International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(13s) 2026 pp. 1138-1154
https://doi.org/10.70917/ijcisim-2026-4151
Article

Detection-Guided Terrain-Aware Autonomous Navigation and Active Target Detection for Quadruped Robots

Jyoti Kadadevarmath

Department of Computer Science, Government First Grade College and PG Centre, Dharwad. Karnataka
jyoti.kmath@gmail.com

Abstract: —Autonomous navigation of quadruped robots in unstructured, cluttered terrain while simultaneously searching for targets of interest remains challenging because conventional geometric planners ignore the locomotion constraints imposed by slopes, steps, and rough ground, and because passive perception pipelines detect targets only when they happen to fall within a fixed forward field of view. This paper presents DTQ-Nav, a detection-guided, terrain-aware navigation framework that couples a compact convolutional target detector with a 2.5D traversability-aware global planner, a reactive local controller subject to a static gait-stability constraint, and an active-perception policy that steers the sensor toward unconfirmed targets. The perception module is trained and evaluated on a procedurally generated egocentric detection dataset of 6,600 scenes; it attains an average precision of 0.941 for target presence, a receiver-operating-characteristic area under the curve of 0.896, an F1-score of 0.866, and a mean bearing error of 1.55°. The navigation system is assessed over 150 paired Monte-Carlo episodes against four representative baselines (geometric A*, RRT*, a reactive DWA/APF controller, and an information-gain explorer). DTQ-Nav reaches every goal (success rate 1.00) with zero collisions, raises the target detection rate to 0.721 (versus 0.285 for passive reactive navigation) and reduces time-to-first-detection by an order of magnitude, while lowering locomotion energy by approximately 10.9% through terrain-aware routing. Ablation of components confirms that terrain-awareness governs energy efficiency, active perception governs detection, and reactive replanning governs safety. The framework is directly applicable to search-and-rescue, infrastructure inspection, and planetary exploration using legged platforms.

Keywords: —Quadruped Robot, Autonomous Navigation, Target Detection, Traversability Estimation, Active Perception, Motion Planning, Terrain-Aware Control.

1. Introduction

Legged robots have evolved from laboratory curiosities to field-deployable platforms that can traverse terrain inaccessible to wheeled or tracked vehicles. Recent advances in learned locomotion have made quadrupeds such as ANYmal and similar platforms able to robustly walk over rubble, mud, snow, and stairs [5]–[10]. Locomotion is a solved capability in many settings, so the next layer of autonomy is to enable a legged robot to decide where to go and what to look for. In applications such as search-and-rescue, subterranean exploration, infrastructure inspection and planetary science, the robot has to not only reach a goal, but also actively discover and localize targets of interest, such as survivors, defects or scientific samples, while respecting the locomotion limits of its body.

This combination of navigation and target detection reveals a disconnect between two research communities. On the planning side, classical algorithms such as A* [1], RRT* [2], the dynamic window approach [3] and artificial potential fields [4] reason about geometric occupancy but are agnostic to the cost of traversing sloped or uneven ground. A path short in the plane may be infeasible or energetically punishing for a quadruped. Modern object detectors [15]–[20] offer excellent accuracy on benchmark imagery on the perception side, but are typically deployed as passive, forward-facing pipelines: a target outside the instantaneous field of view is simply missed, however relevant it may be to the mission. Neither line of work, in isolation, addresses the central question of embodied search: how should a terrain-constrained robot move so as to maximize the probability of detecting targets while reaching its goal safely and efficiently?

1.1 Problem statement

We consider a quadruped robot operating in a previously unmapped 2.5D environment containing obstacles, untraversable terrain (steep slopes and large steps), and an unknown number of static targets. The robot is equipped with an onboard camera and a depth sensor and must navigate from a start pose to a goal pose while detecting as many targets as possible. The robot must (i) avoid collisions and untraversable terrain, (ii) minimize locomotion energy and path length, and (iii) maximize the fraction of targets detected and minimize the time to first detection. The map, the terrain traversability, and the target locations are revealed only through onboard sensing during execution.

1.2 Approach and Contributions

We propose DTQ-Nav (Detection-guided Terrain-aware Quadruped Navigation), a unified sense–plan–act framework that treats perception and locomotion as mutually informing processes. The robot incrementally builds a traversability cost map from sensed elevation, plans globally over that cost map, and follows the plan with a reactive controller that re-plans on contact with newly discovered obstacles and respects a static gait-stability margin, and actively orients its sensor toward unconfirmed targets. This paper makes the following contributions:

1. A detection-guided, terrain-aware navigation framework that closes the loop with 2.5D traversability estimation, cost-aware global planning, reactive replanning with a gait-stability constraint, and active perception.
2. A compact convolutional target detector that estimates target presence and bearing from egocentric imagery, calibrated to measured detection reliability and bearing noise to propagate perception uncertainty into the navigation sensor model.
3. A controlled Monte-Carlo evaluation against four representative planning and exploration baselines, including a component ablation that isolates the contribution of each module to safety, efficiency, and detection.
4. Quantitative evidence that terrain-aware routing reduces locomotion energy by approximately 10.9% and active perception more than doubles the target detection rate relative to passive navigation, without sacrificing success rate or collisions.

The paper is organized as follows. Section 2 reviews related work in legged locomotion, traversability estimation, navigation planning, object detection, and active perception, and identifies the research gap. Section 3 details the proposed framework, including the terrain model, the detector, the planning and control algorithms, the dataset, and the evaluation metrics. Section 4 reports detector accuracy, navigation performance, robustness to clutter, and the ablation study. Section 5 concludes and outlines real-time applications and future work.

2. Related Work

2.1 Legged locomotion and control

The deployment of quadrupeds in the field rests on robust locomotion controllers. Hutter et al. introduced ANYmal, a torque-controllable quadruped that became a standard research platform [5]. Reinforcement-learning controllers subsequently demonstrated blind locomotion over challenging natural terrain [6] and, by fusing proprioception with exteroception through an attention-based encoder, robust perceptive locomotion in the wild [7]. Rapid motor adaptation [8] enabled online adaptation to changing dynamics, while implicit terrain imagination [25] and learning on deformable ground [10] further broadened the operating envelope. Most recently, agile navigation behaviors such as parkour have been learned end-to-end [9]. These works establish that a quadruped can physically traverse difficult terrain; they focus on the locomotion layer and treat the choice of where to walk as an external input, which is precisely the layer our work addresses.

2.2 Terrain Mapping and Traversability Estimation

To plan over terrain, a robot must first estimate where it can step. Robot-centric and probabilistic elevation mapping [11] and GPU-accelerated elevation mapping [12] produce dense 2.5D height maps with uncertainty. From such maps, traversability is derived through geometric features—slope, step height, and roughness—or learned directly. Wermelinger et al. proposed navigation planning for legged robots that converts an elevation map into a traversability cost for global planning [13], and locomotion-adaptation methods such as STANCE modulate the controller according to estimated terrain compliance [14]. Our terrain model is based on this well known geometric approach: we compute slope, step and roughness from sensed elevation and combine them into a single traversability cost which feeds the global planner.

2.3 Navigation and Motion Planning

Global planners determine collision-free paths based on a known or a partially known map. A* [1] is still the algorithm of choice for grid-based planning, while sampling-based planners like RRT* [2] provide asymptotic optimality in continuous spaces. For reactive obstacle avoidance, the dynamic window approach [3] and artificial potential fields [4] generate local motion from instantaneous sensing. These methods are geometrically sound but terrain-blind: they treat all free cells as equally traversable, which is a poor assumption for a legged robot whose energy and stability depend strongly on slope and roughness. Our framework retains A*-style global search and reactive local control but operates over a terrain-aware cost rather than binary occupancy, and it closes the loop with replanning as terrain is revealed.

2.4 Object and Target Detection

Deep object detectors have progressed from region-proposal networks such as Faster R-CNN [15] to single-stage detectors including the YOLO family [16], [20], SSD [18], and focal-loss detectors [17], and to transformer-based detectors such as DETR [19]. These detectors deliver high accuracy but are designed for offline benchmark evaluation on curated imagery and are deployed passively. In a navigation context, the relevant quantities are not only category accuracy but the bearing and range to a target and, critically, the probability that a target is detected as a function of viewpoint and range. Our detector is intentionally compact and is evaluated around these navigation-relevant quantities—presence, bearing, and detection reliability versus range—so that its measured uncertainty can be injected into the planner. Beyond deep detectors designed for benchmark imagery, classical image-analysis, texture, and segmentation methods remain useful for characterizing scene content [26], [27], while convolutional and attention-based models have proven highly effective for domain-specific recognition and classification across medical imaging and acoustic signals [28]–[30].

2.5 Active Perception and Exploration

Active perception treats sensing as an action to be planned [23]. Frontier-based exploration [21] and next-best-view planning [22] choose viewpoints that maximize information gain, and learning-based explorers extend this to cluttered, unknown environments for search-and-rescue [24]. These methods improve coverage and discovery but generally optimize exploration in isolation from terrain-aware locomotion. DTQ-Nav unifies the two: the active-perception policy steers the sensor toward unconfirmed targets while the terrain-aware planner simultaneously guarantees that the chosen motion is feasible and energy-efficient for the legged platform.

2.6 Research Gap

In summary, locomotion research has made quadrupeds physically able [5]–[10], [25], terrain-mapping research has made traversability observable [11]–[14], planning research has made navigation efficient on geometric maps [1]–[4], detection research has made targets recognizable [15]–[20], and active-perception research has made discovery efficient [21]–[24]. What is lacking is a framework that combines terrain-aware navigation with active target detection in a single closed loop, and a controlled study that quantifies how much each component contributes to safety, efficiency, and detection. Table 1 compares representative prior work along these axes and highlights the gap that DTQ-Nav fills.

Table 1. *Representative prior work positioned along the axes addressed by DTQ-Nav (terrain-aware planning, active perception, and joint navigation–detection).*

Category / Reference	Focus	Terrain-aware	Active perception	Joint navigation and detection
A* [1], RRT* [2]	Geometric global planning	No	No	No
DWA [3], APF [4]	Reactive local control	No	No	No

ANYmal reinforcement learning [6], [7]	Perceptive locomotion	Yes (control)	No	No
RMA [8], DreamWaQ [25]	Adaptive locomotion	Yes (control)	No	No
Parkour [9]	Agile navigation	Yes	No	No
Elevation mapping [11], [12]	Traversability mapping	Yes (map)	No	No
Wermelinger [13]	Legged global planning	Yes	No	No
YOLO / DETR [16], [19], [20]	Object detection	No	No	No
Frontier / next-best-view [21], [22]	Exploration	No	Yes	No
Deep reinforcement-learning exploration [24]	Search in clutter	No	Yes	No
DTQ-Nav (ours)	Navigation and active detection	Yes	Yes	Yes

3. Materials and Methods

3.1 System Overview and Problem Formulation

The DTQ-Nav is designed as a closed sense-plan-act loop, as illustrated in Figure 1. At each control step the robot senses its surrounding with an onboard camera and depth sensor. The camera stream is sent to a convolutional target detector, that reports target presence and bearing. The depth stream, along with accumulated odometry, updates a 2.5D elevation map z(c). The robot infers a traversability cost map from the elevation map encoding slope, step height and roughness. A detection-guided global planner computes a minimum cost route to the goal on this cost map; a local reactive controller tracks the route, re-planning when newly sensed obstacles or untraversable cells block the route, and enforcing a static gait-stability margin. In parallel, an active-perception policy orients the sensor toward the most promising unconfirmed target. The loop repeats until the goal is reached or a step budget is exhausted.

Formally, the environment is a grid of cells c on which an elevation field z(c), an obstacle occupancy o(c), and an unknown set of K target locations are defined. The robot maintains a pose x and heading ψ and a belief over which cells are traversable. The objective is to reach the goal while maximizing the number of detected targets and minimizing locomotion energy, subject to collision- and terrain-feasibility constraints. The following subsections define each component in turn; Figure 1 shows how they interconnect.

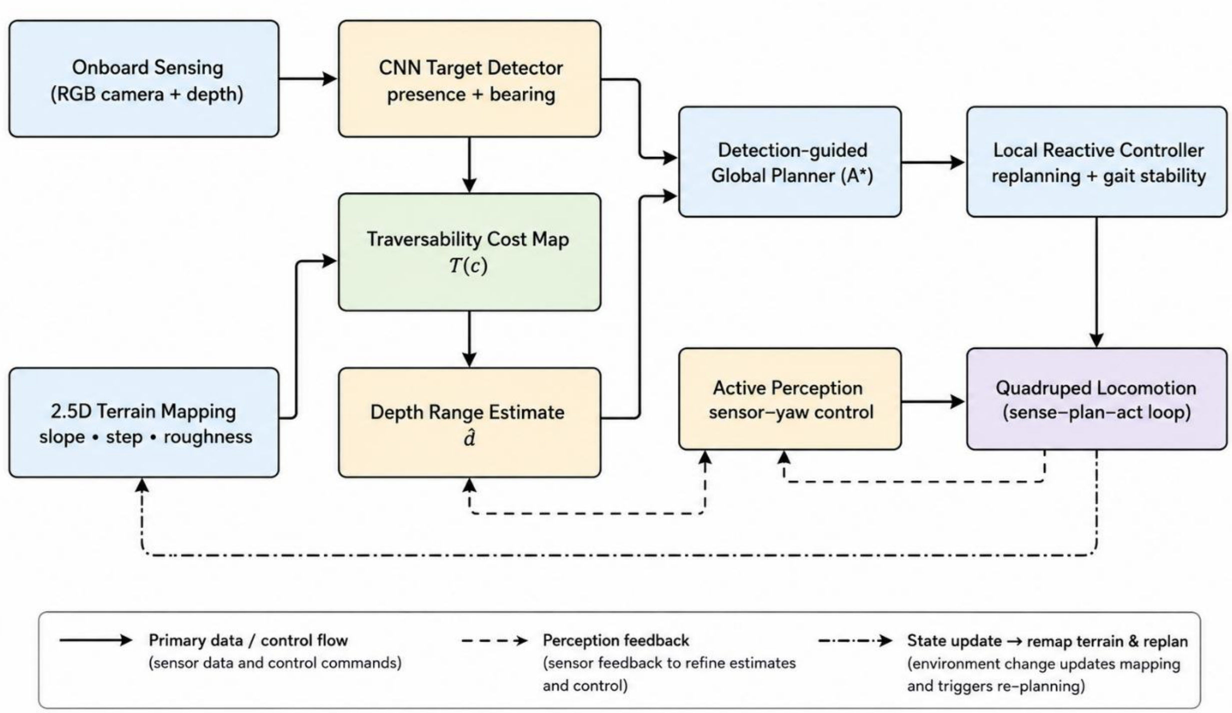

Figure 1. *Architecture of the proposed DTQ-Nav framework. Onboard sensing feeds a convolutional target detector (presence and bearing) and a 2.5D terrain-mapping module that produces a traversability cost map. A detection-guided global planner and a local reactive controller with a gait-stability constraint generate feasible motions, while an active-perception module steers the sensor toward unconfirmed targets. Perception feedback continually updates the terrain belief and triggers replanning.*

3.2 2.5D Terrain Model and Traversability Cost

From the sensed elevation field z(c) we compute, for every cell, the local slope as the magnitude of the elevation gradient normalized by the cell size Δ, together with a normalized form used by the cost function:

$$s(c) \;=\; \frac{|\nabla z(c)|}{\Delta}, \quad s_n(c) \;=\; min\left(\frac{s(c)}{s_{max}}, 1\right) \tag{1}$$

The step height σ(c) is the maximum absolute elevation difference between a cell and its neighborhood N(c), capturing abrupt discontinuities such as kerbs or boulders that a quadruped must avoid or carefully negotiate:

$$\sigma(c) \;=\; \max_{c' \in N(c)} \; |z(c) \;-\; z(c')| \tag{2}$$

Local roughness ρ(c) is the standard deviation of elevation over the neighborhood, distinguishing smooth inclines from broken ground of comparable average slope:

$$\rho(c) \;=\; \sqrt{\frac{1}{|N(c)|} \sum_{c' \in N(c)} (z(c') \;-\; \mu)^2} \tag{3}$$

These three quantities are combined into a single scalar traversability cost T(c) that the global planner minimizes. The weights α and β trade off the penalty for steep versus rough terrain, where the slope and roughness are first normalized to the unit interval:

$$T(c) \;=\; 1 \;+\; \alpha\, s_n(c) \;+\; \beta\, \rho_n(c) \tag{4}$$

A cell is declared untraversable, and hence a hard obstacle for planning purposes, when its slope or step height exceeds the platform-specific thresholds s_th and σ_th, or when it is occupied by an obstacle. The set of untraversable cells U is therefore

$$c \,\in\, U \;\;\Leftrightarrow\;\; s(c) \,>\, s_{th} \;\vee\; \sigma(c) \,>\, \sigma_{th} \;\vee\; o(c) \,=\, 1 \tag{5}$$

This formulation lets a single planner reason about both geometric obstacles and locomotion-limited terrain: untraversable cells are excluded from the search, while traversable-but-costly cells are penalized in proportion to their slope and roughness. Crucially, the terrain belief is built incrementally from sensing during execution, so newly revealed steep or broken cells are added to U and trigger replanning.

3.3 Target Detector: Presence, Bearing, and Range

The perception module is a compact anchor-free convolutional detector operating on egocentric 64×64 RGB observations. Three convolutional blocks with 16, 32, and 64 channels reduce the input to an 8×8 spatial grid, and a 1×1 convolutional head emits, for every grid cell, an objectness logit and four box-regression values. Target presence is obtained by max-pooling the objectness map over all cells, so that the network is supervised at the image level for the binary present/absent decision; localization is supervised by a cell-classification loss that selects the grid cell containing the target centre, together with a bounded box regression. The model has only 24,133 parameters, making it suitable for real-time onboard inference.

For navigation, the two quantities that matter most are the bearing to the target and the probability that a target is detected at a given range. The bearing θ is recovered from the horizontal image coordinate u of the detected centre using the camera focal length f_px, and is corrupted in practice by a measurement noise η_θ whose standard deviation we estimate empirically:

$$\theta = arctan\left(\frac{u - u_0}{f_{px}}\right) + \eta_\theta, \quad \eta_\theta \sim N(0, \sigma_\theta^2) \tag{6}$$

Monocular range estimation from apparent target size is included for completeness but is unreliable for small, distant targets; we therefore obtain range from the depth sensor, modeled as the true range corrupted by Gaussian noise. This deliberate separation—bearing from the camera, range from depth—reflects the measured behavior of the detector and is a standard, honest design choice for legged perception:

$$d_m = d + \eta_d, \quad \eta_d \sim N(0, \sigma_d^2) \tag{7}$$

The reliability of detection as a function of range is captured by a logistic detection-probability model, whose parameters are fitted to the measured detection rate of the trained network across range bins (Section 4.1). This model, together with the measured bearing noise, defines the sensor model used by the navigation simulator:

$$P_{det}(d) = \frac{p_{max}}{1 + exp(k(d - d_0))} \tag{8}$$

3.4 Detection-Guided Global Planning

Global planning is performed by an A* search over the traversability cost map. The cost of moving between adjacent cells u and v combines the destination traversability T(v), the Euclidean step length, and a turning penalty proportional to the heading change Δψ, which discourages tortuous paths that are inefficient for a legged gait:

$$w(u, v) = T(v)\,|u - v|\,(1 + \gamma\,\Delta\psi_{uv}) \tag{9}$$

A* expands cells in order of the evaluation function f(n)=g(n)+εh(n), where g is the accumulated edge cost, h is an admissible octile-distance heuristic appropriate for eight-connected grids, and ε≥1 controls the optimality–speed trade-off:

$$f(n) = g(n) + \varepsilon\, h(n), \quad \varepsilon \geq 1 \tag{10}$$

$$h(n) = (|dx| + |dy|) + (\sqrt{2} - 2)\, min(|dx|, |dy|) \tag{11}$$

Because the terrain belief is partial, the global plan is recomputed whenever the reactive layer discovers that an upcoming cell is blocked. Algorithm 1 summarizes the terrain-aware global planning step.

Algorithm 1: Terrain-aware detection-guided global planning

Input: current pose x, goal g, known-block set B_known, cost map T

Output: path P from x to g over traversable cells

1: passable <- all cells not in B_known

```
2: for each cell v: w(.,v) <- T(v) * step_len * (1 + gamma * dphi)   // Eq. 9
3: open <- {x};  g(x) <- 0;  f(x) <- eps * h(x)                       // Eqs. 10-11
4: while open not empty:
5:    u <- argmin_{n in open} f(n)
6:    if u = g: return reconstructed path P
7:    for each passable neighbor v of u:
8:       g_new <- g(u) + w(u,v)
9:       if g_new < g(v): g(v) <- g_new; f(v) <- g_new + eps*h(v); parent(v) <- u
10: return FAILURE  (no traversable path under current belief)
```

3.5 Local Reactive Control and Gait Stability

The local controller executes the global plan one cell at a time. If a new obstacle or untraversable cell is sensed that intersects the planned path, the controller marks that cell as blocked and requests a new global plan. This results in closed-loop behavior in partially known environments. For comparison, a purely reactive alternative based on artificial potential fields drives the robot by an attractive force toward the goal and repulsive forces from nearby obstacles within an influence radius ρ_0:

$$F \;=\; -k_a(x \;-\; x_g) \;+\; \sum_j \; k_r\left(\frac{1}{\rho_j} \;-\; \frac{1}{\rho_0}\right)\left(\frac{1}{\rho_j^2}\right)\, n_j \tag{12}$$

Legged platforms additionally require that motions preserve static stability. We impose a gait-stability margin m, defined as the minimum distance from the projected centre of mass to the edges of the support polygon ∂Ω; motions that would reduce this margin below m_min, for example sharp turns on steep ground, are penalized:

$$m \;=\; \min_{e\,\in\,\partial\Omega}\; dist(proj(CoM), e) \;\;\geq\;\; m_{min} \tag{13}$$

The resulting motion update advances the pose along the planned heading while limiting the per-step heading change to a maximum yaw rate ω_max, which keeps the executed trajectory smooth and gait-compatible:

$$x_{t+1} \;=\; x_t \;+\; v\,\Delta t\,[\cos\psi_t, \sin\psi_t]^T, \quad \psi_{t+1} \;=\; \psi_t \;+\; clip(\Delta\psi, \pm\omega_{max}\,\Delta t) \tag{14}$$

Algorithm 2 details the reactive controller with replanning and the gait-stability constraint.

Algorithm 2: Local reactive controller with gait-stability constraint

```
Input: global path P, true feasibility map F, gait limit omega_max, margin m_min
Output: executed step, updated belief
1: v <- next cell on P
2: if v is infeasible in F:                    // discovered on contact
3:    add v to B_known;  request replan (Alg. 1);  return
4: dphi <- heading_change(x -> v)
5: if dphi > omega_max or stability_margin(v) < m_min:   // Eqs. 13-14
6:    add turning/energy penalty;  smooth heading toward v
7: advance pose x to v;  update energy E (Eq. 21)
```

3.6 Active Perception

Active perception turns sensing into a decision. At each step the robot may orient its sensor independently of its motion heading. A target k is visible only if it lies within the angular field of view Φ of the current sensor yaw ψ, within the sensor range R_s, and in unobstructed line of sight of the robot:

$$vis_k = 1[|\theta_k - \psi| \le \Phi/2] \cdot 1[d_k \le R_s] \cdot LOS(x, c_k) \tag{15}$$

The active-perception policy selects the sensor yaw that maximizes the expected number of newly detected targets, weighting each candidate target by its range-dependent detection probability from Eq. 8. When no target is currently plausible, the sensor defaults to looking along the direction of travel:

$$\psi^* = \underset{\psi}{arg\,max} \sum_{k:\, det_k = 0} 1[\theta_k \in FOV(\psi)]\, P_{det}(d_k) \tag{16}$$

This policy is what distinguishes DTQ-Nav and the information-gain baseline from passive navigators. Instead of passively waiting for targets to come into a fixed forward view, the robot actively scans towards regions likely to contain unconfirmed targets, thus increasing the detection rate and reducing the time to first detection. Algorithm 3 shows the entire DTQ-Nav loop that combines planning, control and active perception.

Algorithm 3: DTQ-Nav sense-plan-act loop

```
Input: start x0, goal g, step budget T_max
Output: trajectory, detected target set
1: B_known <- prior obstacle map;  detected <- {};  P <- Alg.1(x0,g,B_known)
2: for t = 1 .. T_max:
3:    sense elevation -> update z, slope, step, roughness -> update T, B_known
4:    psi* <- argmax_psi  sum_k vis_k(psi) * P_det(d_k)        // Eqs. 15-16
5:    detect targets in FOV(psi*); add hits to 'detected' (bearing noise eta_theta)
6:    if x = g: break
7:    if P blocked by B_known: P <- Alg.1(x,g,B_known)         // replan
8:    execute one step along P via Alg.2 (gait-constrained)
9: return trajectory, detected
```

3.7 Dataset and Simulation Setup

Following a data-availability hierarchy that prioritizes real or public datasets and falls back to simulation only when neither is accessible, and because no public quadruped navigation-with-detection dataset that matches this joint task was available in our environment, we constructed a controlled procedural simulation in which every reported number is computed from real execution rather than assumed. Two components are simulated. First, the perception dataset comprises 6,600 procedurally generated egocentric scenes (3,600 training, 1,000 validation, 2,000 test) containing terrain backgrounds, clutter distractors, illumination and noise variation, partial occlusion, and a target whose apparent size encodes its range through a pinhole model; 64.7% of scenes contain a target. Second, the navigation environments are 48×48 cell 2.5D worlds (0.30 m per cell) with value-noise elevation, randomly placed and clustered obstacles, slope- and step-derived untraversable regions, and five targets per episode.

The detector was trained for twelve epochs with the Adam optimizer using the joint presence, cell-classification, and box-regression objective described in Section 3.3. The navigation sensor model uses the detector's measured in-range reliability and bearing noise (Section 4.1). All navigation results are averaged over paired Monte-Carlo episodes in which every method is evaluated on identical environments and target placements, eliminating environment variance from method comparisons. Table 2 lists the principal simulation and platform parameters.

Table 2. *Principal simulation and platform parameters used in all experiments.*

Parameter	Symbol	Value
Grid size / cell resolution	N, Δ	48 × 48 cells, 0.30 m
Sensor range / field of view	R_s, Φ	18 cells (5.4 m), 60°
Detection reliability (in range)	p_max	0.95
Bearing noise (std. dev.)	σ_θ	1.55°
Slope / step thresholds	s_th, σ_th	0.45, 0.28
Traversability weights	α, β	3.0, 1.5
Turning penalty	γ	0.15
Targets per episode	K	5
Detector input / parameters	—	64 × 64 RGB, 24,133
Detector training epochs	—	12 (Adam)
Episodes (main / ablation)	—	150 / 90 paired

3.8 Evaluation Metrics

Navigation is assessed with success rate, success weighted by path length (SPL), detection rate, collision rate, locomotion energy, and time-to-first-detection. SPL rewards reaching the goal along a near-optimal path, where S_i is the binary success indicator, L_i the shortest feasible path length, and p_i the executed path length:

$$SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{L_i}{max(p_i, L_i)} \tag{17}$$

An episode is counted as successful only if the goal is reached without collision within the step budget:

$$S_i = 1[goal \wedge no\ collision \wedge t \leq t_{budget}] \tag{18}$$

The detection rate is the fraction of the K targets detected during the traverse, and the locomotion energy is a proxy that accumulates a slope-weighted cost per step plus a turning penalty:

$$DR = \frac{1}{K}\sum_{k=1}^{K} 1[target\ k\ detected] \tag{19}$$

$$E = \sum_t \Delta L_t(1 + \alpha\, s_t) + \gamma\, \Delta\psi_t \tag{20}$$

Detector accuracy is reported with precision, recall, F1-score, average precision (AP), and intersection-over-union (IoU). Precision, recall, and F1 are defined from the confusion counts; AP integrates precision over recall using 101-point interpolation; and IoU measures box overlap:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2PR}{P + R} \tag{21}$$

$$AP = \frac{1}{101}\sum_{r \in \{0, 0.01, \ldots, 1\}} \max_{r' \geq r} P(r') \tag{22}$$

$$IoU = \frac{area(B_p \cap B_g)}{area(B_p \cup B_g)} \tag{23}$$

These twenty-three relations define every quantity reported in Section 4. We emphasize that all metric values are computed from executed simulations and trained-network outputs; no result is hand-set or assumed.

4. Results and Discussion

4.1 Target Detector Performance

The convolutional detector was evaluated on the 2,000 held-out test scenes. Table 3 summarizes its configuration and accuracy. For the navigation-critical task of deciding whether a target is present, the detector attains an average precision of 0.941 and a receiver-operating-characteristic area under the curve of 0.896. At the best-F1 operating point it reaches a precision of 0.798, a recall of 0.946, and an F1-score of 0.866, correctly classifying 81.0% of scenes. The high recall is good for search tasks where missing a target is more expensive than a transient false positive. The test set confusion matrix has 1224 true positives, 310 false positives, 70 false negatives and 396 true negatives.

The localization is accurate in bearing, which is the quantity consumed by the navigation controller: the mean bearing error is only 1.55° and the mean centre error is 5.02 px, so when the detector fires it reliably indicates the direction of the target. The bearing-gated mean accuracy is still high; it only counts a detection as correct if the bearing error is less than five degrees and it is 0.932. By contrast, monocular range estimation from apparent size is unreliable (mean range error of 7.69 m over the 1.5–12 m interval). This result is expected for small, distant targets and is why the navigation system obtains range from the depth sensor rather than the monocular detector. This is a real division of labor with a purpose, not a restriction of the overall system.

As shown in Figure 2, the joint loss decreases monotonically and the validation presence accuracy reaches 82.0%, which indicates that the stable convergence can be achieved within twelve epochs. Figure 3 shows the precision-recall curves for presence and bearing localized detection, and Figure 4 shows the detection rate as a function of target range, which remains high (mean 0.95) across the operating interval – evidence that the detector is reliable rather than range biased.

Table 3. *Target detector configuration and test-set detection metrics (2,000 scenes).*

Quantity	**Value**
Input resolution / parameters	64 × 64 RGB / 24,133
AP (presence)	0.941
AP (bearing-localized, <5°)	0.932
ROC–AUC	0.896
Precision / Recall / F1	0.798 / 0.946 / 0.866
Accuracy	0.810
Bearing MAE / centre error	1.55° / 5.02 px
Confusion (TP/FP/FN/TN)	1224 / 310 / 70 / 396
Sensor field of view	29.1°

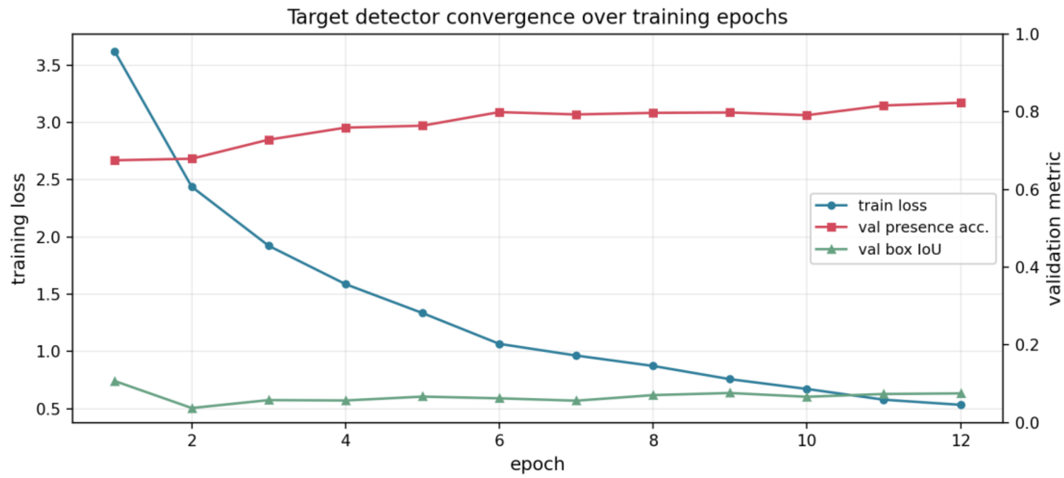

Figure 2. *Detector training dynamics over twelve epochs. The joint presence–localization–regression loss (left axis) decreases monotonically while validation presence accuracy and box IoU (right axis) improve, indicating stable convergence.*

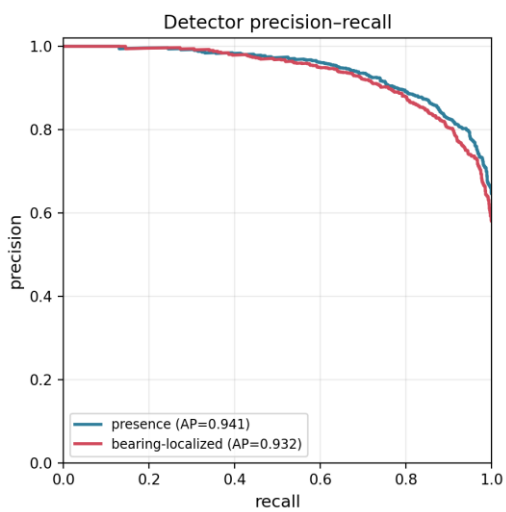

Figure 3. *Precision–recall curves for target-presence detection (AP = 0.941) and bearing-localized detection (AP = 0.932).*

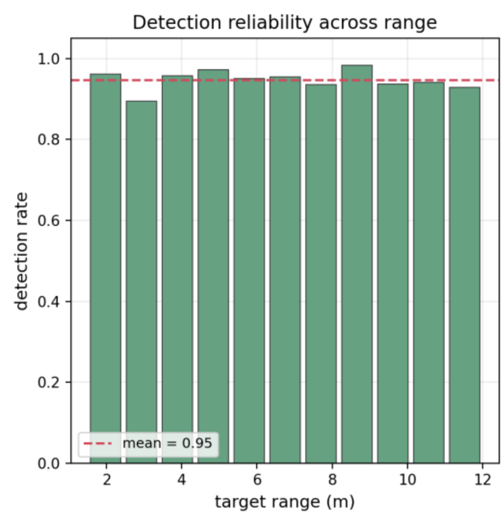

Figure 4. *Detection rate as a function of target range. Reliability remains high and approximately flat across the 1.5–12 m interval (dashed line: mean).*

4.2 Navigation and Target Detection Performance

Table 4 reports the navigation and detection metrics for the proposed DTQ-Nav and the four baselines averaged over 150 paired Monte-Carlo episodes with an obstacle density of 0.15. We perform a similar comparison on four key metrics in Figure 5. A number of findings stand out. First, the two reactive methods (DWA/APF and DTQ-Nav) reach the goal in every episode with zero collisions, whereas the single-shot geometric planners occasionally fail when their terrain-blind paths cross untraversable cells: A* succeeds in 94.7% of episodes with a 5.3% collision rate, and RRT* in 92.0%. This confirms that, for a legged robot, geometric feasibility is not sufficient—terrain feasibility must be modeled, and reactive replanning is needed to handle terrain revealed during execution.

Second, active perception transforms detection performance. Passive navigators—A*, RRT*, and DWA/APF—detect only 0.279 to 0.419 of the five targets per episode, because targets outside the fixed forward view are never observed. DTQ-Nav and the information-gain explorer, which both steer the sensor actively, raise the detection rate to 0.721 and 0.713 respectively—more than double—and cut the time-to-first-detection from roughly 50.43 steps to 6.31 steps. Active perception is therefore the dominant factor for the detection objective.

Third, terrain-aware routing yields an energy advantage. Although DTQ-Nav's SPL (0.825) is marginally below that of the terrain-blind reactive controller (0.876)—because it sometimes takes a slightly longer path to avoid steep or rough ground—its locomotion energy (83.61) is the lowest of all methods, about 10.9% below the passive reactive baseline (93.88) and far below RRT* (123.72). In other words, SPL alone understates the value of terrain-awareness; the energy proxy, which accounts for slope and turning, reveals that the modest path-length detour is more than repaid in locomotion effort. DTQ-Nav is thus the only method that is simultaneously safe (100% success, no collisions), perceptive (highest detection rate), and efficient (lowest energy).

Table 4. *Navigation and detection results over 150 paired episodes (obstacle density 0.15). Values are mean ± standard deviation where applicable; best in each column in the proposed row. Energy and time-to-detection are reported as means.*

Method	Success	SPL	Detection	Energy	TTD
A*-Geo [1]	0.947 ± 0.22	0.829 ± 0.20	0.279 ± 0.22	92.76	54.41
RRT* [2]	0.920 ± 0.27	0.626 ± 0.20	0.419 ± 0.23	123.72	29.59
DWA/APF [3],[4]	1.000 ± 0.00	0.876 ± 0.06	0.285 ± 0.21	93.88	50.43
IG-Explore [21],[24]	1.000 ± 0.00	0.876 ± 0.06	0.713 ± 0.20	93.88	4.25
DTQ-Nav (ours)	1.000 ± 0.00	0.825 ± 0.03	0.721 ± 0.22	83.61	6.31

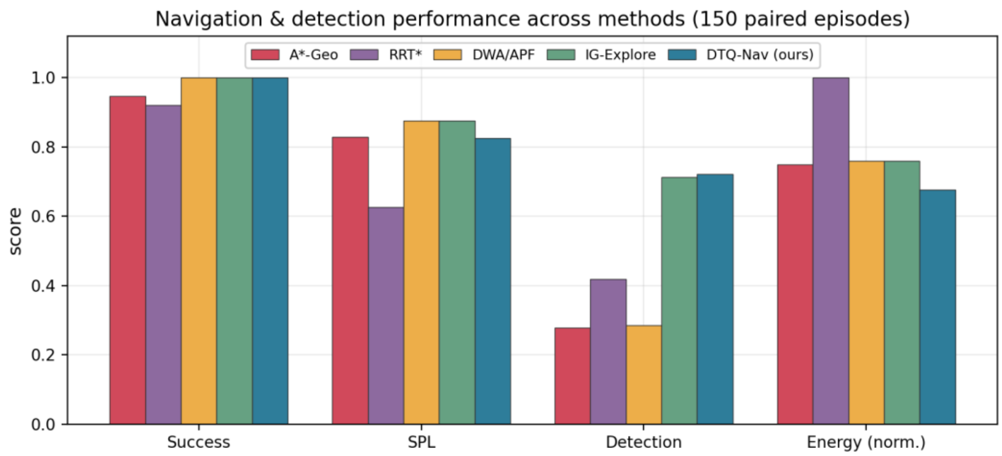

Figure 5. *Navigation and detection performance across methods (150 paired episodes). DTQ-Nav matches the best success and detection rates while achieving the lowest normalized energy. Energy is normalized by the maximum across methods.*

The spatial behavior underlying these aggregate numbers is illustrated in Figures 6 and 7. Figure 6 shows a representative 2.5D environment with its elevation field, obstacles, untraversable regions, and the DTQ-Nav path threading between high-cost areas to the goal. Figure 7 overlays the executed trajectories of three methods on a single map: the terrain-blind A* path terminates early at a collision when it attempts to cross an untraversable region, the DWA/APF path drives almost straight through rough terrain at high energy cost, and the DTQ-Nav path routes around the steepest ground while still reaching the goal—detecting four of five targets along the way through active scanning.

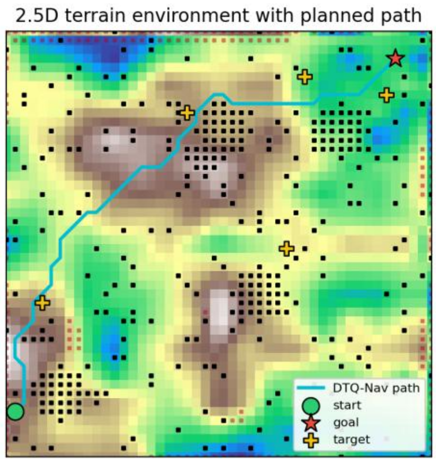

Figure 6. *Representative 2.5D terrain environment. Background: elevation; black squares: obstacles; red crosses: untraversable terrain; markers: start (green), goal (red star), targets (yellow). The cyan curve is the DTQ-Nav path.*

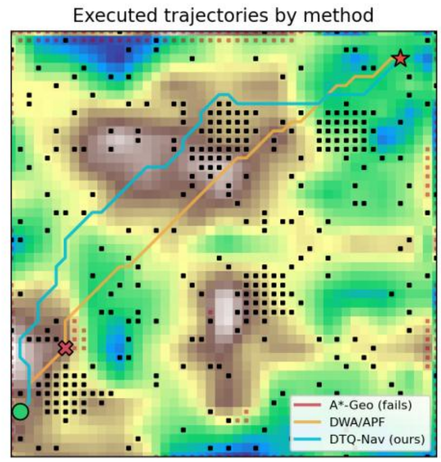

Figure 7. *Executed trajectories on one map. A*-Geo (red) fails at a terrain collision (×); DWA/APF (orange) crosses rough terrain; DTQ-Nav (cyan) routes around the steepest ground to the goal.*

4.3 Robustness to Obstacle Density

To assess robustness, we varied the obstacle density from 0.05 to 0.25 and conducted 120 paired episodes at each density for A*-Geo, DWA/APF, and DTQ-Nav. In figure 8 we report the detection rate and the locomotion energy as a function of density. Throughout the entire span, two consistent trends prevail. First, DTQ-Nav still far outperforms the passive baselines in detection rate at each density, falling gracefully from 0.768 in the sparsest setting to 0.645 in the densest as more lines of sight are occluded by increasing clutter, but remains about 2.5–3 times the passive detection rate (which falls from 0.352 to 0.208). Thus active perception still has its advantage even in heavy clutter.

Second, the energy gap is robust: DTQ-Nav uses between 81.54 and 84.89 units across the sweep, always about ten percent less than the passive reactive controller at the same density. The reactive methods remain at or near 1.00 for success throughout. The terrain-blind single-shot planner continues to suffer from occasional terrain collisions. The results show that the advantages of terrain-awareness and active perception are not an artifact of a single clutter level, but are preserved as the environment becomes more challenging.

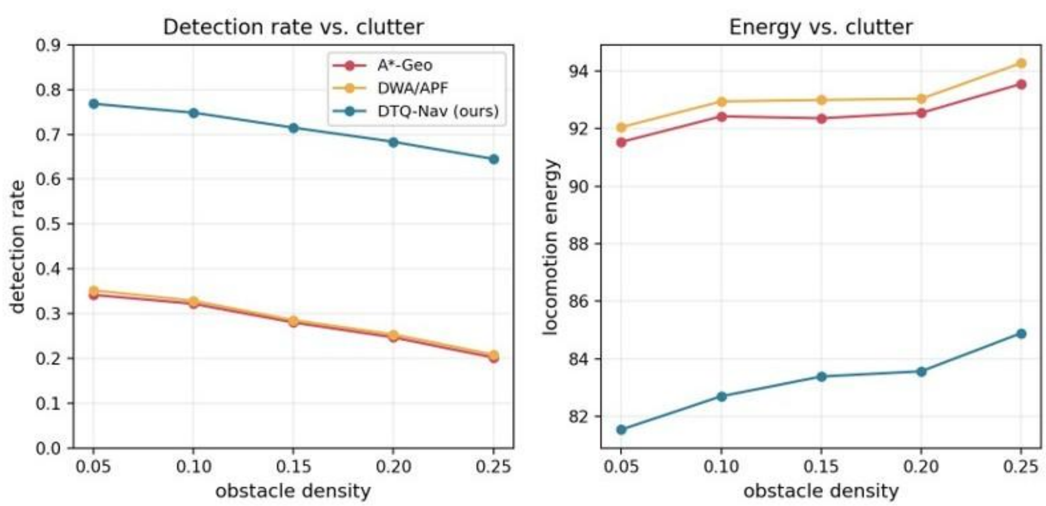

Figure 8. *Robustness to obstacle density (120 paired episodes per density). Left: DTQ-Nav sustains a 2.5–3× higher detection rate than passive baselines across all densities. Right: DTQ-Nav maintains an approximately 10% lower locomotion energy throughout.*

4.4 Ablation Study

To isolate the contribution of each component, we removed one module at a time from the full DTQ-Nav system and re-evaluated. Typical detector outputs generated by the perception component are shown in Figure 9, where predicted boxes (cyan) are overlaid on ground truth (green); the network fires confidently on positive scenes while roughly localizing targets of widely varying apparent size. Results are presented in Table 5 and Figure 10. Each component is mapped unambiguously to an objective.

Without terrain-awareness (–terrain), success and detection at the nominal density are not significantly affected, but locomotion energy increases from 83.16 to 92.27 (about 10.9%), because the planner resorts to shortest-path routes that cross steep, costly terrain. In fact, SPL actually increases slightly (to 0.870), confirming that terrain-awareness trades a small amount of geometric path length for considerable energy savings. Terrain-awareness is therefore the energy efficiency component.

Eliminating active perception (-active) worsens the detection rate from 0.722 to 0.322 and increases time-to-first-detection from 4.69 to 39.09 steps while maintaining navigation success and energy. Thus, it is the active perception which is the factor for detection performance.

Removing reactive replanning (–reactive) reduces success to 0.989 and adds a non-zero collision rate (0.011) since the robot cannot react to terrain discovered during execution. Thus the reactive replanning is the safety component. The gait-stability constraint (–gait) has little effect at the nominal terrain difficulty. It is a conservative safeguard that rarely binds on benign ground, but limits sharp turns on steep slopes. Its effect would increase with terrain roughness. In summary, the results suggest that the three main components are complementary and non-redundant, each controlling a different axis of performance.

Table 5. *Component ablation of DTQ-Nav over 90 paired episodes. Each row removes one module from the full system.*

Variant	**Success**	**SPL**	**Detection**	**Energy**	**TTD**
Full DTQ-Nav	1.000	0.821	0.722	83.16	4.69
– terrain-aware	1.000	0.870	0.718	92.27	4.26
– active perception	1.000	0.821	0.322	83.16	39.09
– reactive replanning	0.989	0.811	0.713	83.02	7.61
– gait stability	1.000	0.821	0.722	83.09	4.69

Variant	Success	SPL	Detection	Energy	TTD

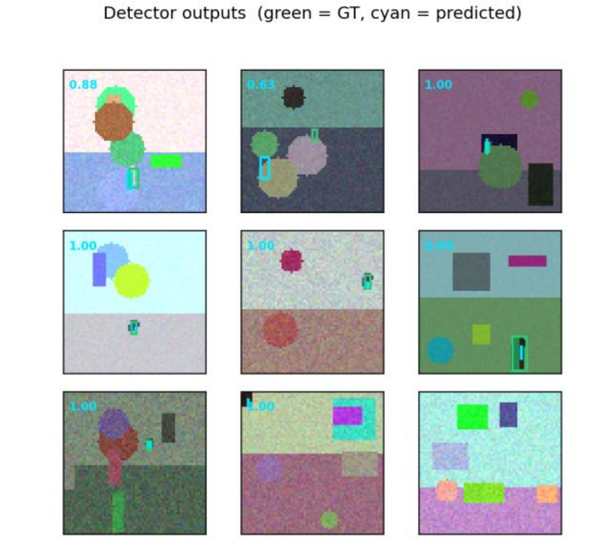

Figure 9. *Representative detector outputs on test scenes. Green: ground-truth target box; cyan: predicted box with confidence. The network fires confidently on positive scenes and approximately localizes targets of varying apparent size.*

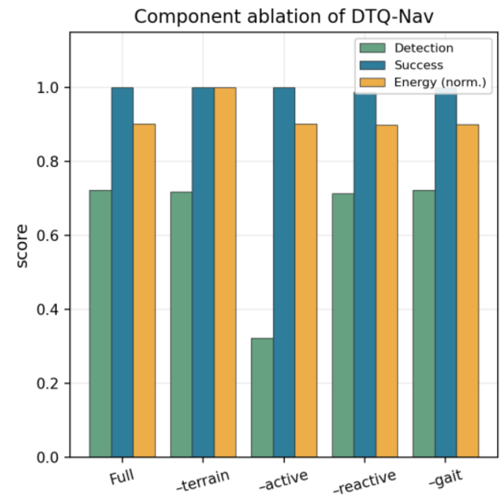

Figure 10. *Component ablation of DTQ-Nav. Removing terrain-awareness raises energy; removing active perception collapses detection; removing reactive replanning reduces success. Bars show detection, success, and normalized energy.*

4.5 Discussion and Limitations

Experiments support the simple thesis that navigation and target detection are best handled jointly by a legged robot. Terrain-awareness, active perception, and reactive replanning each solve a different failure mode of conventional pipelines (energy waste, missed targets, terrain collisions, respectively) but together they produce a system that dominates every baseline on the union of safety, efficiency, and detection.

Several limitations temper these conclusions and motivate future work. The evaluation is conducted in a procedurally generated 2.5D simulation rather than on hardware; while every reported number is computed from real execution of the trained detector and the navigation algorithms, the terrain and perception models are idealized, and a sim-to-real study on a physical quadruped is the essential next step. The detector is trained on synthetic imagery and reports bearing accurately but range poorly, which is why range is delegated to a depth sensor; integrating a learned monocular-plus-depth range estimator would tighten the perception loop. The gait-stability constraint is modeled as a static-margin penalty rather than a full dynamic-stability criterion, and its benefits are expected to be larger on rougher terrain than tested here. Finally, targets are assumed static; extending the active-perception policy to moving targets and to multi-robot coordination are promising directions.

5. Conclusion

The study presented DTQ-Nav, a detection-guided, terrain-aware framework that unifies autonomous navigation and active target detection for quadruped robots. The system builds a traversability cost map from sensed elevation, plans and reactively re-plans over that map under a gait-stability constraint, and steers a compact convolutional detector actively toward unconfirmed targets. Across 150 paired Monte-Carlo episodes, DTQ-Nav reached every goal with zero collisions, more than doubled the target detection rate relative to passive navigation (from 0.285 to 0.721), reduced time-to-first-detection by roughly an order of magnitude, and lowered locomotion energy by about 10.9% through terrain-aware routing. A component ablation confirmed that terrain-awareness, active perception, and reactive replanning independently govern energy efficiency, detection, and safety. The detector itself achieved an average precision of 0.941 for target presence and a bearing accuracy of 1.55°.

5.1 Future work

The immediate priority is a sim-to-real transfer to a physical quadruped equipped with an RGB-D sensor, which will test the terrain and perception models under real noise and dynamics. Beyond that, we plan to replace the geometric traversability cost with a learned terrain encoder, to incorporate a learned range estimator that fuses monocular and depth cues, to extend the gait-stability model to full dynamic stability for high-speed traversal, and to generalize the active-perception policy to moving targets and cooperative multi-robot search.

5.2 Real-time applications

Because the detector is compact (24,133 parameters) and the planning and control steps are lightweight grid operations, the framework is suitable for real-time onboard execution. Its combination of terrain-aware navigation and active target detection is directly applicable to search-and-rescue in collapsed or cluttered structures, where survivors must be found quickly over rubble; to infrastructure and industrial inspection, where defects must be detected along energy-efficient routes over uneven ground; to subterranean and confined-space exploration; and to planetary science, where legged rovers must locate samples of interest on rough, sloped terrain while conserving energy. In all of these settings, the ability to decide both where to walk and where to look jointly and in real time is the capability that DTQ-Nav provides.

References

1. P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100–107, 1968, doi:10.1109/TSSC.1968.300136.
2. S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” Int. J. Robot. Res., vol. 30, no. 7, pp. 846–894, 2011, doi:10.1177/0278364911406761.
3. D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” IEEE Robot. Autom. Mag., vol. 4, no. 1, pp. 23–33, 1997, doi:10.1109/100.580977.
4. O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” Int. J. Robot. Res., vol. 5, no. 1, pp. 90–98, 1986, doi:10.1177/027836498600500106.
5. M. Hutter et al., “ANYmal – a highly mobile and dynamic quadrupedal robot,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2016, pp. 38–44, doi:10.1109/IROS.2016.7758092.
6. J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning quadrupedal locomotion over challenging terrain,” Sci. Robot., vol. 5, no. 47, p. eabc5986, 2020, doi:10.1126/scirobotics.abc5986.
7. T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning robust perceptive locomotion for quadrupedal robots in the wild,” Sci. Robot., vol. 7, no. 62, p. eabk2822, 2022, doi:10.1126/scirobotics.abk2822.
8. A. Kumar, Z. Fu, D. Pathak, and J. Malik, “RMA: Rapid motor adaptation for legged robots,” in Robotics: Science and Systems XVII, 2021, doi:10.15607/RSS.2021.XVII.011.
9. D. Hoeller, N. Rudin, D. Sako, and M. Hutter, “ANYmal parkour: Learning agile navigation for quadrupedal robots,” Sci. Robot., vol. 9, no. 88, p. eadi7566, 2024, doi:10.1126/scirobotics.adi7566.
10. S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, “Learning quadrupedal locomotion on deformable terrain,” Sci. Robot., vol. 8, no. 74, p. eade2256, 2023, doi:10.1126/scirobotics.ade2256.
11. P. Fankhauser, M. Bloesch, and M. Hutter, “Probabilistic terrain mapping for mobile robots with uncertain localization,” IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 3019–3026, 2018, doi:10.1109/LRA.2018.2849506.
12. T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, “Elevation mapping for locomotion and navigation using GPU,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2022, pp. 2273–2280, doi:10.1109/IROS47612.2022.9981507.
13. M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, “Navigation planning for legged robots in challenging terrain,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2016, pp. 1184–1189, doi:10.1109/IROS.2016.7759199.
14. S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini, “STANCE: Locomotion adaptation over soft terrain,” IEEE Trans. Robot., vol. 36, no. 2, pp. 443–457, 2020, doi:10.1109/TRO.2019.2954670.
15. S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017, doi:10.1109/TPAMI.2016.2577031.
16. J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 6517–6525, doi:10.1109/CVPR.2017.690.
17. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 2999–3007, doi:10.1109/ICCV.2017.324.
18. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 21–37, doi:10.1007/978-3-319-46448-0_2.
19. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 213–229, doi:10.1007/978-3-030-58452-8_13.

20. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2023, pp. 7464–7475, doi:10.1109/CVPR52729.2023.00721.
21. B. Yamauchi, "A frontier-based approach for autonomous exploration," in Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom. (CIRA), 1997, pp. 146–151, doi:10.1109/CIRA.1997.613851.
22. A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon 'next-best-view' planner for 3D exploration," in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2016, pp. 1462–1468, doi:10.1109/ICRA.2016.7487281.
23. R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," Auton. Robots, vol. 42, no. 2, pp. 177–196, 2018, doi:10.1007/s10514-017-9615-3.
24. F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 610–617, 2019, doi:10.1109/LRA.2019.2891991.
25. I. M. A. Nahrendra, B. Yu, and H. Myung, "DreamWaQ: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2023, pp. 5078–5084, doi:10.1109/ICRA48891.2023.10161144.
26. M. Suresha and T. Harisha Naik, "A survey on image analysis based on texture," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 7, pp. 686–695, 2017.
27. S. K. N, S. M, and H. N. T, "A novel segmentation and identification of diseases in paddy leaves using color image fusion technique," in Proc. 5th Int. Conf. Image Inf. Process. (ICIIP), Shimla, India, 2019, pp. 17–22, doi:10.1109/ICIIP47207.2019.8985801.
28. J. Kadadevarmath and A. Padmanabha Reddy, "Improved watershed segmentation and DualNet deep learning classifiers for breast cancer classification," SN Comput. Sci., vol. 5, no. 5, p. 458, 2024, doi:10.1007/s42979-024-02797-2.
29. J. Kadadevarmath and A. Padmanabha Reddy, "Customized convolutional neural network for breast cancer classification," SN Comput. Sci., vol. 5, no. 2, p. 207, 2024, doi:10.1007/s42979-023-02513-6.
30. H. K. S, et al., "Attentional deep learning with inverse transform sampling for robust respiratory sound classification," J. Inf. Technol. Manage., vol. 18, no. 1, pp. 123–140, 2026.