three conditions. Positive completion change indicates more completed tasks; positive event change indicates more recorded events. The batch contains 135 attempts. Contacts and falls are non-exclusive incidences sampled at 50 Hz; full counts are retained in `component_outcome_counts.csv`.

Local detours increase aggregate completion by 11.1 percentage points relative to disabling them. Yet removing entry braking or the terrain override can improve completion. These effects depend on the interaction between execution rules and the fixed actors. The paired effect table retains all tested removals, including adverse effects. Removing readjustment reduces obstacle-contact incidence by 14.8 points while reducing completion by 3.7 points. Across common successes, paired completion-time differences have median zero. Falls, blocked states, and timeouts remain in the full source counts.

Outcome accounting. Ordered completion uses the frozen goal coordinates and ordering. We report contact, fall, and termination events for every attempt. Completion times are compared only within matched cells where both configurations complete, with that denominator reported. Controller, scene, task, and perturbation settings remain fixed within a comparison. The component-removal batch is retained separately from the structural comparisons and the displacement diagnostic. Contact and termination statistics are reported for each batch separately.

A.1 Execution-environment and seed comparisons

The structural study contains 108 macOS attempts, 108 exact-source Linux recheck attempts, and 108 additional Linux attempts with new seeds. The exact-source recheck uses the same pinned source and asset files as the macOS evaluation. macOS arm64 uses Python 3.12.14; Linux x86_64 uses Python 3.10.16. MuJoCo 3.3.1, ONNX Runtime 1.19.2, NumPy 1.26.4, and SciPy 1.13.1 match. Packaged sources are controlled; operating system, processor architecture, interpreter, and

Batch	Prior reference Completion (%)	Change from prior reference (pp) Live sensing	Shortest path	Body-aware
macOS	70.4	-7.4	-14.8	-18.5
Linux recheck	51.9	+18.5	+11.1	+14.8
Linux new seeds	48.1	+11.1	+14.8	+18.5

Table 2: **Batch-level performance and paired configuration contrasts.** The three batches contain 324 attempts in total. Within each batch, three route segments and three conditions receive equal weight, with three seeds per cell and four configurations. Contrasts subtract the prior-enabled reference within the same batch. Batches remain separate because execution environments and seed sets differ.

numerical effects vary together. The additional-seed batch uses the earlier baseline runner on the same reconstructed course and is reported separately.

Condition-level counts are retained in the accompanying source-data file `development_condition_counts.csv`. The graph adapter completes more inserted-obstacle cases than the live-sensing reference in each batch; the batch-level contrasts in Table 2 summarize the reversal across execution environments.

A.2 Entry timing and observation selection

On segment WP05–WP08, we cross terrain-prior selection, gait-entry lead, and entry deceleration under nominal execution and pose noise, using seeds 11, 23, and 37. The complete design contains 48 attempts: 30 additional runs and 18 matching records from the component study. Scene, task, route, packaged sources, and effective navigation settings match; the latter differ only in the three declared factors. A 0.6 m lead advances the gait request, while deceleration is either disabled or set to 0.8 in the existing entry rule. All other execution settings remain shared.

Observation	Mean completion effect of a 0.6 m entry lead (pp)
Live sensing	+25.0
Route prior	-50.0

Table 3: **Entry timing interacts with observation selection.** Each effect subtracts the no-lead setting on matched cells. Nominal/pose-noise conditions and deceleration off/on receive equal weights. The complete design contains 48 executions, including 18 reused records. Effects summarize this fixed-segment diagnostic. All sixteen condition-level count cells remain in `entry_factor_counts.csv`.

B Implementation and controller interfaces

The field implementation compiles pose trails, waypoint marks, and gait events into a reloadable route/terrain bundle. Corridor widths and speed caps are configured; crossing geometry and gait intervals combine recorded events with geometric rules. Route progress restricts projection on nearby return legs, while a bounded observation deque, retained lateral-transition anchor, and gait-boundary hysteresis preserve continuity within a run through explicitly maintained state.

Field execution supports flat, stairs, and platform gait requests. The simulation comparison covers flat and stairs actors. The frozen simulation exports are `J3100.onnx` and `stairs_1150.onnx`; actor hashes and environment versions accompany the evaluation manifests. The archived controller-training configuration is documented separately in Appendix K.

The graph baselines use the same frozen simulation corridor sampled at 0.5 m intervals, with five lateral offsets at fractions $-1, -0.5, 0, 0.5, 1$ of its half-width. Adjacent layers are connected only where sampled map cells are known and free of static obstacles. The shortest-path variant minimizes edge length. The RB-TRG adapter applies its body-aware planner to this graph and the common elevation grid, using a 1.0×0.6 m body footprint. Only centerline coordinates are replaced; segment

Information	Origin	Runtime use
Centerline and goals	Selected pose trails, passing marks, map-checked geometric processing	Route projection and candidate coordinates
Corridor and detour permission	Configured widths and permissions, with clearance checks	Bounds on local alternatives
Ledge event and approach pose	Platform-gait trail events, local displacement, optional map adjustment	Event-specific approach tracking and checks
Gait intervals	Recorded gait votes, map steepness, configured platform extents	Requested locomotion mode and boundary timing
Speed caps and approach extents	Configured route fields and runtime settings	Command limits and activation windows
Retry and handoff rules	Shared execution configuration	Re-approach, hold, and switch coordination

Table 4: **Information supplied to the execution interface.** Demonstration-derived fields and configured rules are distinct. A route bundle stores event-associated information; shared runtime settings determine how the executor consumes it.

endpoints and other execution fields are retained. The adapter integrates the repository's planning core into our shared executor.

For Figure 14, each recorded robot position is projected onto every nonzero line segment of that run's frozen original centerline. The minimum point-to-segment distance is plotted, keeping the measurement reference fixed through changes to the follower's current path coordinates during rejoining.

C Instrumented entry-conditioning diagnostic

To distinguish a configured rule from an active intervention, we instrument the entry-conditioning branch without changing commands. On the first route, seeds 11, 23, and 37 cross entry conditioning on/off with a fixed force on/off, for 12 cells. The reference, live-sensing permissions, actors, goals, and enabled rejoining procedure remain fixed. At route progress 8 m, the force is applied leftward for 0.3 s with magnitude $0.1mg = 18.63$ N, where the model's total mass is 18.99 kg. All six force-on cells record the force event; all 12 cells complete.

The logs retain the command before and after conditioning, distance to the next boundary, computed speed threshold, and a flag indicating whether braking replaces the command. With conditioning enabled, the threshold is computed on 719 ticks without the force and 725 ticks with it, summed over the three seeds in each group. No command exceeds its threshold, so the braking replacement fires in zero cells. The gait lead still changes the first transition time. This separates two effects of the combined setting: gait scheduling is active, whereas the deceleration rule is inactive in this tested route/actor/speed regime.

D Teaching traces and local compilation

E Kinematic diagnostic on the compiled field reference

Geometry and event information. The compiled centerline contains 2,711 XY samples and measures 270.923 m. Each of the three stored crossing positions lies within 0.4 mm of this polyline. Given a marked crossing location, the tangent from a symmetric ± 0.25 m arc window agrees with its stored heading within $0.01°$; a ± 1 m window differs by at most $1.48°$. These values quantify internal consistency between the saved curve and labelled events. Event records identify the ledge locations and approach windows, while configured rules specify checks, speed limits, and retries (Table 4).

Rollouts and measurements. The diagnostic loads the field route, terrain records, gait plan, and navigation core. The three stored ledges are visited in route order. The main range-extension study

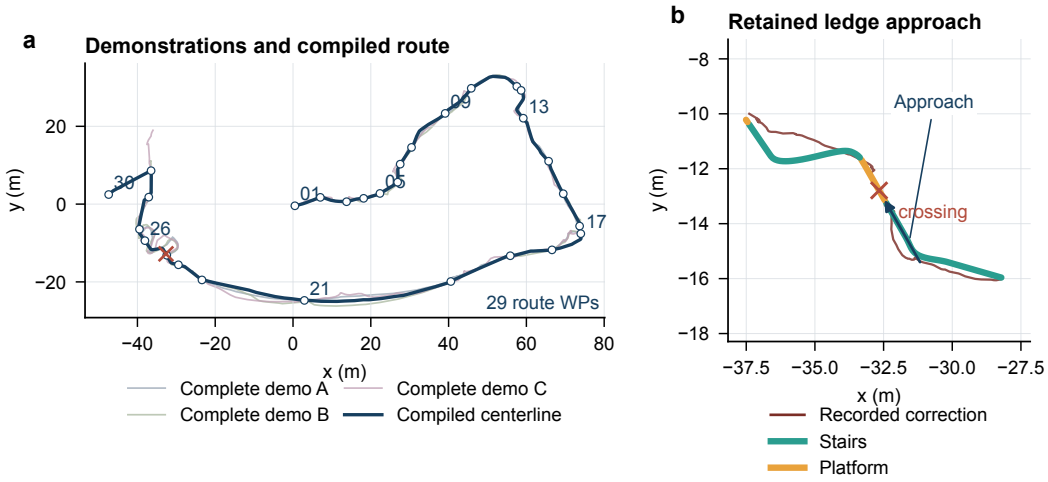

Figure 7: **From recorded passages to reusable conditions.** (a) Three complete teaching traces and the compiled route, with its 29 waypoint entries. (b) A recorded local correction and the compiled crossing, approach direction, and gait intervals near a ledge. The curve and fields show the compiler's stored output.

crosses ledge handling on/off with the seven lateral and seven heading offsets specified in the main text, producing 98 deterministic rollouts and 294 ledge records. An initial diagnostic used lateral offsets $0, \pm 0.20$ m and heading offsets $0, \pm 15°$, producing 18 rollouts and 54 records; geometry compliance was 37.0% with handling disabled and 100% with handling enabled (27 records per setting). The two grids are reported separately. Each offset is applied once per ledge when projected progress first lies 0.72–0.80 m before its stored arc coordinate. In the main grid, six records are zero-offset controls and 288 receive a nonzero offset. All records are retained, with three repeated ledge measurements grouped within each rollout.

Planar pose is integrated from body-frame velocity commands at 20 Hz, with a 0.30 s steady-state command delay after seven initial zero-command ticks. The core receives zero reported velocity in its pose input. Gait starts as flat, and a local feedback model changes the reported gait after a pending request has lasted more than 1.5 s. The common flat and stair speed overrides and walking speed are 1.0 m/s, while near-ledge limits remain active when ledge handling is enabled. The diagnostic uses kinematic integration and synthetic command/gait feedback to evaluate runtime decisions.

Both settings receive the same flat-observation input and common gait-zone prior configuration. The enabled setting also uses the after-ledge route-prior window. Thus, its factor combines stored crossing geometry, approach tracking, speed limits, retry behavior, and prior selection. These mechanisms vary together, and prior selection can change the effective observations received during execution.

A crossing is the first sample for the next ordered ledge with projected progress within 0.35 m of its stored arc coordinate, forward ledge-frame coordinate in $[0, 0.30)$ m, and absolute lateral coordinate below 1.5 m. This association excludes unrelated intersections of the route. We then test the configured 0.10 m lateral and $10°$ heading limits using the recorded pose. All 294 main-grid records have an associated crossing; all offsets were applied. The experiment ends on completion, a hold, or progress above 260 m. The diagnostic scores geometry at the three ledges before the 260 m cutoff. Per-tick poses, commands, gait requests, synthetic feedback, and crossing events were logged.

F PHYSICAL TERRAIN COVERAGE AND APPROACH SEQUENCE

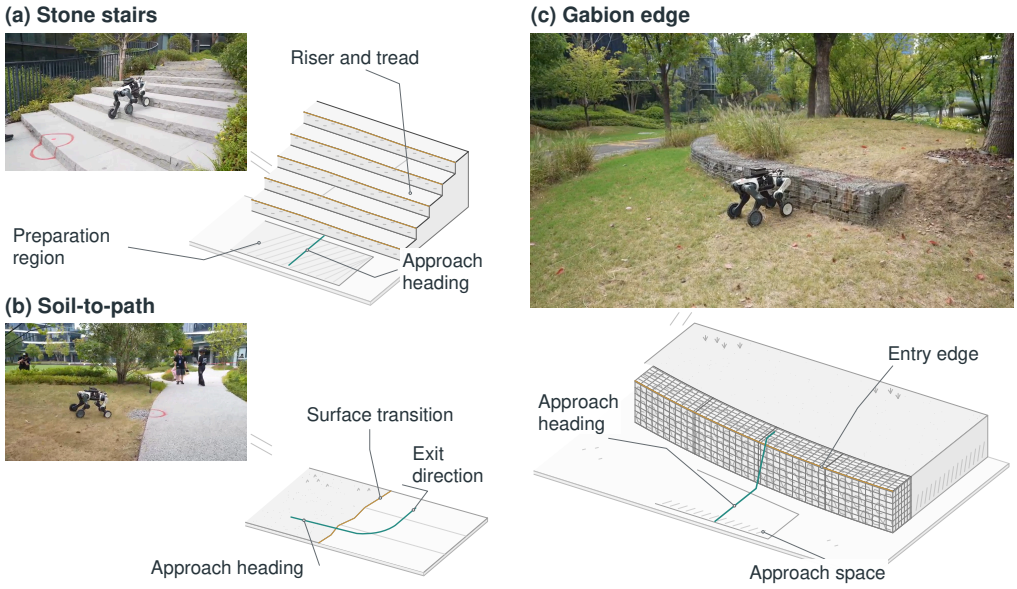

Figure 8: **From field geometry to passage conditions.** (a) Stone stairs, (b) a soil-to-path transition, and (c) a gabion ledge are paired with spatial abstractions of the relevant passage geometry. Amber marks visible terrain boundaries; teal arrows illustrate approach directions on conceptual, non-metric diagrams.

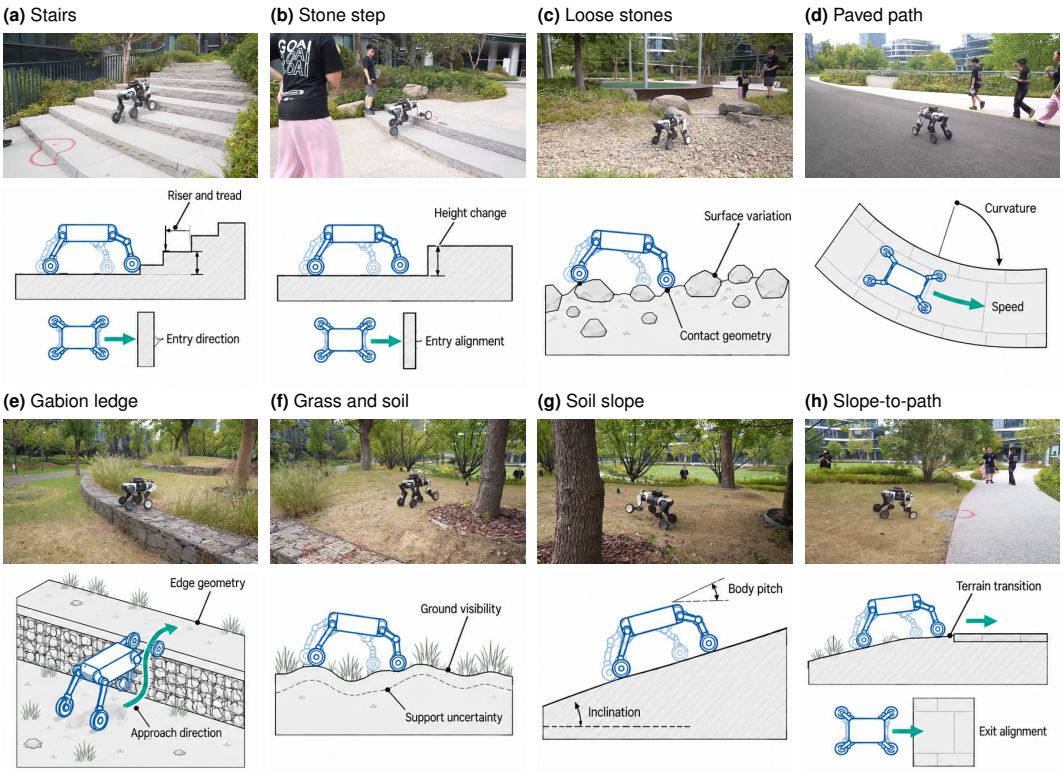

Figure 9: **Terrain transitions and their passage conditions.** Each pair shows a physical scene above and its geometric interpretation below: height and entry direction, local surface variation, route curvature, and slope transitions. The schematics use illustrative proportions.

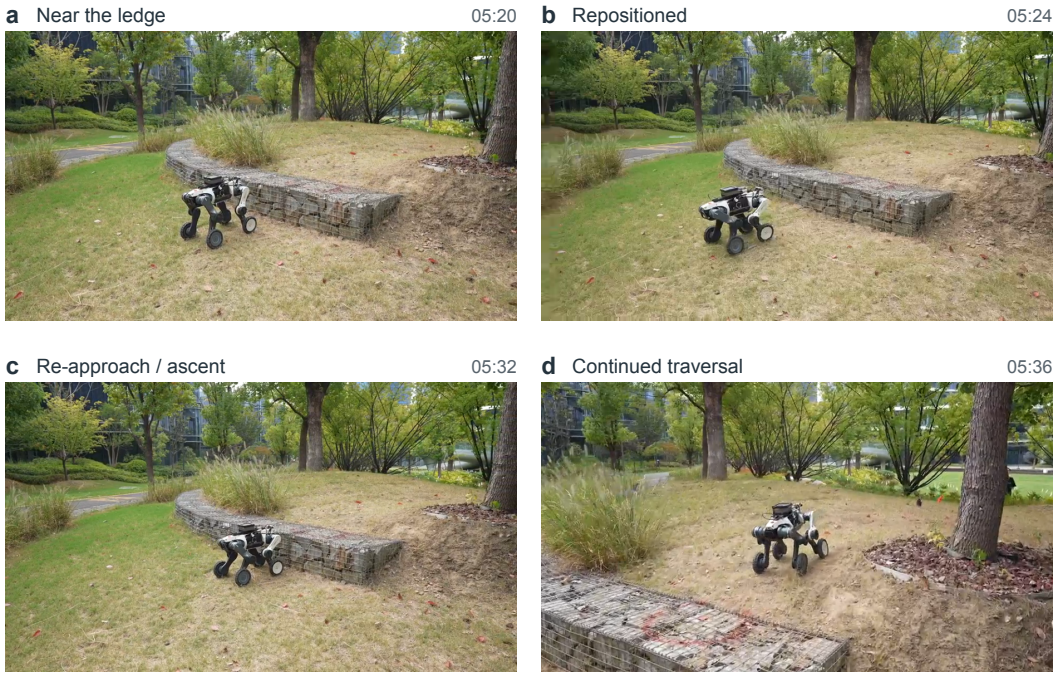

Figure 10: **Ledge approach and repositioning.** Physical frames show approach, repositioning, re-approach/ascent, and continued traversal.

G MAP-GROUND SECTIONS AND SPATIAL METHOD ILLUSTRATION

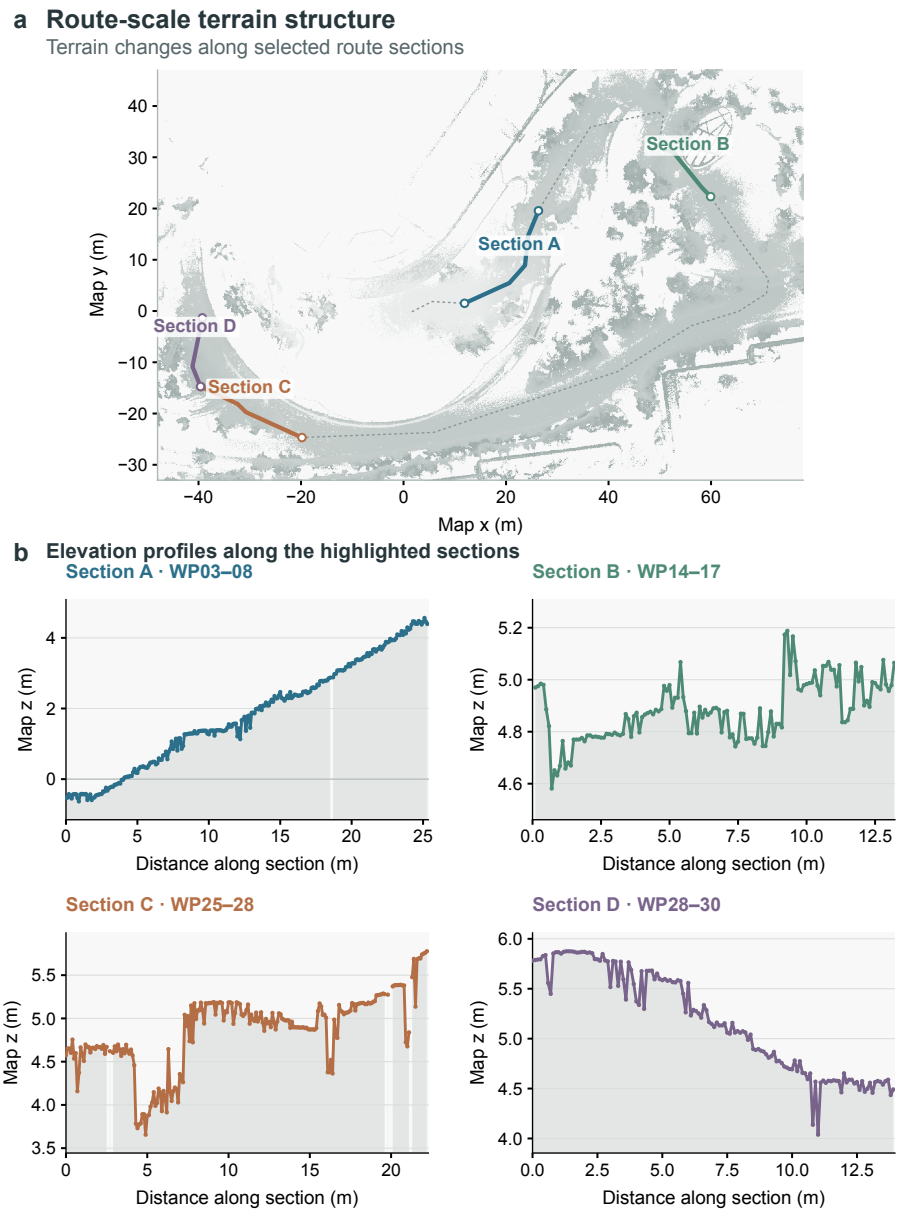

Figure 11: **Route-scale terrain structure and local elevation profiles.** A–D highlight four reference-route sections on a gray map, with matching colours in the profiles. The 0.1 m import surface is derived from the archived point cloud with a per-cell minimum and at least two returns. Sections sample the frozen reference XY every 0.1 m, using only known cells. The sections contain 254, 133, 223, and 140 stations, respectively; missing map samples remain gaps. Panel scales differ. Profiles combine the simulation reference's XY coordinates with elevations from the candidate ground map. Sparse scans, vegetation, and multiple surfaces can affect cell minima. The physical video is unregistered to these section coordinates.

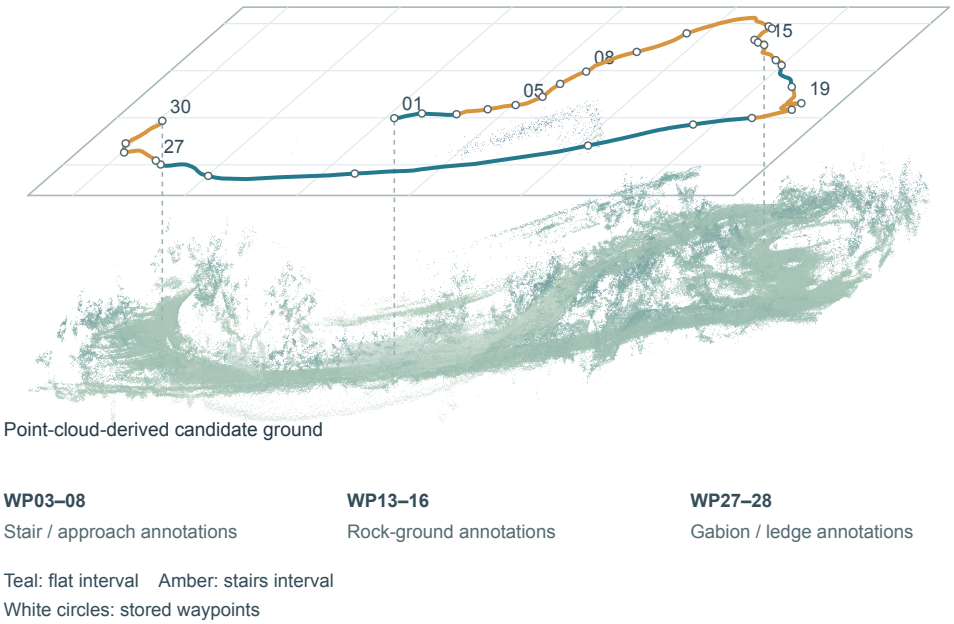

Figure 12: **Source-derived geometry and stored task information.** All 392,452 known 0.1 m cells in the stated display region are rendered from the candidate ground map. The upper plane contains archived simulation-reference XY, waypoints, and flat/stairs segment labels. Terrain names come from stored annotations. The conceptual axonometric view combines these XY task fields with map elevations and exaggerates height for readability.

H GOAL AND ROUTE-BUILD INTERFACES

For the field task, a waypoint is reached when some part of the robot intersects a disk of radius $0.25\,\mathrm{m}$. Let $\mathcal{B}(q)$ be projected body geometry and $W_i = \{p : \|p - w_i\| \le 0.25\,\mathrm{m}\}$. Ordered completion requires

$$\exists\, t_1 < \cdots < t_N : \quad \mathcal{B}(q(t_i)) \cap W_i \neq \varnothing \quad \forall i. \tag{7}$$

The simulation study also measures the stricter centre-distance criterion $0.20\,\mathrm{m}$:

$$C_{0.20} = \mathbf{1}[\exists\, t_1 < \cdots < t_N : \ \|p_{\mathrm{centre}}(t_i) - w_i\|_2 \le 0.20\,\mathrm{m}\ \forall i]. \tag{8}$$

The field compiler separately targets a line within $0.10\,\mathrm{m}$ of each mark, with an additional body-sweep proximity check. This geometric compilation tolerance, the runtime gate radius, and the sampled simulation score serve different purposes. Each evaluation freezes its goal coordinates and ordering before running the methods.

A re-marked waypoint can be incorporated by rebuilding the bundle in the same registered map frame. The compiler selects the latest passing mark, regenerates route and maneuver fields, and reports missing usable marks; the verified field bundle omits one such waypoint. This is an explicit route-build operation, and the new bundle requires clearance checks and a reload before execution.

The field configuration requires a stillness threshold of 0.12 in the monitored translational and yaw velocity channels, sustained for $0.3\,\mathrm{s}$, before switching. Its simulation adapter records the acknowledgement model separately. Comparisons retain the same handoff procedure. The gait-interval state

also resists boundary jitter: a small backward change in projected route progress retains the current interval, with a 2 m retreat allowance in the field implementation.

Recorded crossing correspondence. Figure 7 shows both the route-scale compilation and a retained local condition. Near the illustrated crossing, a recorded pose lies 0.022 m from the compiled crossing, with heading 2.107 rad compared with the stored approach direction 2.089 rad. The bundle assigns successive stairs, platform, and stairs intervals to that region. Crossing position and direction are derived from local trails with map-assisted placement; approach/exit extents are configured. The stored rise field records the change in body height across the passage. This example links recorded passage geometry and gait events to fields consumed by the executor; the following diagnostic tests their effect on approach behavior.

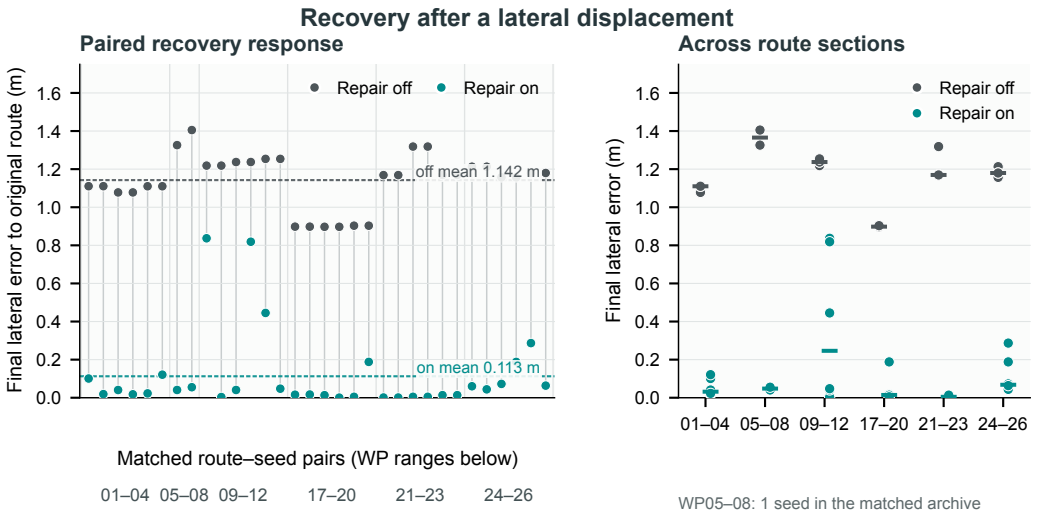

Figure 13: **Persistent-reference recovery across six route segments.** Left: final distance to the original reference in 32 matched rejoining-off/on pairs; lines connect the same route, seed and entry setting. Right: the same observations by segment, with median bars. WP05–08 has one qualifying seed; other segments have three. The mean distance falls from 1.142 m to 0.113 m. The paired analysis requires all four factor combinations to receive the displacement.

I Recovery across route segments

The expanded contact-simulation experiment crosses eight fixed segments (WP01–04, 05–08, 09–12, 13–16, 17–20, 21–23, 24–26, and 27–30), three seeds (101, 211, and 307), rejoining off/on, entry conditioning off/on, and position displacement absent/present. All runs use the live-sensing configuration and the pose-noise model specified in the main protocol. The simulation limit is 180 s. Among the 96 intended displacements, 69 are applied, 11 fail the ground-validity test, and 16 are not reached. The remaining 96 configurations are undisplaced controls. These groups remain in the complete outcome table below.

Rejoining	Entry conditioning	Control	Intended displacement
Off	Off	20.8	0.0
Off	On	29.2	0.0
On	Off	25.0	37.5
On	On	45.8	50.0

Table 5: Ordered completion (%) for the full expanded factorial. Each entry contains 24 attempts: eight route segments and three seeds. Intended-displacement columns use all 24 intended-displacement attempts as the denominator, including ground-invalid and trigger-not-reached cases.

For the paired perturbation analysis, a route–seed set qualifies only if all four rejoining-by-entry combinations receive the prescribed displacement. This yields 16 sets and 64 perturbed executions, corresponding to 32 repair-off/on pairs. The pairs cover six segments; WP05–08 contributes one qualifying seed and the other five contribute three. The two entry settings are paired measurements within each route–seed set; we report matched counts and distances. Repair-enabled executions return inside the original segment corridor in all 32 matched cases, compared with none when disabled. Ordered completion is 59.4% versus 0%, respectively. A return is counted only after raw XY is first observed outside the corresponding frozen segment's corridor and subsequently inside it. Terminal distance uses raw XY projected onto the original frozen centerline. Figure 13 displays every matched terminal error, including runs that later fail to complete.

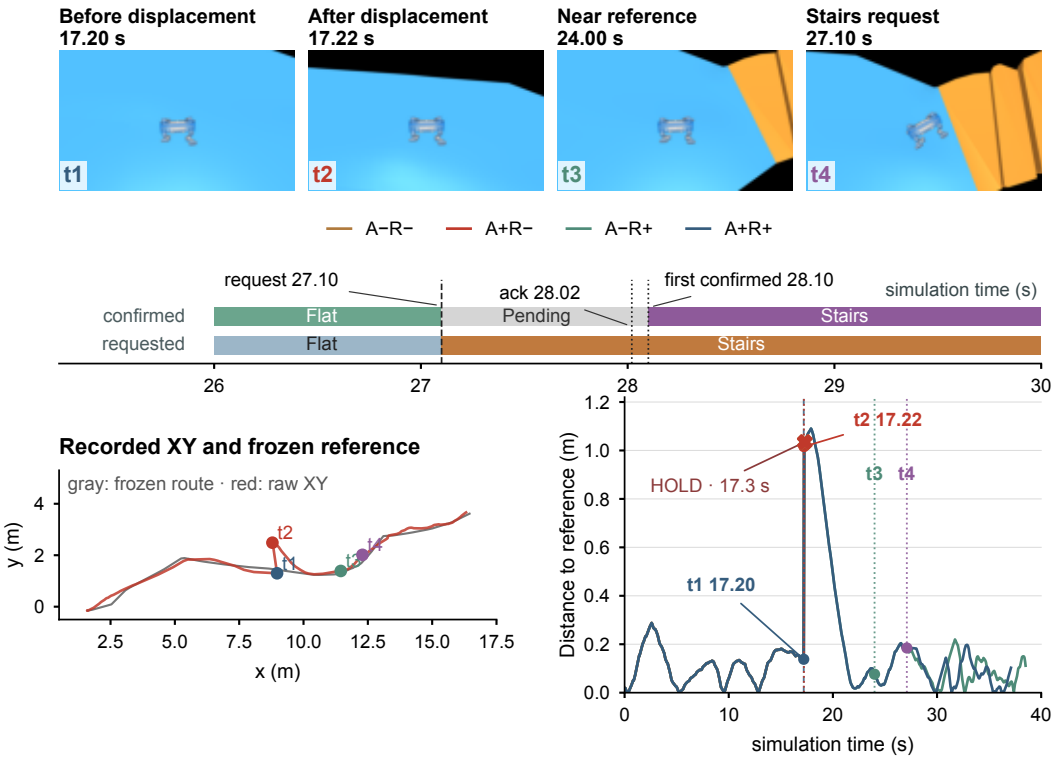

Figure 14: **A displaced robot recovers progress toward the next transition.** Four replay frames, XY trajectory, and event markers come from one earlier nominal rejoining-on, approach-on run. Distance curves compare all four matched settings (A: entry conditioning; R: rejoining) and use raw XY projected onto the frozen original centerline. Overlapping curves are plotted without offsets. The request and confirmed gait strips zoom into 26–30 s of the illustrated run; the last replay frame precedes the acknowledgement. Replays render stored simulator states at fixed simulation times, using a robot-centred camera with a shared view direction and distance.

Earlier nominal mechanism diagnostic. We prescribe a 1.2 m lateral position displacement at route progress 8 m, using the existing ground-validity check before applying it. Of the 36 planned cells (three routes, three seeds, four factor combinations), 24 receive the displacement on two routes; the remaining route fails the injection validity check in all 12 cells and is retained as an unperturbed control. Among the actually displaced cells, rejoining completes 100% and disabling it completes 0%. Both entry settings have the same completion outcome within each rejoining state. All 12 uninjected controls complete. The replay in Figure 14 comes from this nominal diagnostic. On one of the two displaced routes all settings remain in flat gait, so that segment identifies reference recovery with the flat gait retained throughout.

J FIXED LOCOMOTION ACTOR INTERFACE

Temporal inputs to the fixed locomotion actors. In the simulation adapter, the active flat or stair actor has the interface

$$a_t = \pi_{\theta_{g_t}}\big(z_t, a_{t-1}, \sin\phi_t, \cos\phi_t\big), \tag{9}$$

where g_t denotes the active actor, $z_t \in \mathbb{R}^{41}$ contains scaled body angular velocity, projected gravity, the velocity command, relative joint positions, and joint velocities, and $a_t \in \mathbb{R}^{16}$ is its raw action. The preceding action and gait phase are explicit temporal inputs. The wrapper resets the preceding action and phase when an actor takes over. It advances the flat actor's phase on a fixed clock and advances the stair actor's phase when the velocity command exceeds its activation threshold. Parameters θ_g remain fixed during the navigation comparisons. This actor interface is distinct from both the persistent route reference L and the runtime navigation state S_t; the physical deployment uses a separate factory gait stack.

Offline temporal-input diagnostic. We checked whether the two frozen simulation actors respond to their explicit temporal inputs. For each actor, we selected 16 equally spaced recorded states from the corresponding active-controller interval of one completed rollout. The preceding actions and phases are absent from the saved trace, so we constructed inputs with zero preceding action and four prescribed phases. Central differences perturbed each preceding-action coordinate by ± 0.05 and phase by ± 0.05 radians. Both actors showed nonzero action sensitivity to each input group. Re-querying the same base inputs after intervening calls produced identical outputs in this diagnostic. The measurements characterize conditional action sensitivity to the prescribed temporal inputs.

The 128 controlled observations are constructed from 32 recorded body states and four phases, yielding offline actor queries. In the flat and stair actors, respectively, the median scaled leg-target Jacobian spectral norms with respect to previous action are 0.2172 and 0.2154 rad, and the corresponding phase finite-difference vector norms are 0.1096 and 0.1538 rad/rad. Because actual previous actions and phases are absent from the logs, these constructed inputs can lie outside the closed-loop observation distribution.

K CONTROLLER-TRAINING CONFIGURATION

The project also contains an Isaac Lab / RSL-RL PPO training branch with an asymmetric actor–critic. Figure 15 documents this configuration. Its actor receives 59 values, including previous actions and two phase terms; its critic receives 249 values, adding linear velocity and a 187-value height scan. Both networks use hidden widths 512, 256, and 128 with ELU activations. The 16 action channels specify twelve leg-position targets and four wheel-speed targets. This configuration belongs to a separate archived training branch. Its checkpoint relationship to the evaluated ONNX actors and factory gait stack is undocumented.

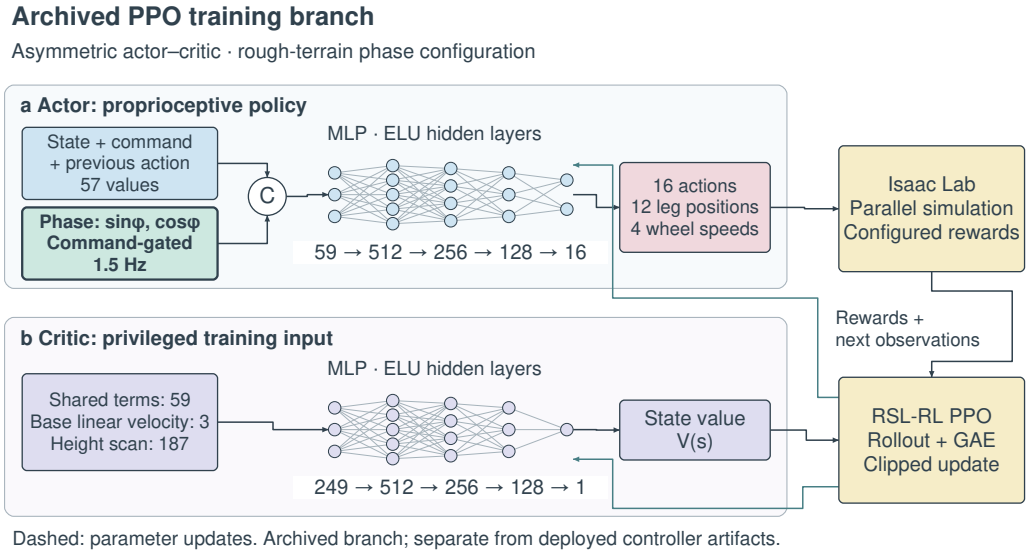

Figure 15: **Asymmetric actor–critic in the archived controller-training branch.** The critic receives privileged training information; the actor uses its configured proprioceptive, command, previous-action, and phase inputs. Hidden widths are given by the numerical labels; drawn neuron counts are schematic. Solid arrows show data flow and dashed arrows denote parameter updates. The actor and critic use corresponding observation terms with different scaling/noise settings. The diagram describes the archived controller-training infrastructure.

L EXPANDED MULTI-TERRAIN ROUTE COMPARISON

The final cohort combines 20 initial and 30 additional seeds. For configuration m, the reported task-balanced completion is

$$\widehat{R}_m = \sum_{t=1}^{7} \frac{1}{7} \sum_{c=1}^{2} \frac{1}{2} \left(\frac{1}{50} \sum_{s=1}^{50} y_{m,t,c,s} \right), \tag{10}$$

where y indicates ordered completion. The weights are equal across tasks and conditions. Because every stratum has the same number of trials, this equals the pooled completion rate. Contrasts are $100(\widehat{R}_{\text{live}} - \widehat{R}_m)$ in percentage points. Each seed is paired across seven fixed subroutes, nominal/pose-noise conditions, and all four configurations. These subroutes are fixed, potentially overlapping portions of one reconstructed course. Execution and ordered-goal criteria are the same as in the main protocol. Confidence intervals resample complete seed blocks with replacement 20,000 times and use the 2.5th and 97.5th percentiles. Paired differences are computed within task, condition, and seed before block resampling. Intervals quantify seed variability conditional on these tasks and settings.

Rock-area initialization. The initial WP11 start placed the robot in an inverted configuration. All 160 attempts at that setup remain in a separate setup-invalid ledger. The corrected task starts earlier at WP10 and includes the approach to the rock region. Its 160 initial runs and 240 additional runs are included in the final 2,800 valid trials. All remaining failure outcomes are retained. The trajectory scoring and file-integrity audit were completed before the final plots were assembled.

	Reference + prior		Live sensing		Shortest path		RB-TRG core	
Task	C	F	C	F	C	F	C	F
Flat	87	0	87	0	88	0	72	0
Stairs	18	72	67	16	76	10	41	56
Rock approach	58	36	30	63	8	91	8	89
Platform	45	42	58	23	42	46	48	44
Terrace	5	91	58	34	12	84	17	80
Descent	22	67	74	13	34	53	73	11
Mixed	50	42	64	23	48	46	46	44
Total	285	350	438	172	308	330	305	324

Table 6: Complete raw counts for the final route comparison. C: ordered completion; F: recorded fall. Each task has 100 trials per configuration (50 seeds and two conditions); each configuration has 700 trials in total. Falls and goal completion are separately scored outcome fields.

M VERTICAL GEOMETRY STRESS TEST

A separate host runs ten paired pose-noise seeds on four tasks at each vertical scale, including an identity-scale control. Scene heights change while horizontal geometry and robot parameters remain fixed. The descent task includes an approach waypoint and differs from the earlier multi-terrain descent subroute. All 640 runs are retained. Aggregate completion is 25.6% for the prior-enabled reference, 27.5% for live sensing, 38.8% for shortest-path planning, and 27.5% for the RB-TRG adapter. The stress test measures the combined response of route preparation, entry conditioning, and the frozen locomotion actors to the geometry change. The cohort is reported independently of the 2,800-run comparison.

Scale	Task	Prior	Live	Shortest	RB-TRG
0.8	wp05_08_stairs	2	5	7	1
0.8	wp17_21_platform	7	2	4	2
0.8	wp26_29_terrace	1	1	5	6
0.8	wp28_30_descent	3	7	6	9
1.0	wp05_08_stairs	3	5	5	1
1.0	wp17_21_platform	5	2	4	5
1.0	wp26_29_terrace	1	4	6	5
1.0	wp28_30_descent	4	3	4	4
1.1	wp05_08_stairs	3	3	6	1
1.1	wp17_21_platform	2	2	1	1
1.1	wp26_29_terrace	0	1	0	0
1.1	wp28_30_descent	1	3	3	4
1.2	wp05_08_stairs	3	1	4	0
1.2	wp17_21_platform	2	4	4	1
1.2	wp26_29_terrace	0	0	1	0
1.2	wp28_30_descent	4	1	2	4

Table 7: Complete ordered-completion counts for the geometry cohort. Each cell contains ten attempts, all under pose noise. Tasks and paired starts are fixed within each scale. The 1.0 setting is the same-host identity control.

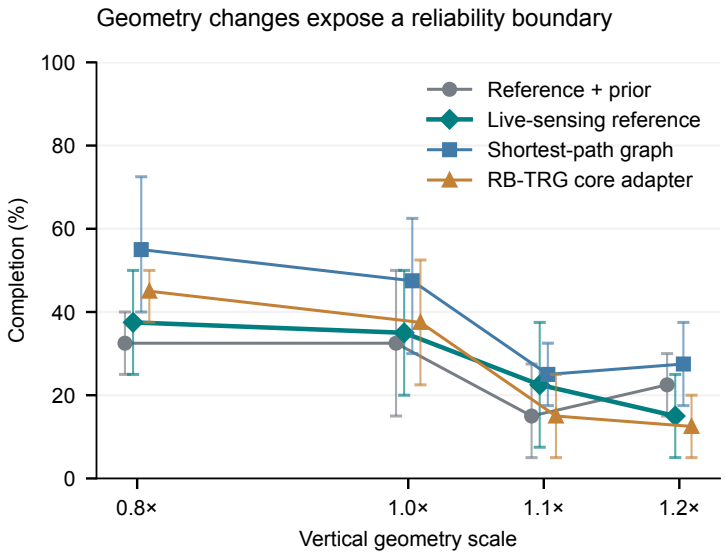

Figure 16: **Vertical geometry changes limit transfer of the fixed-course advantage.** Each point aggregates four tasks and ten paired pose-noise seeds (40 trials per method and scale). Error bars are 95% percentile seed-block bootstrap intervals from 20,000 draws, preserving all four tasks within each seed. Horizontal offsets separate method markers visually; all four methods use exactly the same indicated scale. Lines visually connect the four tested settings. Vertical scaling changes one scene; robot parameters remain fixed. The descent task includes an approach waypoint and differs from the earlier multi-terrain descent subroute.

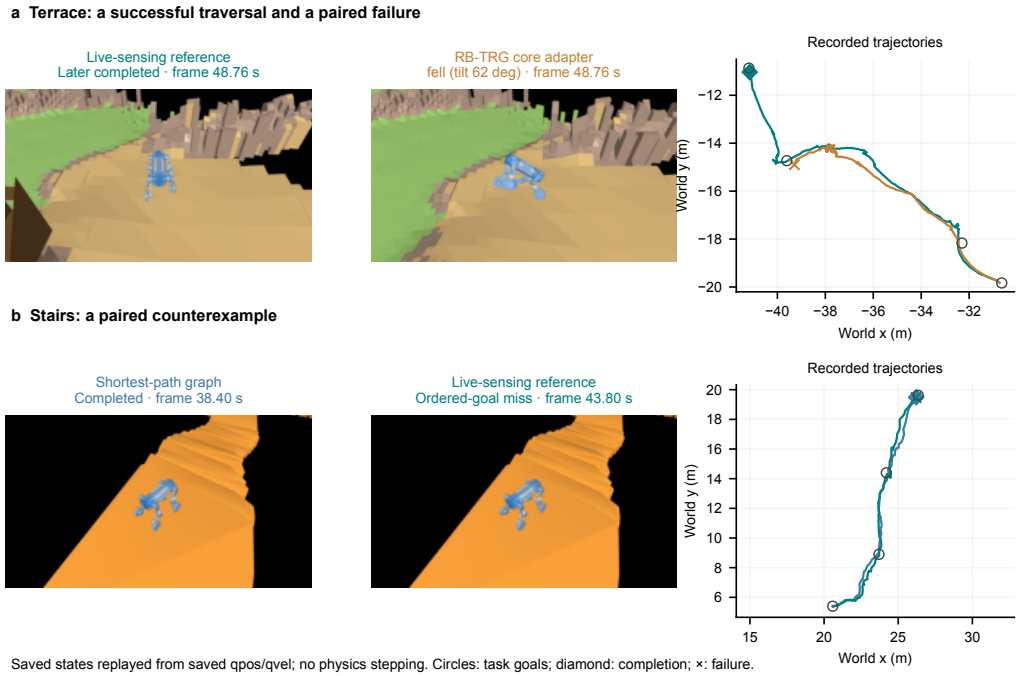

Figure 17: **Recorded simulations illustrate both an advantage and a counterexample.** Saved-state frames and full XY traces use saved states from matched pose-noise trials: terrace seed 3007 and stairs seed 3012. For each task, we select the lower-median successful completion time among discordant pairs in the additional 30-seed cohort (two terrace candidates and six stair candidates). These selected cases illustrate discordant outcomes. On stairs, the live-sensing run ends with software completion but misses the ordered-goal criterion; it is scored as a failure. Terrace frames use the matched failure time of 48.76 s; the live-sensing run subsequently completes at 61.50 s. Stair frames show their respective labelled terminal times. Replays load recorded qpos/qvel and compute forward kinematics at each saved state. Both methods use the same camera orientation and distance, centered on their recorded robot positions.