KAGGLEYEs ACADEMY E-PRINT REGISTRY
ARCHIVAL PREPRINT / OPEN SCHOLARLY RECORD

KAGG RECORD
KAGG:2608.00005

Range-Plane Abstractions for Legged Robot Navigation in Cluttered Interiors

Omar Farouk • **Maximilian Weber** • **Lin Zhao**

Perception and Control Group, Technical University of Munich, Munich, Germany

PUBLISHED	REGISTERED	RECORD TYPE
August 9, 2026	August 10, 2026	Preprint

ABSTRACT

Legged navigation in building interiors needs more than a floor map but much less than a dense volumetric forecast. Two-dimensional floor projections discard the step heights and overhead clearances that decide whether a quadruped must crawl, walk normally, or raise its body, whereas full three-dimensional future-range prediction spends most of its output on elevations that no motion mode queries. This paper introduces range-plane abstractions: a small set of control-aligned planes, namely floor, low-obstacle, and headroom, each carrying its own future-range map along the directions the robot is predicted to execute. The height bands of the planes are anchored to the body envelope of the active motion mode (crawl, standard, or raised), and the local planner selects the plane set together with the mode, so that each feasibility query reduces to a first-hit range and an extremal height within one band. In Isaac Lab apartment suites (8 floor plans, 5 seeds, 30 trials each) and on a 900 m four-floor office-building course traversed by an ANYmal, the abstraction reduces predictor output dimensionality by 61.4 percent relative to a dense reference, against 56.6 percent for full three-dimensional future-range prediction. It also lowers p95 query latency from 9.2 to 8.9 ms, uses 842k instead of 931k predictor parameters, and raises task success from 85.9 to 90.2 percent. Ablations attribute the largest share of the gain to the low-obstacle plane; recorded failures, including a late mode switch at a 12 cm step, are analyzed.

INDEX TERMS legged robot navigation, future range prediction, range-plane abstraction, motion modes, quadrupedal robots, indoor navigation

KAGG NOTE Interior navigation geometry that matters to a legged robot lives in a few body-anchored height bands; predicting future range per band rather than per elevation keeps the queries the planner needs and discards the rest.

01 Introduction

Quadrupedal robots are increasingly deployed inside buildings for inspection, monitoring, and delivery. Interiors are geometrically benign in one sense, since most of the traversable surface is a flat floor, but they are hostile in another: kitchens, plant rooms, and service corridors contain counters, chair legs, door thresholds, isolated steps, pipe racks, and low ceilings. For a legged robot, these obstacles do not all mean the same thing. A threshold can be stepped over, a table can be passed underneath by lowering the body, and a short step may require a slower, more careful gait. Learned locomotion controllers already expose such behaviors as discrete or continuous motion modes [5, 11, 10]; the perception module must tell the planner which mode makes a given direction passable.

Two representation choices dominate current practice. The first projects geometry onto the floor, producing a two-dimensional cost, elevation, or free-space map [15, 16, 17]. Floor projections are compact and fast, but they collapse the vertical axis: an overhang at 0.6 m and a box at 0.2 m can map to the same occupied cell, and a step edge is reduced to a cost value without the signed height that determines whether a gait can negotiate it. The second choice predicts range or occupancy densely in three dimensions [24, 20, 21]. Recent work on control-aligned future range, in particular FutureRay [1], shows that predicting free range along the robot's own predicted control-aligned directions makes obstacle queries consistent with what the controller can execute. Extending such a predictor to a full elevation fan, however, multiplies its output by the number of elevation bins, even though a planner choosing among a handful of motion modes only ever asks a few questions about the vertical axis: is the floor continuous, is there something the legs cannot clear, and is there something the body cannot pass under?

This paper takes that observation literally. We introduce *range-plane abstractions*: a small set of control-aligned planes (floor, low-obstacle, and headroom), each carrying its own future-range map in the FutureRay sense. The height bands of the planes are anchored to the body envelope of the active motion mode, and the local planner selects the plane configuration together with the mode (crawl, standard, raised). Under a stated envelope assumption, the free-range query of the active mode is recovered exactly from the planes, while extremal heights stored per plane let the planner propose mode switches without evaluating every elevation.

The contributions are as follows.

1. A formulation of navigation-relevant vertical queries for multi-mode legged robots, and the range-plane abstraction that answers them with three control-aligned future-range maps whose height bands are anchored to the active mode's body envelope.
2. A query-sufficiency result (Proposition 1 in Section 4) stating conditions under which the active-mode free range com-

KAGGLEYEs ACADEMY • E-PRINT REGISTRY • KAGG:2608.00005

puted from range planes equals the one computed from a full three-dimensional future-range field, together with a conservative extension to alternative modes based on extremal heights and gated additional layers.

3. A mode-aware sampling-based local planner that queries the planes, enforces stopping-distance constraints, and switches between crawl, standard, and raised modes with verification and hysteresis.

4. An evaluation in Isaac Lab apartment suites and on a 900 m four-floor office-building course with an ANYmal, reporting predictor output dimensionality, query latency, parameter count, and task success against full three-dimensional prediction, floor projection, and present-range baselines, together with ablations and an analysis of recorded failures.

02 Related Work

2.1 Control-aligned future range

FutureRay [1] is the direct starting point of this work. Rather than predicting a map in a fixed world or body frame, FutureRay predicts the future free range along the directions that the robot itself is predicted to follow under candidate controls. Obstacle queries are therefore expressed in the same coordinates as the decision the controller faces: how far it can travel along an executable motion before contact. This alignment avoids the mismatch that arises when a planner must convert a generic map into constraints on commands, and it gives the predictor a learning target that discounts geometry the robot will never approach. FutureRay was developed for agile quadruped navigation, where the main difficulty is speed and anticipation. The present work keeps its control-aligned future-range target unchanged but asks a different question: how much vertical structure must such a predictor carry when the robot can change its body configuration? Range planes can be read as a structured factorization of a FutureRay-style output along the height axis, chosen so that each factor corresponds to a constraint of a motion mode. Where FutureRay treats the robot as a single body moving through free range, our formulation treats the body envelope as a mode-dependent variable and keeps only the height bands that enter its constraints.

2.2 Learned perceptive locomotion and local navigation

Reinforcement learning has made robust legged locomotion practical. Hwangbo et al. [2] trained ANYmal policies in simulation with a learned actuator model, Lee et al. [3] obtained blind but robust locomotion over challenging terrain via teacher-student training, and Miki et al. [4] fused exteroception with proprioception through a belief encoder so that the policy can discount unreliable height samples. Massively parallel simulation [5, 39] reduced training time to minutes, and adaptation modules such as RMA [6] handle changes in dynamics online. Egocentric-vision policies [7, 21] and parkour controllers [8, 9] extend these skills to stairs, gaps, and climbing. These works define the low-level capabilities that our planner relies on; they consume terrain observations local to the feet and do not address which vertical structure a navigation-level predictor must expose.

Closer to our setting, several works learn navigation on top of or jointly with locomotion. Rudin et al. [11] learn locomotion and local navigation end-to-end, and ANYmal Parkour [10] combines a perception module, a navigation module, and a set of locomotion skills, including crouching under obstacles, with a high-level selector. Hoeller et al. [12] learn a compact state representation for navigation among clutter and moving obstacles, and Zhang et al. [13] train a navigation policy that remains functional when perception is compromised. Fu et al. [22] and ViNL [23] couple vision and proprioception for legged navigation over obstacles, and Lee et al. [14] demonstrate long-range autonomy for wheeled-legged robots. Skill-selection architectures such as ANYmal Parkour are the closest in spirit to our mode selection; they rely on a learned latent or a height-scan observation, whereas we expose an explicit, geometrically interpretable query interface whose dimensionality is set by the number of modes rather than by the sensor resolution.

2.3 Maps and volumetric representations for planning

Classical legged navigation stacks maintain robot-centric elevation maps [15] and plan over them with reachability or traversability models [16, 17], while model-based controllers optimize footholds and base motion against the same maps [18, 19]. Elevation maps store one height per cell and therefore cannot represent overhangs, which is precisely the case of a table, a pipe rack, or a low ceiling; the floor-projection baseline in Section 5 is representative of this family. Volumetric alternatives such as signed distance fields [24] and learned neural scene representations [20, 21] preserve full three-dimensional structure, and aerial planners operate directly on such fields [25]. Their cost is output size and query time. Our full three-dimensional future-range baseline represents this end of the spectrum in control-aligned coordinates. Sampling-based model predictive control, notably MPPI [26], provides the template for our planner: candidate command sequences are rolled out and scored, and the perception interface must answer many such queries per cycle, which is why output dimensionality and query latency matter.

2.4 Factorized planar representations of 3D scenes

Outside robotics, factorizing three-dimensional fields into planes is an established strategy for reducing memory while preserving query accuracy. Bird's-eye-view perception lifts images to a ground plane [27, 28]; tri-perspective view [29] adds two orthogonal planes to recover height information that a single bird's-eye-view plane loses, and occupancy networks [30] predict dense voxels at higher cost. In generative modeling and neural rendering, tri-planes [31] and K-Planes [32] represent volumes as products or sums of planar feature grids. Range planes share the motivation of these factorizations but differ in two respects: the planes are horizontal bands defined by the robot's body envelope rather than orthogonal coordinate planes, and each plane carries a control-aligned future-range target rather than features or occupancy. The factorization is thus task-derived: it keeps exactly the height partitions that the planner's constraints distinguish.

2.5 Navigation systems, simulation, and transfer

Field systems such as Autonomous Spot [36] and CERBERUS [37] demonstrate legged autonomy in subterranean and industrial settings and highlight the importance of conservative local planning under perception uncertainty. Learning-based navigation systems trained from experience [33] or from heterogeneous multi-robot data [34, 35] learn goal-directed behavior at a higher level of abstraction and typically assume a fixed body. Our experiments use Isaac Lab, the successor of Orbit [38] built

KAGGLEYEs ACADEMY • E-PRINT REGISTRY • KAGG:2608.00005

on GPU-parallel physics [39]; policies of the kind we deploy are typically trained with PPO [40]. Truong et al. [41] show that navigation-level sim-to-real transfer can benefit from abstracting away low-level simulation detail, an argument that also favors compact, geometry-level interfaces such as range planes.

03 Problem Formulation

3.1 Robot, motion modes, and commands

The robot operates in a gravity-aligned local frame attached to its base. At planning step t it receives an exteroceptive observation o_t (a fused point cloud) and a goal g_t expressed in the local frame. Its locomotion controller accepts a base twist command $u = (v_x, v_y, \omega) \in \mathcal{U}_m$ together with a motion mode

$$m \in \mathcal{M} = \{\mathrm{CR}, \mathrm{ST}, \mathrm{RA}\}, \tag{1}$$

for crawl, standard, and raised. Each mode has a body envelope

$$\mathcal{E}_m = (c_m,\ \tau_m,\ w_m,\ s_m), \tag{2}$$

where c_m is the obstacle height the legs can clear in that mode, τ_m is the height of the top of the body and its sensors above the floor, w_m is the half-width of the swept footprint, and s_m is the largest step height (up or down) the mode can negotiate. Lowering the body reduces τ_m and typically c_m; raising it increases both. The admissible command set $\mathcal{U}_m$ is mode dependent, with smaller speed limits in crawl and raised modes than in standard mode.

3.2 Control-aligned future range

Following FutureRay [1], the planner evaluates K candidate command sequences $U^{(k)} = (u_0^{(k)}, \dots, u_{H-1}^{(k)})$, $k = 1, \dots, K$, over H steps of duration Δt. Integrating the base kinematics gives predicted poses

$$\xi_{h+1}^{(k)} = \xi_h^{(k)} \oplus \left(\Delta t\, u_h^{(k)}\right), \qquad \xi_0^{(k)} = \xi_t, \tag{3}$$

where $\oplus$ denotes composition in $SE(2)$. Let $\gamma_h^{(k)}(\sigma)$ be the path obtained by continuing the motion of $U^{(k)}$ from $\xi_h^{(k)}$ for arc length $\sigma \geq 0$, and let $\mathcal{C}_h^{(k)}(\sigma; w)$ be the planar corridor of half-width w swept along this path. For a set of obstacle points $\mathcal{P}_t \subset \mathbb{R}^3$ with heights $z(p)$ measured relative to the local floor, and a height set $B \subset \mathbb{R}$, the first-hit range is

$$\rho_h^{(k)}(B; w) = \min\Big(r_{\max},\ \inf\big\{\sigma : \exists\, p \in \mathcal{P}_t,\ p_{xy} \in \mathcal{C}_h^{(k)}(\sigma; w),\ z(p) \in B\big\}\Big), \tag{4}$$

with $r_{\max}$ the range cap of the local predictor. A full three-dimensional future-range predictor estimates $\rho_h^{(k)}(B_e; w)$ for E elevation bins $B_1, \dots, B_E$ that tile the vertical extent of interest; its output dimensionality is $D_{\mathrm{3D}} = KHE$.

3.3 Navigation-relevant queries

The planner needs, for each candidate k, horizon step h, and mode m, the free range

$$r_h^{m,(k)} = \min\Big(\rho_h^{(k)}\big((c_m, \tau_m + \delta];\ w_m\big),\ \phi_h^{(k)}(s_m;\ w_m)\Big), \tag{5}$$

where $\delta > 0$ is a clearance margin and $\phi_h^{(k)}(s; w)$ is the range to the first floor discontinuity whose magnitude exceeds s (a step up, a step down, or a drop-off). Equation (5) encodes an envelope assumption, stated formally below. It also needs two mode-switch queries: whether the first obstacle blocking mode m could be stepped over by a mode with larger $c_{m'}$, and whether it could be passed underneath by a mode with smaller $\tau_{m'}$.

ASSUMPTION A1 (ENVELOPE)

For mode m, a motion along $\gamma_h^{(k)}$ up to arc length σ is collision free and dynamically negotiable if and only if no obstacle point inside $\mathcal{C}_h^{(k)}(\sigma; w_m)$ has height in $(c_m, \tau_m + \delta]$ and no floor discontinuity larger than s_m lies on the corridor.

A1 abstracts the robot as a vertical slab that swings legs over anything below c_m; it ignores leg placement detail, which is delegated to the locomotion controller [4, 11].

3.4 Objective

Given o_t and g_t, the goal is a predictor f_θ and a planner that select (m_t, U_t) such that the robot reaches the goal without collision, while (i) the predictor output dimensionality D is small relative to a dense reference D_{ref}, (ii) the p95 latency of answering the queries of one planning cycle is small compared with the planning period, and (iii) the free-range query (5) for the active mode is preserved. We report the relative reduction

$$\eta = 1 - D/D_{\mathrm{ref}}, \tag{6}$$

where D counts the values emitted to the planner per cycle, averaged over an episode, and D_{ref} is a common dense reference shared by all compared predictors.

04 Method

4.1 Range planes

For the active mode m, we partition the vertical axis into three bands anchored to $\mathcal{E}_m$:

$$\begin{aligned} F_m &= [-\bar{z}_{\mathrm{s}},\ c_m], \\ L_m &= (c_m,\ \lambda_m], \\ H_m &= (\lambda_m,\ \bar{\tau}], \end{aligned} \tag{7}$$

where $\bar{z}_{\mathrm{s}}$ bounds the depth of step-downs considered, $\lambda_m \in (c_m, \tau_m + \delta)$ is the split height between low obstacles and overhangs (we use the underside of the trunk in mode m), and $\bar{\tau} = \max_{m'} \tau_{m'} + \delta$ is the highest clearance any mode can require, so that the headroom band always covers the raised mode. Each band carries one control-aligned future-range map:

- *Floor plane* $\mathcal{F}$: the range $R_{k,h}^{\mathcal{F}} = \phi_h^{(k)}(s_m; w_m)$ to the first support discontinuity exceeding s_m, and its signed height $S_{k,h}$ (positive for steps up).
- *Low-obstacle plane* $\mathcal{L}$: the range $R_{k,h}^{\mathcal{L}} = \rho_h^{(k)}(L_m; w_m)$ and the maximum height $Z_{k,h}^{\mathcal{L}}$ of the obstacle at the first hit.
- *Headroom plane* $\mathcal{H}$: the range $R_{k,h}^{\mathcal{H}} = \rho_h^{(k)}(H_m \cap (c_m, \bar{\tau}]; w_m)$ and the minimum underside height $Z_{k,h}^{\mathcal{H}}$ of the obstacle at the first hit.

Each plane therefore has $C = 2$ channels per entry, giving $6KH$ values per cycle in the single-layer case. The extremal-height channels are what distinguishes range planes from a single slab: they encode, for the first obstacle in each band, whether a different envelope would clear it.

KAGGLEYEs ACADEMY • E-PRINT REGISTRY • KAGG:2608.00005

4.2 Query sufficiency

PROPOSITION 1 (ACTIVE-MODE SUFFICIENCY)

Under A1, and for bands defined by (7) with $\lambda_m < \tau_m + \delta \le \bar{\tau}$, the active-mode free range satisfies

$$r_h^{m,(k)} = \min\big(R^{\mathcal{F}}_{k,h},\, R^{\mathcal{L}}_{k,h},\, \rho_h^{(k)}((\lambda_m, \tau_m{+}\delta]; w_m)\big),$$

and the last term equals $R^{\mathcal{H}}_{k,h}$ whenever $Z^{\mathcal{H}}_{k,h} \le \tau_m + \delta$.

Proof sketch. By A1 the blocking height set for mode m is $(c_m, \tau_m + \delta] = L_m \cup (\lambda_m, \tau_m + \delta]$. The first-hit operator (4) is a minimum over points, so the first hit of a union of disjoint height sets is the minimum of their first hits. The floor term is defined identically in (5) and in $\mathcal{F}$. For the headroom band, the first hit in $(\lambda_m, \bar{\tau}]$ coincides with the first hit in $(\lambda_m, \tau_m + \delta]$ when its underside lies at or below $\tau_m + \delta$; otherwise the obstacle is above the body and does not block. □

When $Z^{\mathcal{H}}_{k,h} > \tau_m + \delta$, the first headroom hit is passable and the true blocking range may lie farther away. The single-layer planes then provide only the lower bound $r_h^{m,(k)} \ge \min(R^{\mathcal{F}}, R^{\mathcal{L}}, R^{\mathcal{H}})$, which the planner uses as a conservative value. The same reasoning applies to an alternative mode m': a first hit whose extremal height clears $\mathcal{E}_{m'}$ (that is, $Z^{\mathcal{L}} \le c_{m'}$ or $Z^{\mathcal{H}} > \tau_{m'} + \delta$) may hide a blocking obstacle behind it. Two mechanisms address this.

Gated layers. The low-obstacle and headroom planes may emit up to Λ layers, where layer j stores the range and extremal height of the j-th distinct hit in the band along the corridor. A gate $\hat{g}_{p,j,k,h} \in [0, 1]$ indicates whether layer j exists, and an entry is emitted to the planner only if $\hat{g} > 0.5$. The emitted dimensionality is then

$$D_{\mathrm{RP}} = \sum_{p \in \{\mathcal{F}, \mathcal{L}, \mathcal{H}\}} \sum_{j=1}^{\Lambda_p} C \cdot \big|\{(k, h) : \hat{g}_{p,j,k,h} > 0.5\}\big|, \quad (8)$$

with $\Lambda_{\mathcal{F}} = 1$ and the first layer of each plane always emitted. In open rooms, D_{RP} stays close to $6KH$; in clutter, more layers activate and D_{RP} grows. This makes the reduction η in (6) scene dependent, as observed in Section 5.5. With Λ layers, Proposition 1 extends to every mode for which at most $\Lambda - 1$ passable hits precede the first blocking hit in each band.

Verification before switching. A mode switch proposed from extremal heights is committed only after the predictor has been evaluated with the band geometry of the proposed mode, so that Proposition 1 holds for the mode actually executed (Section 4.5).

4.3 Predictor

The predictor f_θ maps the observation, the candidate set, and the mode to the plane outputs,

$$\{R^p_{j,k,h},\, Z^p_{j,k,h},\, \hat{g}_{p,j,k,h}\} = f_\theta\big(o_t,\, \{U^{(k)}\}_{k=1}^{K},\, m\big). \quad (9)$$

It has three parts. A pillar-style encoder discretizes the gravity-aligned point cloud into vertical columns on a robot-centric grid and produces a bird's-eye-view feature map; unlike a pure floor projection, each pillar feature is computed from the heights of its points, so vertical structure is retained in the features even though the grid is planar. A mode embedding modulates the encoder output through feature-wise affine transformation, so that the same trunk serves all three band geometries. A query decoder samples features along each predicted corridor $\mathcal{C}_h^{(k)}$, adds an encoding of (k, h) and of the commanded velocity, and applies one small multilayer perceptron per plane. Ranges are parameterized as $R = r_{\max}\,\mathrm{sigmoid}(\cdot)$ and heights relative to the band limits, which keeps outputs bounded and band consistent. Because the plane heads replace an E-way elevation head, the parameter count is lower than that of the full three-dimensional variant built on the same trunk (842k against 931k in Table 1).

4.4 Training

Supervision is generated in simulation by evaluating (4) with privileged scene geometry for the band geometry of each mode, which yields targets $R^{p,*}$, $Z^{p,*}$, and layer existence labels g^*. The loss is

$$\mathcal{L}(\theta) = \sum_{p,j,k,h} g^*_{p,j,k,h} \Big[\beta_z \big|\hat{Z}^p_{j,k,h} - Z^{p,*}_{j,k,h}\big| + \ell_\kappa\big(\log(1{+}\hat{R}^p_{j,k,h}) - \log(1{+}R^{p,*}_{j,k,h})\big)\Big] + \beta_g\, \mathrm{BCE}(\hat{g}, g^*), \quad (10)$$

where ℓ_κ is the Huber loss with threshold κ, the logarithm emphasizes accuracy at short range where collisions occur, and $\beta_z, \beta_g > 0$ weight the height and gate terms. Height errors are only penalized where a hit exists. Modes are sampled uniformly during training so that the shared trunk learns all three band geometries.

4.5 Mode-aware local planner

The planner runs at a period $T_p = 50$ ms and sends twist commands and the mode to a learned locomotion policy that tracks them at 50 Hz, in the style of [5, 11]. At each cycle it samples K command sequences around the previous solution, as in MPPI [26], rolls them out with (3), queries the planes of the active mode, and scores each candidate with

$$J(k, m) = w_g \big\| g_t - \xi_H^{(k)} \big\| + w_c \sum_{h=0}^{H-1} \exp\Big(-\frac{\hat{r}_h^{m,(k)} - \sigma_h^{(k)}}{\sigma_c}\Big) + w_u \sum_h \big\| u_h^{(k)} - u_{h-1}^{(k)} \big\|^2 + w_s \mathbb{1}[m \ne m_{t-1}], \quad (11)$$

where $\hat{r}_h^{m,(k)}$ is the predicted free range from Proposition 1 (or its conservative lower bound), $\sigma_h^{(k)}$ is the arc length the candidate travels between steps h and $h + 1$, σ_c sets the softness of the clearance term, and the w terms are nonnegative weights. A candidate is admissible only if it can stop within the predicted free range,

$$\hat{r}_h^{m,(k)} \ge \sigma_h^{(k)} + \frac{\|v_h^{(k)}\|^2}{2a_m} + d_{\mathrm{safe}} \quad \forall h, \quad (12)$$

with a_m the braking deceleration available in mode m and d_{safe} a fixed margin. The executed command is the cost-weighted average of admissible candidates; if none is admissible, the robot stops.

Mode switching. A switch to m' is proposed when the best admissible candidate in mode m is blocked by an obstacle

KAGGLEYEs ACADEMY • E-PRINT REGISTRY • KAGG:2608.00005

that $\mathcal{E}_{m'}$ would clear according to the extremal heights, that is, $Z^{\mathcal{H}} > \tau_{m'} + \delta$ (crawl under), $Z^{\mathcal{L}} \le c_{m'}$ (raise over), or a floor discontinuity $|S| \le s_{m'}$ (careful step). The predictor is then also evaluated with the band geometry of m', and the switch is committed if

$$\min_k J(k, m') < \min_k J(k, m) - \Delta_J \tag{13}$$

holds for N_{h} consecutive cycles. The hysteresis margin Δ_J and the count N_{h} suppress chattering between modes near band boundaries, at the price of a reaction delay of at least $N_{\mathrm{h}} T_p$ after a switch first becomes favorable. Speed limits are mode dependent: standard mode permits up to 1.0 m/s forward, while crawl and raised modes are limited to lower speeds because the reduced leg workspace leaves less margin for disturbance rejection.

4.6 Why three planes

The number of planes equals the number of distinct vertical constraints in A1 for the given mode set: floor continuity (s_m), leg clearance (c_m), and body clearance (τ_m). Adding a plane would only help if the planner had a fourth constraint type, for example a mode that ducks under an obstacle and simultaneously steps over another, which would require the joint distribution of low and high hits. Removing a plane merges two constraints, and the ablation in Section 5.4 quantifies the cost.

05 Experiments

5.1 Setup and platform

Simulated evaluation uses an Isaac Lab apartment suite consisting of 8 floor plans that include kitchens, stairwells, low-ceiling corridors, and step thresholds. Each floor plan is evaluated with 5 seeds and 30 trials per seed, that is, 1,200 simulated trials per method. Real-world evaluation uses an ANYmal on a 900 m course through a four-floor office building, which includes a corridor with 1.9 m headroom and a 12 cm step. The robot uses the three motion modes crawl, standard, and raised throughout.

5.2 Baselines and metrics

All methods share the planner of Section 4.5 and differ in the perception interface:

- *Full 3D future-range prediction*: a FutureRay-style predictor [1] that outputs future range for a fan of elevation bins; feasibility follows from (5) by taking the minimum over the bins in the blocking set.
- *2D floor projection only*: future range computed on a floor projection of the obstacles inside a fixed height window above the floor, as in two-dimensional costmaps, without band structure or extremal heights.
- *Present range only*: range measured from the current pose, without prediction along future control-aligned directions.
- *Ours*: range planes with mode-anchored bands, gated layers, and verified mode switching.

We report the reduction η of predictor output dimensionality (6) relative to the common dense reference, the p95 latency of a planning-cycle query, the number of predictor parameters, and task success, defined as reaching the goal without collision.

5.3 Main results

Table 1 summarizes the comparison.

Dimensionality. Range planes reach a 61.4% reduction relative to the dense reference, 4.8 percentage points more than full three-dimensional future-range prediction (56.6%) and 4.4 points more than the present-range baseline (57.0%). The floor-projection baseline has the smallest reduction (54.1%), which indicates that dropping the vertical axis does not by itself produce the most compact interface under our accounting.

Latency and size. The p95 query latency is 8.9 ms, against 9.2 ms for full 3D prediction, 9.4 ms for present range, and 9.9 ms for floor projection. All values lie well below the 50 ms planning period, so the latency differences of 0.3 to 1.0 ms are not decisive on their own. The predictor has 842k parameters, 89k (about 9.6%) fewer than the full 3D variant and 73k fewer than the present-range variant.

Task success. Task success is 90.2%, an improvement of 4.3 percentage points over full 3D prediction (85.9%), 8.3 points over present range (81.9%), and 8.8 points over floor projection (81.4%). The gain over full 3D prediction is notable because that baseline has access to strictly more vertical information. A plausible explanation, consistent with Proposition 1, is that the plane outputs place the predictor's capacity on exactly the quantities the feasibility test consumes, whereas errors in any of the E elevation bins of the full 3D predictor can propagate through the minimum in (5); the present data do not isolate this mechanism. The compiled results report a single success rate per method; they do not break success down by floor plan or separate simulated from real trials, so per-scene conclusions are not drawn here.

5.4 Ablation

Table 2 removes planes from the abstraction.

Removing the low-obstacle plane costs 6.7 percentage points (90.2% to 83.5%), the largest single-plane effect. Without it, obstacles between the leg-clearance height and the trunk are either merged into the floor band, where they appear as discontinuities without the height needed to decide whether the raised mode clears them, or into the headroom band, where they are treated as overhangs. Removing the headroom plane costs 2.3 points (87.9%). The smaller effect is consistent with overhangs being rarer than low clutter in the evaluated scenes, but where they occur they are safety critical, as the first failure case below shows. Keeping only the floor plane reduces success to 76.8%, 13.4 points below the full abstraction and 4.6 points below the 2D floor-projection baseline (81.4%) in Table 1. The single-plane variant is thus not a substitute for the vertical information that the low-obstacle and headroom planes provide.

5.5 Failure cases

Low-ceiling corridor (floor plan 6). In a low-ceiling corridor of floor plan 6, the 2D floor-projection baseline collides with a pipe rack in 9 of 30 trials (30%). The pipe rack is an overhang; the collisions are consistent with it lying outside the height window that the projection keeps, so that its footprint does not appear in the projected map. The headroom plane of our method detects the rack as a first hit in H_m with an underside below $\tau_m + \delta$, and the planner either avoids it or proposes the crawl mode.

12 cm step on the real course. On the real course, the standard-mode plane set misclassified the 12 cm step as a step-down, that is, the predicted signed height S had the wrong sign. The switch to the crawl mode happened 0.8 s late. No collision occurred, and the minimum clearance was 5 cm. This

KAGGLEYEs ACADEMY • E-PRINT REGISTRY • KAGG:2608.00005

Table 1: Main comparison. Output dimensionality reduction is relative to a common dense reference (higher is better); latency is the p95 query latency; success is the task success rate.

Method	Output dim. reduction (%)	Query latency p95 (ms)	Predictor parameters (k)	Task success (%)
Full 3D future-range prediction	56.6	9.2	931	85.9
2D floor projection only	54.1	9.9	962	81.4
Present range only	57.0	9.4	915	81.9
Ours (range planes)	**61.4**	**8.9**	**842**	**90.2**

Table 2: Ablation of the range-plane set (task success).

Variant	Task success (%)
Ours (range-plane abstraction)	90.2
w/o low-obstacle plane	83.5
w/o headroom plane	87.9
single plane (floor only)	76.8

case exposes two weaknesses. First, the signed step channel is the most sensitive output of the floor plane, because a sign error does not change the range $R^{\mathcal{F}}$ but changes which mode the planner proposes. Second, the verification and hysteresis of (13) trade switching stability for reaction time; the logs do not separate how much of the 0.8 s delay is due to the sign error and how much to the hysteresis window.

Cluttered kitchen counters. In the kitchen scene with cluttered counters, the output dimensionality reduction drops to 54%, because more low obstacles activate additional layers of the low-obstacle plane in (8). This is the expected behavior of the gated layers: dimensionality grows where the scene requires it. The run logs record that the configuration still corresponds to a $4.1\times$ memory saving. We note that a 54% reduction in output dimensionality alone would correspond to a factor of about 2.2 ($1/0.46$); the $4.1\times$ figure is therefore a separately logged memory measurement that includes more than the emitted output values, and the logs do not decompose it further. We report it as recorded.

06 Limitations

The abstraction rests on Assumption A1, which models the body as a vertical slab with a mode-dependent leg-clearance height. Obstacles whose blocking effect depends on their shape rather than on their height interval, such as slanted surfaces or narrow gaps that only some leg placements can use, are outside this model. The single-layer planes encode only the first hit per band; gated layers mitigate this but make the output size scene dependent, as the kitchen case shows, so memory budgets must be planned for the cluttered case rather than the average case. The signed step channel was the source of the real-world misclassification, and its reliability on unusual step geometry has not been characterized beyond the recorded course. The verified, hysteretic mode switch adds reaction delay, which was 0.8 s in the recorded step case. Finally, the evaluation covers interiors with three motion modes and a single robot. The compiled results report aggregate success rates and do not provide per-scene or per-domain breakdowns or confidence intervals, so differences of a few percentage points should be interpreted with corresponding caution. Extending the abstraction to outdoor terrain, where the floor itself is not planar, would require replacing the floor band with a terrain-relative band.

07 Conclusion

Range-plane abstractions replace a full elevation fan of control-aligned future range with three planes whose height bands are tied to the body envelope of the active motion mode. Under a stated envelope assumption, the active-mode free-range query is preserved exactly, and extremal heights per plane support mode-switch proposals that are verified before commitment. In simulated apartment suites and on a real office-building course with an ANYmal, the abstraction achieved a 61.4% output dimensionality reduction relative to a dense reference, a p95 query latency of 8.9 ms, 842k predictor parameters, and 90.2% task success, compared with 56.6%, 9.2 ms, 931k, and 85.9% for full three-dimensional future-range prediction. The ablations indicate that the low-obstacle plane carries the largest share of the benefit, and the recorded failures point to the signed step channel and the switching delay as the next components to improve.

References

[1] T. Zang, S. Wang, H. Huang, and S. Cheng. FutureRay: Control-Aligned Future Range for Agile Quadruped Navigation. arXiv:2608.01197, 2026.

[2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning Agile and Dynamic Motor Skills for Legged Robots. arXiv:1901.08652, 2019.

[3] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. arXiv:2010.11251, 2020.

[4] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Robust Perceptive Locomotion for Quadrupedal Robots in the Wild. arXiv:2201.08117, 2022.

[5] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. arXiv:2109.11978, 2021.

[6] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. arXiv:2107.04034, 2021.

[7] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. arXiv:2211.07638, 2022.

[8] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot Parkour Learning. arXiv:2309.05665, 2023.

[9] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. arXiv:2309.14341, 2023.

[10] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. arXiv:2306.14874, 2023.

[11] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. arXiv:2209.12827, 2022.

[12] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter. Learning a State Representation and Navigation in Cluttered and Dynamic Environments. arXiv:2103.04351, 2021.

KAGGLEYEs ACADEMY • E-PRINT REGISTRY • KAGG:2608.00005

[13] C. Zhang, J. Jin, J. Frey, N. Rudin, M. Mattamala, C. Cadena, and M. Hutter. Resilient Legged Local Navigation: Learning to Traverse with Compromised Perception End-to-End. arXiv:2310.03581, 2023.

[14] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter. Learning Robust Autonomous Navigation and Locomotion for Wheeled-Legged Robots. arXiv:2405.01792, 2024.

[15] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter. Elevation Mapping for Locomotion and Navigation using GPU. arXiv:2204.12876, 2022.

[16] L. Wellhausen and M. Hutter. ArtPlanner: Robust Legged Robot Navigation in the Field. arXiv:2303.01420, 2023.

[17] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter. Fast Traversability Estimation for Wild Visual Navigation. arXiv:2305.08510, 2023.

[18] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter. Perceptive Locomotion through Nonlinear Model Predictive Control. arXiv:2208.08373, 2022.

[19] F. Jenelten, R. Grandia, F. Farshidian, and M. Hutter. TAMOLS: Terrain-Aware Motion Optimization for Legged Systems. arXiv:2206.14049, 2022.

[20] D. Hoeller, N. Rudin, C. Choy, A. Anandkumar, and M. Hutter. Neural Scene Representation for Locomotion on Structured Terrain. arXiv:2206.08077, 2022.

[21] R. Yang, G. Yang, and X. Wang. Neural Volumetric Memory for Visual Locomotion Control. arXiv:2304.01201, 2023.

[22] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak. Coupling Vision and Proprioception for Navigation of Legged Robots. arXiv:2112.02094, 2021.

[23] S. Kareer, N. Yokoyama, D. Batra, S. Ha, and J. Truong. ViNL: Visual Navigation and Locomotion Over Obstacles. arXiv:2210.14791, 2022.

[24] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning. arXiv:1611.03631, 2016.

[25] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao. EGO-Planner: An ESDF-free Gradient-based Local Planner for Quadrotors. arXiv:2008.08835, 2020.

[26] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving. arXiv:1707.02342, 2017.

[27] J. Philion and S. Fidler. Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. arXiv:2008.05625, 2020.

[28] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai. BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers. arXiv:2203.17270, 2022.

[29] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu. Tri-Perspective View for Vision-Based 3D Semantic Occupancy Prediction. arXiv:2302.07817, 2023.

[30] W. Tong, C. Sima, T. Wang, L. Chen, S. Wu, H. Deng, Y. Gu, L. Lu, P. Luo, D. Lin, and H. Li. Scene as Occupancy. arXiv:2306.02851, 2023.

[31] E. R. Chan et al.. Efficient Geometry-aware 3D Generative Adversarial Networks. arXiv:2112.07945, 2021.

[32] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. arXiv:2301.10241, 2023.

[33] G. Kahn, P. Abbeel, and S. Levine. BADGR: An Autonomous Self-Supervised Learning-Based Navigation System. arXiv:2002.05700, 2020.

[34] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot. arXiv:2210.03370, 2022.

[35] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A Foundation Model for Visual Navigation. arXiv:2306.14846, 2023.

[36] A. Bouman et al.. Autonomous Spot: Long-Range Autonomous Exploration of Extreme Environments with Legged Locomotion. arXiv:2010.09259, 2020.

[37] M. Tranzatto et al.. CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge. arXiv:2201.07067, 2022.

[38] M. Mittal et al.. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. arXiv:2301.04195, 2023.

[39] V. Makoviychuk et al.. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. arXiv:2108.10470, 2021.

[40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. arXiv:1707.06347, 2017.

[41] J. Truong, M. Rudolph, N. Yokoyama, S. Chernova, D. Batra, and A. Rai. Rethinking Sim2Real: Lower Fidelity Simulation Leads to Higher Sim2Real Transfer in Navigation. arXiv:2207.10821, 2022.

KAGGLEYEs ACADEMY • E-PRINT REGISTRY • KAGG:2608.00005