Multi-Phase Vision-Based Navigation and Inspection for Legged Robots with Online Goal Refinement and Vision-Only Halting

Debasish Mishra[1] Xiaogang Wang[1] Jia Hao Toh[2] Wangsheng Liu[2] Yash Shah[1]

[1]AWS [2]Certis

{misdebas, xiaogawa, syash}@amazon.com {EvinJH_TOH, WangSheng_LIU}@certisgroup.com

Abstract

Vision-based navigation is important for autonomous legged robots, enabling semantic target search and inspection without dense maps or specialized sensors. However, existing goal-conditioned navigation methods often rely on pre-curated goal images, lack reliable stopping mechanisms, and are sensitive to detector noise and gait-induced camera perturbations. To address these limitations, we propose a multi-phase vision-based navigation and inspection framework for quadrupeds using only monocular RGB input. First, we introduce online detection-driven goal refinement, which extracts target crops from the robot's live camera stream and progressively updates the goal representation as the robot approaches the target, reducing dependence on pre-collected goal images and improving robustness to distant or low-quality detections. Second, we design a phase-factorized policy structure that decomposes the task into long-range approach and close-range orbital inspection, using two specialized checkpoints of the same navigation backbone to better handle the different control requirements of each phase. Third, we develop a vision-only halting and target-lock mechanism that uses the bbox-to-frame-area ratio with temporal smoothing to trigger stable depth-free stopping, while an image-based visual servoing loop keeps the target centered despite detector jitter and gait-induced camera motion. Experiments on a 12-DOF quadruped across on-axis, 60°, and 90° outdoor scenarios show superior performance, achieving 85–95% end-to-end success with consistent ∼1.0–1.2 m stopping distance. Ablations further validate the contribution of each component, and the framework generalizes to both ViNT and NoMaD backbones without architectural modification.

1. Introduction

Vision-based navigation is a core capability for autonomous robots operating in real-world inspection environments. Compared with map-based or depth/LiDAR-based autonomy stacks, monocular vision offers a lightweight and deployable sensing interface that can support semantic target search, goal-directed approach, and close-range inspection using only onboard cameras. This is particularly important for legged robots, which are expected to operate in unstructured sites such as grass, gravel, curbs, and industrial facilities where wheeled platforms may fail. However, practical inspection is not a single-step navigation problem: the robot must first detect a semantic target, approach it from potentially off-axis viewpoints, decide when it has reached a safe and consistent standoff distance, and then execute a structured inspection behavior such as orbiting while keeping the target in view. These requirements make vision-based inspection inherently multi-phase and impose different control demands at different stages of the task.

Recent visual navigation methods have made substantial progress toward learning goal-conditioned policies from image observations. Classical image-based visual servoing regulates image-space feature errors for closed-loop target alignment [1], while learning-based navigation methods such as GNM [7], ViNG [6], ViNT [8], and NoMaD [10] learn reusable visual navigation priors for waypoint prediction and cross-embodiment deployment. NoMaD in particular builds on denoising diffusion probabilistic models (DDPM) [4] for multi-modal action generation, an approach pioneered for visuomotor policy learning by Diffusion Policy [2]. These methods provide strong foundations for visual goal reaching, but several limitations remain when deploying them for real quadruped inspection. First, many goal-conditioned policies assume that a clean goal image is available before deployment, which is often unrealistic when the target is discovered online by the robot itself. Second, a static goal crop can become unreliable as the robot approaches the target, especially when detections are intermittent at long range or disturbed by gait-induced camera pitch and roll. Third, existing navigation backbones mainly predict waypoints and do not directly solve the arrival decision: without depth sensing or a metric map, deciding when to stop at a consistent physical standoff remains non-trivial.

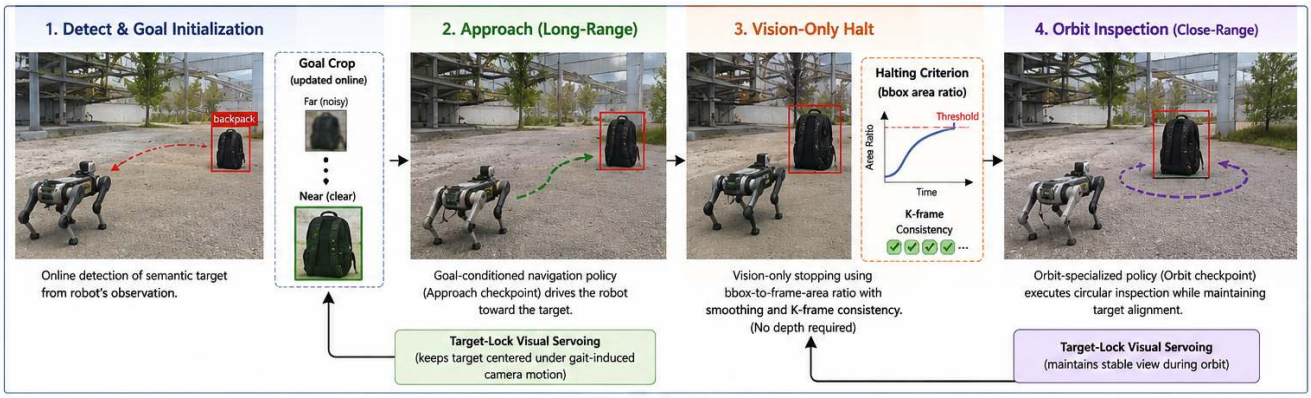

Figure 1. Overview of the proposed vision-based multi-phase navigation and inspection framework. Given only monocular RGB observations, the quadruped first detects the semantic target and initializes an online goal crop from its own camera stream. During long-range approach, the goal representation is progressively refined as clearer target views become available, while target-lock visual servoing keeps the target centered under gait-induced camera motion. When the bbox-to-frame-area ratio remains within a calibrated range for K consecutive frames, the system triggers a vision-only halt without depth sensing. The robot then switches to an orbit-specialized policy to perform close-range circular inspection while maintaining stable visual alignment with the target.

Finally, a single monolithic policy is difficult to optimize for both long-range approach and close-range orbital inspection, because these phases require different motion patterns and stability properties.

Motivated by these limitations, we propose a deployed multi-phase vision-based navigation and inspection framework for quadrupeds using only monocular RGB input as shown in Fig. 1. Our system first performs online detection-driven goal refinement, where target crops are extracted from the robot's live camera stream and progressively updated as the target becomes clearer during approach. The refined goal is then used by a goal-conditioned approach policy, while a lightweight target-lock visual servoing loop continuously corrects yaw to keep the detected target centered despite detector jitter and gait-induced camera motion. Once the target reaches a stable image-space scale, a vision-only halting rule based on bbox-to-frame-area ratio, exponential smoothing, and consecutive-frame consistency triggers the transition from approach to inspection. After this transition, a separate orbit-specialized checkpoint of the same navigation backbone executes circular inspection while sharing the same target-lock mechanism for visual stabilization. In summary, our main contributions are:

- **Online goal refinement:** we eliminate the need for pre-curated goal repositories by updating the goal representation from the robot's own observation stream.
- **Phase-factorized navigation and inspection:** we decompose the task into long-range approach and close-range orbiting using two specialized checkpoints of the same backbone.
- **Vision-only halting and target locking:** we combine bbox-area-based stopping with image-space visual servoing to achieve stable depth-free arrival and target alignment.
- **Real-world quadruped evaluation:** we validate the system on a 12-DOF quadruped across on-axis, 60°, and 90° outdoor target-placement scenarios with component-level ablations.
- **Backbone-agnostic integration:** we demonstrate that the proposed deployment pattern can be integrated with both ViNT and NoMaD without architectural modification.

2. Related Works

Goal-conditioned navigation backbones such as ViNT [8] and NoMaD [10] encode observation–goal image pairs and predict control outputs, enabling strong cross-platform transfer but leaving open system-level questions around online goal formation, stopping consistency, and post-arrival behaviors. GNM [7] established the general navigation model paradigm for cross-embodiment deployment, and ViNG [6] studies open-world navigation with visual goals but relies on curated goal sets. Classical image-based visual servoing (IBVS) [1] regulates image-feature error to maintain target alignment.

Classical autonomy stacks for legged robots typically combine a lidar- or depth-based SLAM layer (e.g., LIO-SAM [9], FAST-LIO [11]) with a traversability-aware local planner running on top of the vendor locomotion controller (e.g., Boston Dynamics Spot Autonomy, ANYbotics ANYmal autonomy, or ROS `move_base_flex` adapted to legged platforms). These stacks provide strong geometric guarantees but depend on (i) depth or lidar sensing and (ii) a metric map or an online-built occupancy/elevation grid — neither of which is available in our monocular, mapless, semantic-target deployment. Purely image-based alternatives such as visual servoing on bounding-box geometry can

track a target once it is in view but cannot produce a coherent long-range approach when the detector is intermittent, the heading is off-axis, or the camera pitches during gait. We use this observation to motivate the learned approach policy and include a bbox-only servo heuristic as a non-learning reference point in our ablations.

Our work targets *end-to-end multi-phase inspection for legged robots using only monocular vision*: we contribute (i) online goal refinement from the robot's stream, (ii) phase-factorized approach/inspection policies, (iii) a vision-only halting criterion with temporal-consistency guarantees, and (iv) systematic system-level deployment and evaluation. The system architecture is shown in Figure 2.

3. Method

Our framework decomposes vision-based quadruped inspection into four coupled runtime modules: online goal refinement, goal-conditioned approach, vision-only halting, and circular inspection. Given only a monocular RGB stream, the robot first detects the semantic target and constructs a goal crop from its own observations. As the robot approaches, this goal representation is updated when higher-quality target views become available. The refined goal is then used by a goal-conditioned navigation policy to drive long-range approach, while a lightweight target-lock visual servoing module continuously corrects yaw to keep the target centered. When the target reaches a stable image-space scale, a bbox-area-based halting rule triggers the phase transition. The system then switches from the approach checkpoint to an orbit-specialized checkpoint of the same backbone to perform close-range inspection. This phase-factorized design allows the same visual navigation backbone to handle different control regimes without changing the architecture.

3.1. Perception and Online Goal Refinement

At time t, the robot receives an RGB observation $I_t \in \mathbb{R}^{H\times W\times 3}$. A target detector $D(\cdot)$ predicts a bounding box $b_t = (u_t, v_t, w_t, h_t)$ and confidence c_t. Detections with $c_t < c_{\text{det}}$ are discarded to remove low-confidence false positives. For each valid detection, we compute the normalized bbox-to-frame area ratio

$$r_t = \frac{w_t h_t}{HW}, \tag{1}$$

and extract a target crop $G_t = \mathcal{C}(I_t, b_t)$.

Instead of using a fixed pre-collected goal image, we maintain a cached goal $\hat{G}_t$ that is updated online from the robot's own camera stream. A cache update is accepted only when the current detection is sufficiently confident, approximately centered, and provides a larger target view than the previous cached crop:

$$\hat{G}_t = \begin{cases} G_t, & c_t \geq c_{\text{cache}} \wedge |\psi_{\text{err},t}| \leq \psi_{\max} \wedge r_t \geq r_{\text{cache}} + \tau_r, \\ \hat{G}_{t-1}, & \text{otherwise.} \end{cases} \tag{2}$$

Here $c_{\text{cache}} > c_{\text{det}}$ is a stricter cache-update threshold, r_{cache} is the area ratio of the cached target crop, and $\psi_{\text{err},t}$ is the approximate yaw error estimated from the bbox center. In practice, we require the update condition to hold for M consecutive frames before replacing the cache. This avoids goal thrashing caused by gait-induced camera pitch/roll oscillations and transient detector jitter. If the detector temporarily drops out, the most recent cached goal remains valid for a time-to-live window T_{ttl}, allowing the navigation policy to continue using a stable goal representation.

3.2. Goal-Conditioned Approach with Target-Lock Servoing

During the approach phase, the navigation backbone receives a short temporal context of observations I_t^{ctx} and the refined goal crop $\hat{G}_t$, and predicts a local waypoint and yaw command:

$$\hat{a}_t = (\Delta x_t, \Delta y_t, \Delta\psi_t^{\text{nav}}) = \pi_\theta\Big(I_t^{\text{ctx}}, \hat{G}_t\Big). \tag{3}$$

The waypoint components drive the robot toward the target, while $\Delta\psi_t^{\text{nav}}$ provides the policy-predicted heading change.

However, on a legged robot, the body-mounted camera undergoes periodic pitch and roll motion during walking, which causes the detected target to drift in the image even when the global navigation direction is correct. To stabilize target alignment, we add a lightweight image-based target-lock servoing module. Given bbox $b_t = (u_t, v_t, w_t, h_t)$, the bbox center and image center are

$$p_t = \begin{bmatrix} u_t + w_t/2 \\ v_t + h_t/2 \end{bmatrix}, \qquad p^* = \begin{bmatrix} W/2 \\ H/2 \end{bmatrix}. \tag{4}$$

We use only the horizontal offset for yaw correction:

$$e_t = \frac{p_{t,x} - p_x^*}{W} = \frac{u_t + w_t/2 - W/2}{W}. \tag{5}$$

Under a pinhole camera model, this normalized offset is mapped to an angular error by

$$\alpha_t \approx e_t \theta_{\text{FOV}}, \tag{6}$$

which is the small-angle approximation of $\arctan(2e_t \tan(\theta_{\text{FOV}}/2))$.

To suppress small detector fluctuations, we apply a deadband:

$$\delta(\alpha_t) = \begin{cases} 0, & |\alpha_t| \leq \epsilon, \\ \alpha_t, & |\alpha_t| > \epsilon. \end{cases} \tag{7}$$

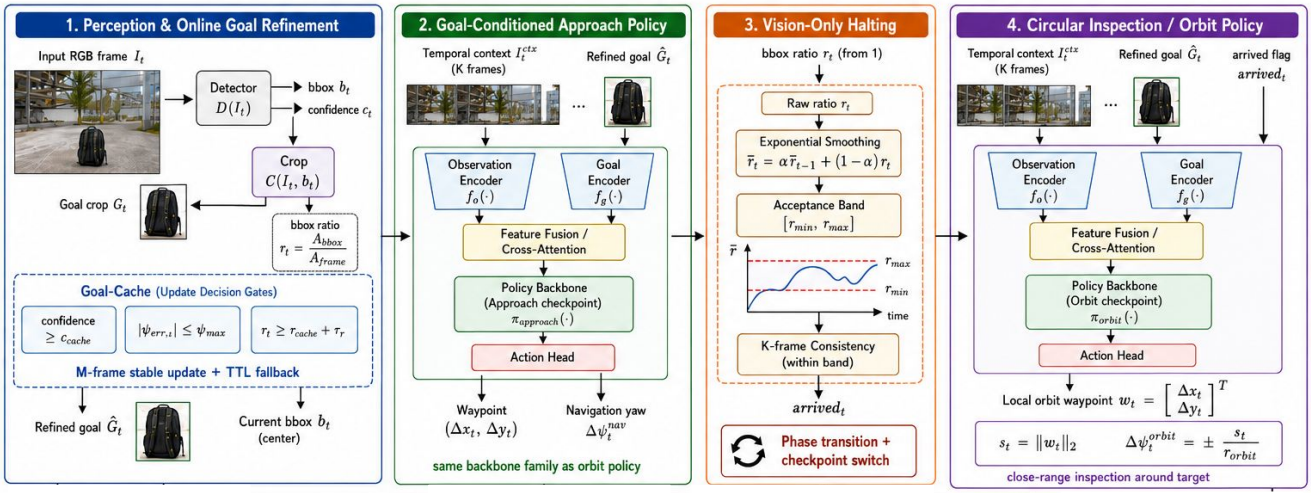

Figure 2. Method overview of the proposed multi-phase vision-based navigation and inspection framework. Given a monocular RGB frame, the system first detects the semantic target and performs online goal refinement by updating a cached goal crop only when the detection satisfies confidence, alignment, and area-improvement conditions. The refined goal is then used by a goal-conditioned approach policy, where observation and goal encoders are fused through a shared navigation backbone to predict waypoints and navigation yaw. A vision-only halting module monitors the bbox-to-frame-area ratio, applies exponential smoothing, and triggers the phase transition only when the smoothed ratio remains within a calibrated acceptance band for K consecutive frames. After arrival, the system switches to an orbit-specialized checkpoint of the same backbone to generate local circular-inspection waypoints around the target.

The target-lock yaw-rate correction is then computed as

$$\omega_t^{\text{lock}} = \text{clip}\left(-k_\omega \delta(\alpha_t), -\omega_{\max}, \omega_{\max}\right), \tag{8}$$

where k_ω is the proportional gain and $\omega_{\max}$ is the maximum allowed yaw rate. The final approach yaw command is obtained by fusing the policy output with the target-lock correction:

$$\omega_t^{\text{approach}} = \frac{\Delta\psi_t^{\text{nav}}}{\Delta t} + \omega_t^{\text{lock}}, \tag{9}$$

where $\Delta t = 1/f_{\text{ctrl}}$ is the control period. Thus, the learned policy provides long-range navigation, while target-lock servoing provides closed-loop image-space stabilization.

3.3. Vision-Only Halting with Smoothing and Hysteresis

The same bbox-area ratio r_t used for goal refinement is also used to decide when the robot has reached the desired standoff distance. Since raw bbox size can fluctuate due to detector noise and legged gait motion, we first apply exponential smoothing:

$$\bar{r}_t = \alpha\bar{r}_{t-1} + (1-\alpha)r_t. \tag{10}$$

The robot is considered to have arrived only when the smoothed ratio stays inside a calibrated acceptance band $[r_{\min}, r_{\max}]$ for K consecutive frames:

$$\text{arrived}_t = \mathbf{1}\left[\forall i \in \{0, \ldots, K-1\},\ r_{\min} \leq \bar{r}_{t-i} \leq r_{\max}\right]. \tag{11}$$

This creates a temporal hysteresis effect: a single noisy frame cannot trigger arrival, and the system switches phase only when the target remains at a stable image scale. In our deployment, the band $[r_{\min}, r_{\max}]$ is calibrated so that the target bbox occupies approximately 20% of the image frame, corresponding to a physical standoff of roughly 1.0–1.2 m for the target object and onboard camera. Once $\text{arrived}_t = 1$, the active checkpoint is switched from the approach policy to the orbit policy.

3.4. Circular Inspection Motion

After halting, the robot enters the close-range inspection phase. This phase uses an orbit-specialized checkpoint of the same navigation backbone, which is trained to generate local waypoints suitable for circular motion around the target. Let the predicted local waypoint be

$$\mathbf{w}_t = \begin{bmatrix} \Delta x_t \\ \Delta y_t \end{bmatrix}. \tag{12}$$

We approximate the local arc length by

$$s_t = \|\mathbf{w}_t\|_2. \tag{13}$$

Given the desired orbit radius r_{orbit}, the corresponding yaw increment around the target is

$$\Delta\psi_t^{\text{orbit}} = \pm\frac{s_t}{r_{\text{orbit}}}, \tag{14}$$

where the sign determines clockwise or counter-clockwise inspection. The final orbit yaw-rate command combines the geometric orbit term with the same target-lock correction used during approach:

$$\omega_t^{\text{orbit}} = \frac{\Delta\psi_t^{\text{orbit}}}{\Delta t} + \omega_t^{\text{lock}}. \tag{15}$$

The orbit term drives the robot along a circular trajectory, while ω_t^{lock} keeps the target centered in the camera view during inspection. As a result, the system maintains visual focus on the target even under gait-induced camera perturbations, enabling stable close-range inspection using only monocular RGB input.

4. Dataset

We collect a real-world navigation and inspection dataset using a 12-DOF quadruped equipped with a forward-facing RGB camera and onboard odometry. The dataset contains 120 trajectories recorded in deployment-relevant environments, including both indoor scenes and outdoor daylight conditions. The use of a legged platform is important for the target application, where inspection sites may contain grass, gravel, curbs, and other terrain transitions that are difficult for wheeled robots [5]. All data are therefore collected directly on the same quadruped platform used for deployment, ensuring that the learned policies are exposed to realistic camera motion, gait-induced viewpoint changes, and terrain-dependent motion patterns.

The trajectories are collected by manually teleoperating the quadruped with a joystick to approach a semantic target, i.e., a backpack, and, when applicable, perform a circular inspection motion around it. Each recorded trajectory is stored as a ROS bag and contains synchronized RGB and odometry streams. The RGB stream provides the visual observations I_t used by the detector, online goal refinement module, and goal-conditioned navigation policy. The odometry stream provides the robot pose (x_t, y_t, ψ_t), which is used for trajectory analysis and evaluation metrics such as approach progress and stopping consistency. After preprocessing, each trajectory contains 120–2000 frames, covering either the full approach behavior or the combined approach and inspection behavior.

Data composition. The dataset covers two types of environments and two motion regimes. In terms of environment, it contains 75 indoor trajectories and 45 outdoor daylight trajectories collected under varying illumination and weather conditions. In terms of motion, it contains 84 approach-only trajectories and 36 approach-plus-inspection trajectories. The approach-only trajectories capture goal-directed navigation toward the target, while the approach-plus-inspection trajectories include both target approach and subsequent circular/orbital inspection behavior.

Sensor streams. Each trajectory contains two synchronized sensor streams:

- **RGB images:** compressed image frames from the forward-facing onboard camera, used as the only perception input for detection, goal refinement, and policy inference.
- **Odometry:** per-timestep robot pose (x_t, y_t, ψ_t), used only for supervision and evaluation rather than as an input to the deployed policy.

Table 1. Summary of the collected real-world quadruped navigation dataset.

Category	Type	# of trajectories
Environment	Indoor	75
	Outdoor daylight	45
Motion regime	Approach-only	84
	Approach + inspection	36
Total	–	120

Relevance to deployment. The dataset is designed to reflect the practical deployment setting of vision-based inspection. Since the robot observes the target from different distances and viewpoints during each trajectory, the data naturally supports online goal refinement from far, noisy detections to closer, clearer target views. The inclusion of both approach-only and approach-plus-inspection trajectories also supports the proposed phase-factorized training strategy, where one checkpoint specializes in long-range approach and another checkpoint specializes in close-range orbital inspection.

5. Experiments

We evaluate on 20 held-out test trajectories (80 train / 20 validation) and through live quadruped deployment. Our experiments compare: **(E1)** pretrained ViNT/NoMaD checkpoints with no fine-tuning; **(E2)** a single fine-tuned policy trained on mixed approach and orbit segments; and **(E3)** two specialized fine-tuned policies, π_B (approach) and π_C (orbit), switched at the phase boundary. Live testing uses the two-phase (E3) policy combined with online goal refinement, bbox-ratio halting, and target-lock visual servoing.

5.1. Scenarios for live testing

Outdoor daylight tests used a backpack placed 7 m from the quadruped starting position. Each scenario was run for $n = 20$ trials with varied placements of the quadruped and backpack:

- **Scenario 1 (on-axis):** target in the line of sight along the quadruped's heading direction.
- **Scenario 2 (off-axis 60°):** target within a 60° angular field-of-view centered on the heading.
- **Scenario 3 (off-axis 90°):** target within a 90° angular field-of-view centered on the heading.
- **Scenario 4 (halting ablation):** on-axis placement at 7 m, identical to Scenario 1, used to isolate the halting crite-

rion; the target-lock loop is active but evaluation focuses on whether the robot stops at the desired ∼1.0–1.2 m standoff without overshoot or premature halt.

- **Scenario 5 (target-lock ablation):** 90° off-axis curved-path approach, identical geometry to Scenario 3, used to isolate the contribution of target-lock visual servoing by toggling the servo loop on/off while holding the approach policy fixed.

Scenarios 4 and 5 are reported in the ablation studies in Section 5.3.1.

5.2. Metrics

Technical metrics — training objective (action loss). We train by optimizing the action mean-squared error (MSE). Let $\hat{A} \in \mathbb{R}^{B\times T\times D}$ denote the predicted actions and $A \in \mathbb{R}^{B\times T\times D}$ the ground-truth action labels. The per-element unreduced MSE is

$$\ell_{\text{mse}}(\hat{A}, A) = (\hat{A} - A) \odot (\hat{A} - A), \tag{16}$$

and the action loss is obtained by reducing (averaging) over non-batch dimensions:

$$\mathcal{L}_{\text{action}} = \text{Reduce}\Big(\ell_{\text{mse}}(\hat{A}, A)\Big), \tag{17}$$

where $\text{Reduce}(\cdot)$ is a mean over the horizon/waypoint dimensions (optionally with masking) [8].

Navigation accuracy (cosine similarities). We evaluate navigational prediction accuracy using cosine similarity on (i) waypoint vectors and (ii) orientation vectors. With $\hat{A}_t^{\text{wp}} = \hat{A}_{t,:,1:2}$ and $A_t^{\text{wp}} = A_{t,:,1:2}$ as the 2D waypoint components, and $\hat{A}_t^{\text{ori}}$, A_t^{ori} as the orientation components, cosine similarity is $\cos(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}/(\|\mathbf{x}\|\|\mathbf{y}\|)$ and the two reported metrics are

$$\text{CosSim}_{\text{wp}} = \text{Reduce}\Big(\cos(\hat{A}^{\text{wp}}, A^{\text{wp}})\Big), \tag{18}$$

$$\text{CosSim}_{\text{ori}} = \text{Reduce}\Big(\cos(\hat{A}^{\text{ori}}, A^{\text{ori}})\Big). \tag{19}$$

Business metrics (live deployment).

- **Phase-1 success %** (detection + approach + stop): fraction of trials where the quadruped navigates to the target and halts for a pre-defined time.
- **Phase-2 success %** (orbit transition + circular motion): fraction of trials where the quadruped completes one full 360° loop around the target and halts.
- **End-to-end success %**: Phase-1 AND Phase-2 succeed in the same run.

For live-testing metrics we report point estimates with 95% Wilson confidence intervals over $n = 20$ trials per cell.

5.3. Quantitative results

Figure 3 shows representative video frames from a navigation trial: starting from the initial position, the robot autonomously navigates toward the bag, transitions from the approach policy to the orbiting policy upon arrival, and completes a full circulation around the target. Table 2 reports held-out trajectory performance using ViNT, while Table 3 reports the corresponding results using NoMaD. Table 4 further reports live-deployment performance with trial counts and 95% Wilson confidence intervals. Component-level ablation studies are reported in Tables 5 and 6.

Table 2. Performance of ViNT on test trajectories

Method	$\mathcal{L}_{\text{action}}\downarrow$	$\text{CosSim}_{\text{wp}}\uparrow$	$\text{CosSim}_{\text{ori}}\uparrow$
Pretrained ViNT (no FT)	2.53	0.22	0.80
Single-policy ViNT (FT)	0.71	0.62	0.95
Two-policy ViNT (FT)	0.67	0.94	0.97

Table 3. Performance of NoMaD on test trajectories

Method	$\text{CosSim}_{\text{wp}}\uparrow$	$\text{CosSim}_{\text{ori}}\uparrow$
Pretrained NoMaD (no FT)	0.27	0.74
Single-policy NoMaD (FT)	0.68	0.94
Two-policy NoMaD (FT)	0.92	0.94

The results show that factorization provides a significant improvement over a single mixed policy. For ViNT, the pretrained checkpoint performs poorly on waypoint prediction, achieving only 0.22 waypoint cosine similarity and an action loss of 2.53. Fine-tuning a single policy improves the action loss to 0.71 and increases waypoint cosine similarity to 0.62, showing that the backbone can adapt to the collected quadruped trajectories. However, the proposed two-policy design further improves performance, reducing the action loss to 0.67 and increasing waypoint cosine similarity to 0.94. This corresponds to a 73.5% reduction in action loss and a 0.72 absolute improvement in waypoint cosine similarity compared with the pretrained ViNT checkpoint. Orientation prediction also improves from 0.80 to 0.97, indicating that the phase-factorized policy better captures both translational and heading control.

A similar trend is observed for NoMaD. After single-policy fine-tuning, waypoint cosine similarity increases to 0.68 and orientation cosine similarity improves to 0.94. The proposed two-policy NoMaD further improves waypoint cosine similarity to 0.92, giving a 0.65 absolute improvement over the pretrained checkpoint and a 0.24 absolute improvement over the single-policy fine-tuned model. The orientation metric remains high at 0.94, suggesting that the main benefit of phase factorization for NoMaD is an improved waypoint prediction rather than a heading prediction.

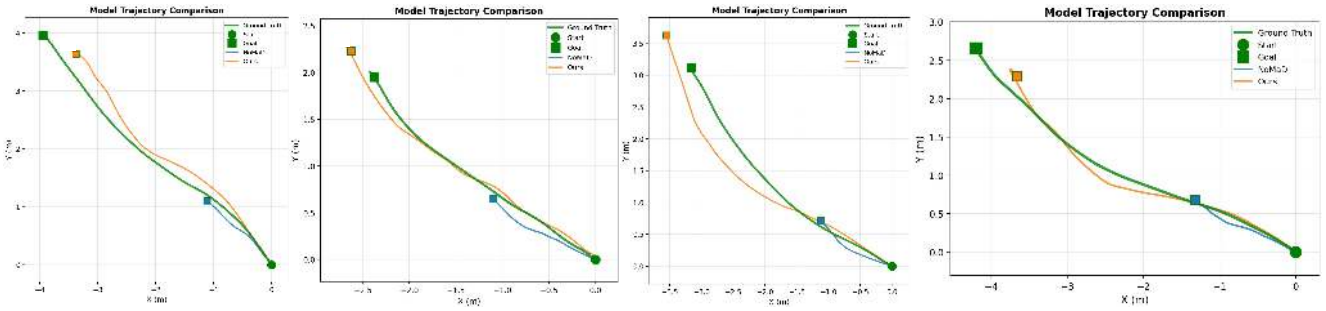

Figure 3. Qualitative comparison of predicted trajectories across four navigation scenarios. The ground truth path is shown in dark green, the NoMaD baseline in blue, and our method in orange.

Overall, the two-policy design consistently achieves the best or near-best performance across both backbones. Compared with single-policy fine-tuning, it improves waypoint cosine similarity from 0.62 to 0.94 for ViNT and from 0.68 to 0.92 for NoMaD. These gains support our motivation that approach and orbiting impose different control requirements: a single policy trained on mixed behaviors tends to average over these regimes, while a separate approach and orbit checkpoints allow the same backbone family to specialize to each phase. The consistency of the improvement across both ViNT and NoMaD also indicates that the proposed phase-factorized deployment is backbone-agnostic rather than specific to one navigation model.

Table 4 reports live-deployment results on the 12-DOF quadruped under three target-placement scenarios. In Scenario 1, where the target is placed on-axis with the robot heading, the pretrained ViNT and NoMaD checkpoints both fail to complete the full inspection task: although each achieves only 5% Phase-1 success, neither can transition to successful orbiting, resulting in 0% end-to-end success. In contrast, fine-tuned phase-factorized policies achieve substantially higher performance. ViNT (FT) obtains 95% Phase-1 success, 95% Phase-2 success, and 90% end-to-end success, while NoMaD (FT) reaches 95% success across all three metrics. This demonstrates that direct deployment of pretrained visual navigation backbones is insufficient for the quadruped inspection setting, whereas task-specific fine-tuning with the proposed multi-phase design enables reliable real-world execution.

Across the off-axis scenarios, the fine-tuned systems maintain strong performance despite increasing viewpoint difficulty. In Scenario 2 (60° off-axis), ViNT (FT) achieves 90% Phase-1 and end-to-end success, while NoMaD (FT) achieves 95% for both. In Scenario 3 (90° off-axis), the task becomes more challenging because the robot must recover target alignment from a larger heading offset. Under this setting, ViNT (FT) obtains 85% Phase-1 success and 80% end-to-end success, while NoMaD (FT) obtains 85% Phase-1 success and 85% end-to-end success. Importantly, Phase-2 success remains consistently high at 95% across all fine-tuned models and scenarios, indicating that once the robot reaches the correct standoff and triggers the phase transition, the orbit-specialized checkpoint can reliably complete the circular inspection behavior.

The end-to-end mission time also remains stable across scenarios. For fine-tuned methods, the average E2E time ranges from 1.8 to 2.5 minutes, with NoMaD (FT) being slightly faster than ViNT (FT) in all reported scenarios. The increase in time from on-axis to 90° off-axis placements reflects the additional correction required for larger heading offsets, but the system still completes the inspection within a practical deployment window. Since each cell contains $n = 20$ trials, the Wilson confidence intervals are relatively wide; for example, a 95% success rate corresponds to a confidence interval of $[76.4, 99.1]$. Therefore, we do not claim statistically significant differences among the fine-tuned success rates. Instead, the main quantitative finding is the clear gap between pretrained and fine-tuned deployment, together with the robustness of the proposed multi-phase system across increasingly difficult target-placement angles.

5.3.1. Ablation studies

Table 5 evaluates the effect of different vision-only stopping criteria during live testing. The proposed goal occupation criterion, which uses the bbox-to-frame-area ratio of the currently observed target, achieves the highest Phase-1 success rate of 90%. This outperforms feature-similarity-based stopping by 25 percentage points and bbox-ratio stopping based on a repository goal by 16 percentage points. The improvement indicates that directly measuring the target's image occupancy provides a more reliable proxy for physical standoff distance than comparing visual features or relying on a fixed pre-collected goal image. In Scenario 4, where the robot starts on-axis 7 m from the target, this criterion enables the quadruped to approach the backpack and stop consistently within the desired $\sim$1.0–1.2 m standoff range.

Table 6 further validates the importance of the target-lock visual servoing module. In the 90° off-axis curved-

Table 4. Live-testing results with trial counts and 95% Wilson confidence intervals. $n = 20$ trials per (Scenario, Method) cell; percentages are shown with CIs in brackets. E2E = end-to-end

Scenario	Method	n	Phase-1 % [95% CI] ↑	Phase-2 % [95% CI] ↑	E2E % [95% CI] ↑	E2E time (s) ↓
1	Pretrained ViNT	20	5 [0.9, 23.6]	0 [0.0, 16.1]	0 [0.0, 16.1]	3.0
	Pretrained NoMaD	20	5 [0.9, 23.6]	0 [0.0, 16.1]	0 [0.0, 16.1]	3.0
	ViNT (FT)	20	95 [76.4, 99.1]	95 [76.4, 99.1]	90 [69.9, 97.2]	2.0
	NoMaD (FT)	20	95 [76.4, 99.1]	95 [76.4, 99.1]	95 [76.4, 99.1]	1.8
2	ViNT (FT)	20	90 [69.9, 97.2]	95 [76.4, 99.1]	90 [69.9, 97.2]	2.2
	NoMaD (FT)	20	95 [76.4, 99.1]	95 [76.4, 99.1]	95 [76.4, 99.1]	2.0
3	ViNT (FT)	20	85 [64.0, 94.8]	95 [76.4, 99.1]	80 [58.4, 91.9]	2.5
	NoMaD (FT)	20	85 [64.0, 94.8]	95 [76.4, 99.1]	85 [64.0, 94.8]	2.3

Table 5. Stopping criteria ablation study during live testing

Stopping criterion	Phase-1 % ↑
(A) Feature similarity	65
(B) Bbox ratio (repository goal)	74
(C) Goal occupation (bbox/frame area)	90

Table 6. Target-lock visual servoing ablation during live testing

Visual servoing enabled	Phase-1 % ↑
(A) Yes	95
(B) No	15

path setting, enabling visual servoing yields 95% Phase-1 success, whereas removing it reduces success to only 15%. This 80-percentage-point drop shows that the learned navigation policy alone is not sufficient to maintain target alignment under large heading offsets and gait-induced camera perturbations. Without target-lock correction, the robot often loses the target from the camera view or moves toward an incorrect heading; the few successful trials correspond mainly to cases where the target remains close to the robot's initial line of sight. These results confirm that both proposed components are necessary: bbox-area-based halting provides stable depth-free arrival, while visual servoing enables robust approach under off-axis target placements.

Non-learning reference baseline. As a sanity check against a purely classical alternative, we implemented a bbox-only PID heuristic on the same legged platform: forward body velocity proportional to $(r_{\text{target}} - \bar{r}_t)$ and yaw rate proportional to the normalized bbox-center offset e_t, with no learned policy and no goal caching. This baseline succeeded in 5/10 line-of-sight trials (Scenario 1) but 0/10 in Scenario 3 (90° off-axis), because it cannot plan a turn through a heading gap: the target leaves the FOV as soon as the quadruped begins yawing, and the controller loses its only feedback signal. Classical map-based legged-autonomy stacks (depth/lidar SLAM + traversability planner) were not evaluated as they require sensors and a metric map that are explicitly out of scope for the customer's monocular, mapless deployment envelope.

5.4. Qualitative results

Figure 3 presents a qualitative comparison of trajectory predictions across four representative navigation scenarios with varying path lengths and curvatures. In all cases, our method (orange) produces trajectories that are visually closer to the ground truth (dark green) than the NoMaD baseline (blue). Specifically, our method better captures the smooth curvature of the reference path, maintaining tighter geometric alignment throughout the entire trajectory rather than cutting corners or deviating laterally in high-curvature segments. This advantage is most pronounced in scenarios with longer distances and sharper turns (panels 1 and 3), where NoMaD exhibits noticeable lateral offsets and slight kinks in the mid-trajectory region, while our method consistently follows the ground truth arc. In gentler, shorter-range scenarios (panels 2 and 4), both methods reach the goal successfully, but our method still demonstrates better path fidelity and approach angle. More qualitative results with generated trajectories comparisons are shown in Appendix.

6. Conclusion

We presented a deployed multi-phase inspection system for quadrupeds combining online goal refinement, a phase-factorized deployment of two specialized policy checkpoints on a 12-DOF quadruped, a vision-only halting rule, and a target-lock servo loop that runs across both phases to absorb gait-induced camera perturbations. Across three outdoor daylight approach scenarios with 20 trials each ($n = 60$ total), fine-tuned phase-factorized systems achieved 85–95% Phase-1, 95% Phase-2, and 80–95% end-to-end success at a consistent ∼1.0–1.2 m standoff (Table 4). Live ablations confirm that the occupation-ratio halting rule and target-lock module are each individually necessary for reliable non-line-of-sight approaches (Tables 5 and 6), and the pipeline integrates with both ViNT and NoMaD backbones without modification. Our evaluation is scoped to outdoor daylight conditions, backpack-class targets, and largely uncluttered approach paths. Extending validated performance to cluttered environments, multi-target scenes, broader object categories, and adaptive halting thresholds that generalize across target sizes are the primary directions for ongoing work, along with open-vocabulary detection and multi-altitude inspection for search-and-rescue, security, and industrial applications.

References

[1] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006. 1, 2

[2] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 1

[3] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 10

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020. 1

[5] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62), 2022. 5

[6] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Chelsea Finn, and Sergey Levine. Ving: Learning open-world navigation with visual goals, 2020. 1, 2

[7] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 1, 2

[8] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. In *Proceedings of The 7th Conference on Robot Learning (CoRL)*, pages 711–733, 2023. 1, 2, 6, 10

[9] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142, 2020. 2

[10] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration, 2023. 1, 2

[11] Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021. 2

Appendix

A. Qualitative results

Figure 4 shows representative video frames from a navigation trial: starting from the initial position, the robot autonomously navigates toward the bag, transitions from the approach policy to the orbiting policy upon arrival, and completes a full circulation around the target.

Another two trajectories generated from the models and compares them with the ground-truth trajectory are shown in Figs. 5 and 6, which demonstrates that our model can achieve more accurate and superior performances on trajectory generation.

B. Stopping criteria comparison

This section presents a comparative analysis of the stopping criteria tested. Each plot shows the stopping-criterion metric (y-axis) as a function of the distance from the approach start position to the target bag (x-axis), where distance is computed from odometry data. Figure 7 shows the criterion based on similarity between the observation frame and the goal frame selected from the repository, Figure 8 shows the ratio of the bbox area of the goal in the observation frame to the bbox area of the goal in the frame selected from the repository, and Figure 9 shows the criterion based on the bbox area of the goal in the observation frame divided by the area of the whole observation frame.

C. Live testing

This section shows the path followed by the quadruped during live testing using the finetuned model and the best stopping criterion (criterion 3). The smooth execution of this composite navigation task—combining goal-directed approach and orbital inspection—validates the practical applicability of our phase-factorized framework.

D. Implementation details

Hardware platform.
- 12-DOF quadruped robot with forward-facing monocular RGB camera.
- Camera resolution: 1920×1080 pixels.
- Frame rate: 5 Hz.
- Onboard compute: NVIDIA Jetson-class edge compute.

Object detection.
- Architecture: Faster RCNN [3].
- Training data: frames containing backpacks at distances up to 7 m.
- Detector admission threshold: $c_{\text{det}} = 0.5$.
- Inference frequency: 5 Hz (synchronized with camera).

Policy training.
- Base architecture: ViNT [8].
- Optimizer: Adam.
- Learning rate: 1.5×10^{-4}.
- Batch size: 64.
- Context type: temporal (stacked frames).
- Context size: 5 frames.
- Training epochs: 20.
- Data augmentation: random color jitter applied to input frames.
- Training hardware: AWS g5.12xlarge instance ($4\times$ NVIDIA A10G GPUs).
- Training time: $\sim$2 hours per policy.
- Fine-tuning: separate policies trained for Phase-1 (approach) and Phase-2 (inspection).

Goal refinement hyperparameters (see Sec. 3.1 in the main text).
- EMA smoothing coefficient: $\alpha = 0.40$.
- Area ratio update threshold: $\tau_r = 0.05$).
- Cache-update confidence gate: $c_{\text{cache}} = 0.7$ (distinct from detector admission threshold $c_{\text{det}} = 0.5$ above).

Halting criterion parameters.
- Acceptance band: $r_{\min} = 0.18$, $r_{\max} = 0.22$).
- Physical standoff corresponding to this band: $\sim$1.0–1.2 m for the target geometry used in training.
- Stability window: $K = 10$ consecutive frames.
- EMA coefficient: $\alpha = 0.40$.

Target-lock parameters
- Proportional yaw gain: $k_\omega = 0.5$).
- Deadband threshold: $\epsilon = 0.1$ rad $\approx 5.7°$).
- Maximum yaw rate: $\omega_{\max} = 0.12$ rad/s).
- Camera horizontal FOV: $\theta_{\text{FOV}} = 2.27$ rad $\approx 130°$).

Circular inspection parameters (see Appendix ??).
- Desired orbit radius: $r_{\text{orbit}} = 1$ m).
- Control frequency: $f_{\text{ctrl}} = 9$ Hz).
- Inspection duration: one complete 360° loop.

System integration.
- Framework: ROS Noetic.
- Deployment platform: onboard computation on the quadruped.

Figure 4. Video frames showing that the robot can navigate successfully from the starting point to the front of the bag and finish the circulation motion around the bag.

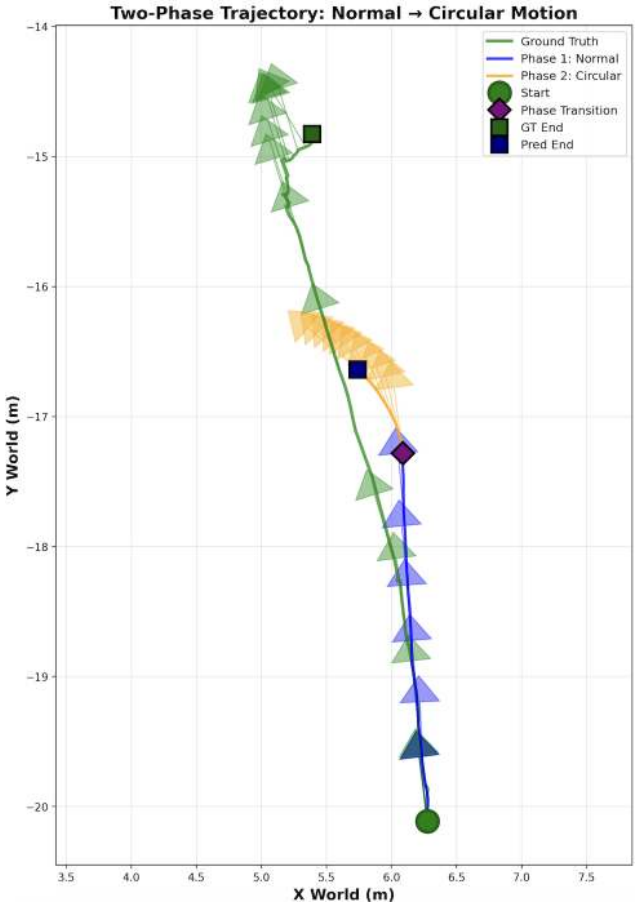

Figure 5. Predicted trajectory using pre-trained NoMaD checkpoint

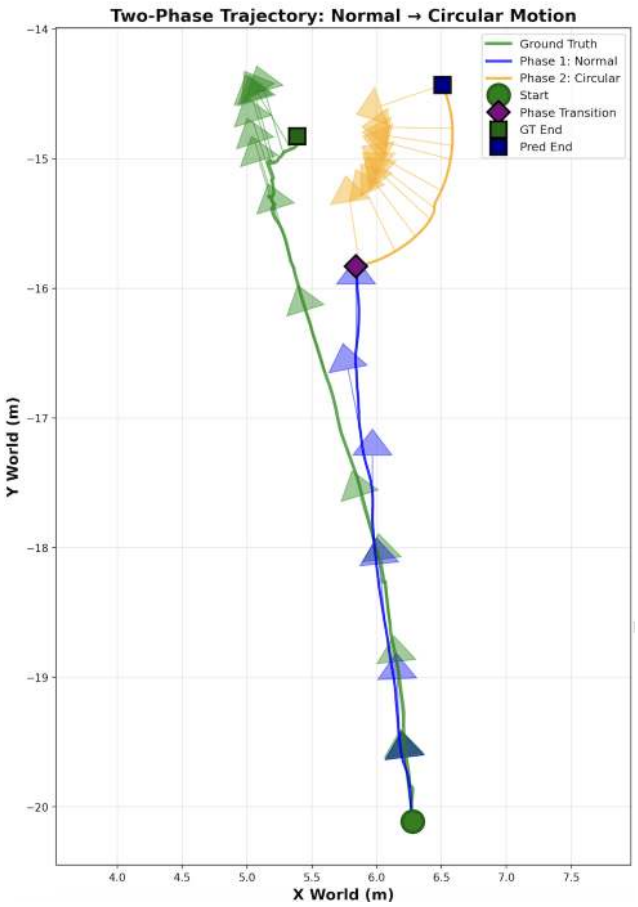

Figure 6. Predicted trajectory from our proposed approach

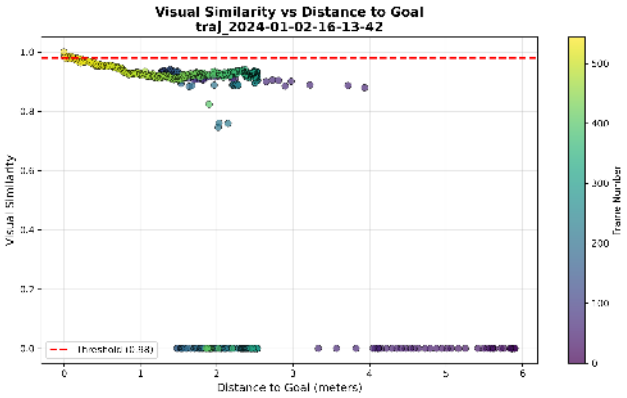

Figure 7. Stopping criterion 1: comparing the encoding of the observation frame and the goal frame from the goal repository

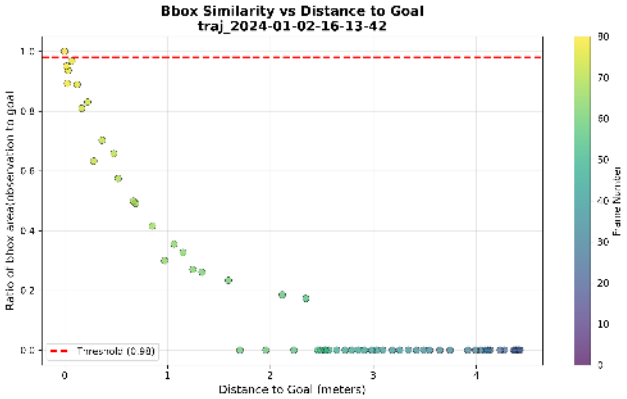

Figure 8. Stopping criterion 2: ratio of the area of the bbox of the goal in the observation frame and area of the bbox of the goal in the frame from the goal repository

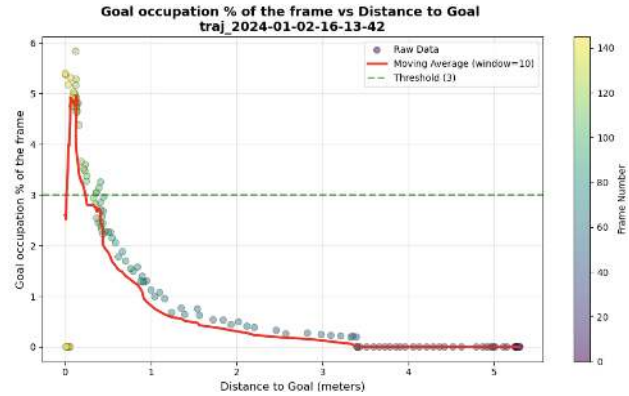

Figure 9. Stopping criterion 3: ratio of the area of the bbox of the goal in the observation frame and the area of the observation frame (occupation ratio). Does not use any goal repository