Matching Scenes, Not Objects: Failure Modes of Image-Goal LLM-VLA Navigation

Peter Bohm, Saimunur Rahman, Abdelwahed Khamis,
Sagun Man Singh Shrestha, Chris McCool, Peyman Moghadam
CSIRO Technology, Australia

***Abstract*—Image-goal navigation, where the destination is an image rather than a coordinate, is arguably the most open-world goal modality for vision-language-action (VLA) navigation, yet on hardware it fails at the last step: the robot reaches the visual goal but does not autonomously stop. We document this failure for an LLM-VLA navigation policy across two robots, and reproduce its core symptom on a second published system. The failure is not one bug but four modes: the predicted trajectory contracts at the goal yet the stop token never fires; the trajectory never contracts and the robot circles; the goal leaves the field of view and the stateless policy reverts to undirected exploration; and the policy drives to the most visually dominant region, a background tower or the open sky, rather than to the goal object. We trace all four to a single cause, goal under-specification: the policy is trained to match a frame sampled a fixed distance along its own future path, but is asked at deployment to reach a specific object, so it matches scenes rather than grounding objects. We discuss the directions a fix must take, from object-grounded and language-conditioned goal supervision to inference-side arrival signals, and validate none: this is a study of the failure, not a solution to it.**

I. Introduction

Image-goal navigation, where the destination is a picture rather than a metric coordinate, is arguably the most open-world goal modality for vision-language-action (VLA) navigation: it assumes no map, coordinate frame, or object vocabulary — any photograph is a valid goal. On hardware, two independent LLM-VLA systems fail the same way: the robot reaches the visual goal but does not autonomously stop. Four of five trials on an outdoor course end with the robot driving past the goal without emitting the trained stop token; the lone successful stop coincides with a near-pixel-perfect match between the current view and the goal image. The contemporaneous OmniVLA system [3] shows the same symptom on the same course, and its published deployment does not implement autonomous stop on any modality.

The failure is not a single bug. Running one trained policy on two physically distinct robots, we observe four separate image-goal failures: the predicted trajectory contracts at the goal but the stop token will not fire (M1); the trajectory never contracts and the robot circles the target (M2); the goal leaves the field of view and the stateless policy reverts to undirected exploration (M3); and the policy drives to the most visually dominant region (a communications tower behind the target, or the open sky) rather than to the goal object (M4). Section V traces all four to one cause. The goal is under-specified: the policy is trained to drive its egocentric view toward a frame sampled a fixed distance along its own future path, yet is asked at deployment to reach a specific object. It learned to match scenes, not to ground objects, so both its arrival criterion and, in the harder cases, its trajectory itself key on global view similarity rather than the designated target.

We contribute (i) a characterisation of image-goal arrival failure for an LLM-VLA navigation policy, organised as a taxonomy of four modes observed on two robot platforms running a single checkpoint, with the core symptom reproduced on a second published system (Section IV); (ii) a diagnosis tracing all four modes to goal under-specification, a train/deploy mismatch between frame-at-distance goal sampling and object-directed deployment (Section V); and (iii) a discussion that explores candidate directions a fix might take, none of which we validate here (Section VI). Our emphasis is on characterising the failure modes rather than resolving them: we ideate and suggest possible solutions but do not claim one.

II. Related work

Image-goal navigation has converged on a trajectory-prediction backbone trained on real-robot data, from GNM [8] and ViNT [9] through NoMaD's masked-diffusion extension [10]; the recent LLM-VLA line [3, 14, 13] retains this objective but routes it through a multimodal LLM. None of these systems presents a working autonomous-stop mechanism on image-goal navigation in its published deployment. Prior work that addresses arrival detection explicitly does so by training a *separate* module: Wu et al. [12] factor navigation into planning, obstacle avoidance, and navigation-ending prediction, motivated by the class-imbalance of one positive per trajectory; ViNG [7] adds a dedicated target-prediction head; VLD [6] learns a self-supervised distance-to-goal predictor from large-scale video for image and text goals and reports more reliable proximity estimates than temporal-distance baselines such as ViNT — exactly the arrival signal we find unreliable here. Suomela et al. [11] quantify at scale how image-goal policies degrade in unknown environments. The released OmniVLA code carries a scaffolded but disconnected regression-distance head (`L1RegressionDistHead`), wired into neither the loss nor the inference path, suggesting the same arrival problem was recognised but left unsolved. Our contribution is orthogonal to all of these: rather than proposing an arrival module, we characterise *why* image-goal arrival fails for an LLM-VLA trained with frame-at-distance goal sampling

Corresponding author: `peter.bohm@csiro.au`

(Section V), and show that the proximity signal such a head would consume is itself unreliable under the deployment goal distribution.

III. Setup

A. GemNav

GemNav [1] is a navigation policy built on the Gemma-4 multimodal LLM [2], fine-tuned with LoRA [4] on the language model only; the vision tower is frozen. At each step the policy consumes the current egocentric RGB frame together with a goal specification and decodes a short trajectory of robot-frame waypoints $w_{1:T}$ and a discrete `<goal_reached>` stop token. The goal may be a 2D pose, a goal image, or both, yielding three deployment modalities: pose, image, and pose-plus-image.

We train on real-robot driving data derived from SCAND [5] with *distance-based* goal sampling: for each training example the goal image is a frame taken from the same trajectory a fixed distance ahead of the current pose, and the stop token is supervised to fire as that frame is reached. The model is thus trained to drive its current view toward a frame recorded further along the same trajectory, a point we return to in Section V. A single checkpoint is used for every result in this paper, with no per-robot tuning and no action history, so the policy is stateless across steps (which bears on M3). Inference runs onboard at roughly $1\,\mathrm{Hz}$.

B. OmniVLA reference

As a generality check we compare against OmniVLA [3], the only other published hardware LLM-VLA for image-goal navigation. Every statement we make about OmniVLA is drawn from public artefacts only, the paper and the released code; we make no claims from private information. Two points matter here: the published system describes no autonomous-stop mechanism for image-goal navigation, and we reproduce the no-stop symptom on our own hardware (Section IV). The released code additionally contains a scaffolded but unused distance head (Section II).

C. Robots, goals, and success criterion

We deploy the one checkpoint on two physically distinct robots: *Spot*, a Boston Dynamics quadruped, and the *Mini-rover*, a WaveShare WAVE ROVER wheeled vehicle with visual-inertial odometry. Goal objects vary by test and are shown in the figures: an orange bin and a pair of traffic cones for Spot, and a pair of bottles for the Mini-rover. A run is a *success* when the policy autonomously emits `<goal_reached>` at the target, and a *failure* when it reaches the target area but never emits the token, so the robot drives past or circles. For the pure image-goal modality there is no metric goal, so final displacement is undefined; arrival is judged from the operator's mark on the target and the onboard video.

TABLE I: Autonomous-stop success rate. GemNav (one checkpoint) is deployed on two robots; the Mini-rover ran the image-only modality only. Spot and OmniVLA cells are five trials each; the Mini-rover image-only cell pools the 21 image-only runs of the session (the runs in Fig. 1). A metric goal fixes stopping on Spot (pose-goal final displacement $\Delta = 0.39\,\mathrm{m}$; pose-plus-image $\Delta = 0.33\,\mathrm{m}$), but the pure image goal stops about one run in five on *both* robots. OmniVLA (run on Spot) never stops: it drives the wrong way on the pose goal and overshoots the image goal.

Goal modality	Spot	Mini-rover	OmniVLA
pose (coordinate)	**5/5**	–	0/5
pose + image	**4/5**	–	–
image only	**1/5**	**4/21**	0/5

IV. Failure characterisation

Across both robots (Section III) the pose-goal and combined pose-plus-image modalities reach the target and stop reliably; the pure image-goal modality is where arrival breaks down. We characterise that breakdown here as four failure modes and, in Section V, trace them to a common cause.

A. Image-goal arrival is unreliable across systems and platforms

Table I reports an outdoor course (Spot) across the three goal modalities, with an OmniVLA [3] reproduction on the same course and goal. A metric (pose) goal stops at the target $5/5$; the image added to the pose preserves stopping $(4/5)$; but the image as the *only* goal stops just $1/5$, the other four runs reaching the target and driving past without emitting the stop token. The single success coincided with a near-pixel-perfect view match. The same checkpoint on the Mini-rover behaves the same way on the image-only goal: only $4/21$ runs autonomously stop (a rate close to Spot's $1/5$), and most of the rest contract to within a metre of the target without firing the stop token. OmniVLA, the only other published hardware LLM-VLA for image-goal navigation, never autonomously stops on this course ($0/5$ on the image goal, though it passes within 0.03–$0.81\,\mathrm{m}$ of the target), so the symptom is not specific to our system or its training data.

On both robots the image-only successes contract their predicted distance to goal smoothly into the arrival band before the stop token fires, while the failures never settle there (Fig. 1); over the same goal object and approach distance, the failing runs break down in the distinct ways below.

B. A taxonomy of four failure modes

Table II lists the four image-goal failures observed across the two platforms. They are not independent defects: Section V shows each is a consequence of how the goal is specified.

Two clarifications forestall a common confusion. First, OmniVLA's action-decoding interface carries an end-of-sequence token that terminates an action chunk; it fires every step regardless of arrival and is unrelated to navigation goal completion, so its presence should not be read as a working

TABLE II: Image-goal failure modes for one checkpoint across two robots. M1–M3 concern arrival; M4 concerns grounding. All four follow from goal under-specification (Section V).

	Description	Observed on
M1	**Stop-classification brittleness.** Trajectory contracts at the goal, but the stop token fires only on a near-exact ego-goal view match.	Spot (4/5 reach and drive past) and Mini-rover (many runs contract below 1 m yet do not stop).
M2	**Absent proximity signal.** Predicted trajectory length never contracts; the robot circles, never registering arrival.	Both robots; predicted distance stays high with wide spread (Fig. 1, failure columns).
M3	**Goal-loss exploration restart.** When the target leaves the field of view the stateless policy reverts to undirected exploration.	Both robots: predicted length collapses below 1 m, then jumps back beyond 9 m as the target exits the frame.
M4	**Distractor grounding.** The policy anchors on the most visually dominant region, not the goal object.	Spot: (4a) drove to a communications tower behind the traffic-cone goal; (4b) with the camera pitched up, arrival worked under clear sky but failed under cloud, whose texture became the salient feature.

autonomous stop.[1] Second, video supplements in this line of work conventionally end each clip at or near the moment of arrival, which omits the post-arrival interval in which a missing autonomous stop would be visible; this is verifiable from the public supplements and is a partial explanation for why the failure has drawn little attention. We make no claim about intent.

V. Diagnosis: goal under-specification

The four modes share one root cause: a mismatch between how the goal is specified at training and at deployment. During training the image goal is a frame sampled from the same trajectory a fixed distance ahead of the current pose. The supervision target is therefore "make the current view come to look like this reference view," for an arbitrary view drawn from the robot's own future path, not a designated object. At deployment we instead point the robot at a *specific* object, a bin or a cone, and supply a photograph of it. The policy was never trained to ground a named object; it was trained to drive its egocentric view toward a frame. Arrival is learned, correspondingly, as ego-goal frame similarity rather than as object proximity. Both the stop criterion and, in the harder cases, the trajectory itself inherit this: they key on global view similarity, which is dominated by whatever is most visually salient in the scene. The deficiency is in the supervision, not the modality: a human handed the same photograph completes the task, and the goal tokens sit in the policy's context at every step — training simply never supplies a reason for them to mean an object rather than a view.

[1] In the released code this terminator appears as a fixed `STOP_INDEX` on the action vocabulary, distinct from any navigation-arrival signal.

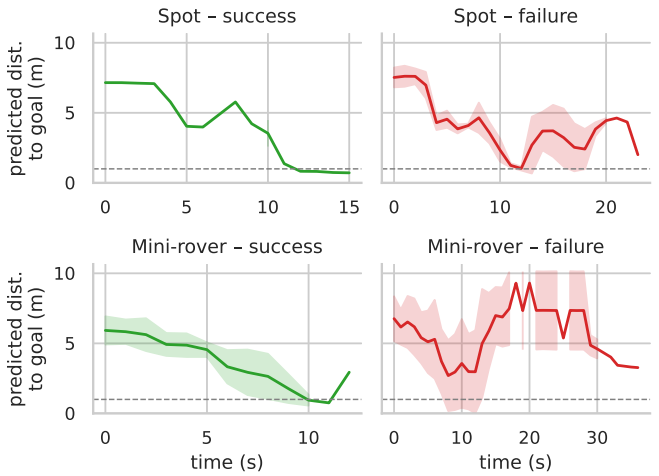

Fig. 1: Predicted distance to goal ($\|w_T\|$, the final predicted waypoint) over time on image-only runs of the same checkpoint, by robot and outcome (green: success; red: failure). Curves are per-timestep means over runs, shaded with ± 1 sd; the Spot success curve is a single run and has no band. The dashed line marks the 1 m arrival band. On failure the predicted distance never settles into that band: the wide spread collects runs that stall far from the goal (M2), lose and re-acquire it (M3), or approach without ever resolving to a stop (M1).

a) M1: stop brittleness: When the deployment goal image happens to resemble the training distribution, a frame-like view with the target ahead, the trajectory regression still contracts near the goal: the policy knows, geometrically, that it is close (Fig. 1, success columns). But the stop token, learned as a similarity threshold, fires only when the live view matches the goal image to near-pixel precision. Ordinary variation in approach angle, lighting, and final offset keeps it below threshold even at the target (Fig. 2). The stop criterion is far tighter than the metre-scale precision at which the rest of the system operates.

b) M2: absent proximity signal: When the goal image is dominated by an object the policy cannot localise as a frame match, even the trajectory regression fails (Fig. 1, failure columns): across failed runs the predicted distance never settles into the arrival band, staying high with a wide spread while the robot circles at constant speed. We observed this on both robots. The geometric proximity signal M1 depends on is absent here, which is also why a purely inference-side length threshold (Section VI) cannot rescue this case.

c) M3: exploration restart: Because the policy is stateless, carrying no goal memory between steps, losing the object from view returns it to its training prior, undirected forward driving, rather than a search for a remembered target. The predicted length collapses below 1 m and then jumps back beyond 9 m at exactly the step the target leaves the frame.

d) M4: distractor grounding: This is the clearest expression of the root cause (Fig. 3). With no object-grounding

(a) Spot, success

(b) Spot, failure (M1)

(c) Mini-rover, success

(d) Mini-rover, failure (M1)

Fig. 2: Live view (left of each pair) against the goal image (right) at the point arrival should be declared, for image-only runs on both robots. Autonomous stops coincide with a near-exact view match (a, c); failures reach the target but differ in angle, position, or lighting enough to hold the similarity-based stop below threshold (b, d). This is the brittleness behind M1.

supervision, "match the goal image" reduces to matching whatever is visually dominant. When a tall communications tower stands behind the cone goal, the policy drives to the tower (4a). When the camera is pitched up so the sky fills the frame, arrival depends on the sky: it works under a clear sky and fails under cloud, whose texture becomes the salient feature to match (4b). The policy is matching scenes, not finding objects.

The common thread is that arrival and grounding are both keyed to global view similarity, never to the goal object. A fix therefore has to change what "goal" means during training, not merely post-process the output. Section VI sketches the candidate directions, none of which we validate here.

(a) clear sky (b) overcast

Fig. 3: M4 distractor grounding. The goal is the traffic cones, but a communications tower dominates the background and the policy drives to it instead (4a). Arrival also depends on the sky: it can succeed under clear sky (left), but under overcast (right) the cloud texture becomes the salient feature and arrival fails (4b). The policy matches the dominant scene content, not the goal object.

VI. Discussion and Limitations

The four modes share a single fix target. Because they stem from goal under-specification (Section V), post-processing the policy output cannot resolve them in general: the goal representation the model is trained against has to change. We sketch candidate directions below and, consistent with a failure-mode study, validate none of them.

Object-grounded goal supervision. The most direct response to M4, and to the root cause, is to make the training goal an object rather than a recorded frame. One concrete route is synthetic goal-image augmentation: render canonical instances of target objects (e.g., from 3D meshes or CAD libraries) at randomised viewpoints, scales, and lighting, and composite them into ego-frame goal images during training. This decouples goal supervision from the precise ego-pose

at which the goal image was captured, and would teach the policy to ground a designated object rather than match the dominant scene content. It targets the distractor-grounding failures directly (a tower or the sky outweighing the cones) and should also sharpen arrival, since proximity would be measured to the object rather than to the whole view.

Language-conditioned goals. Because these policies are built on multimodal language models (GemNav on Gemma-4, OmniVLA on Llama-2), the goal need not be an image alone: the goal image can be paired with a short text instruction that names the target, for instance showing the bin and adding "go to the orange bin." Language supplies the object identity that frame matching leaves implicit, which is exactly what the distractor cases lack: a caption naming the cones gives the policy a reason to prefer them over the tower or the sky (M4), and anchors arrival to the named object rather than to the dominant scene. This reuses the frozen language model's grounding without any new perception module. The current checkpoint was not trained with text goals, so realising it needs language-conditioned examples in the goal distribution (a special case of the object-grounded supervision above); applied to the present model at inference alone it is untested and may simply be ignored, since the policy never learned to condition on a text goal.

An arrival signal, used with care. Reading arrival from the predicted trajectory length, an inference-side threshold on $\|w_T\|$ with debounce, is attractive because it needs no retraining, and OmniVLA's scaffolded distance head suggests others reached for the same idea. It can recover M1, where the trajectory does contract at the goal. But our data shows its limit: under M2 and M3 the proximity signal is itself unreliable (Fig. 1), so a length threshold cannot rescue those cases. We therefore regard it as a partial patch on M1, not a fix. A trained distance-to-object head would be more robust, but only if it is grounded on the object rather than on view similarity, which returns to the supervision question above.

Memory for goal loss. M3 is specific to a stateless policy: when the target leaves the field of view the model reverts to its training prior instead of searching for a remembered goal. A short-horizon memory of the goal, or of the last bearing to it, would let the policy persist through brief occlusions rather than restart exploration.

Alternatives we did not pursue. Two tempting inference-side gates inherit the very brittleness they would gate on. A goal-image similarity gate (stop when the live view is similar enough to the goal) is the criterion whose tightness causes M1. Rescaling the stop-token logit shifts that threshold without changing what it measures. Both fail for the same reason a length threshold fails on M2 and M3: the underlying signal is keyed to global view similarity.

When is image-goal navigation the right interface? Instance-image-goal navigation is demanding to use: the operator must already be at, or know, the goal pose well enough to capture its image, which narrows the practical value to cases such as GPS-denied return to a previously visited spot. A class- or semantic-level image goal ("go to a chair like this") is a more useful interface, but it requires a training distribution that supervises object identity rather than frame appearance, the same shift the directions above call for.

Limitations. This is a small, qualitative study: a handful of trials per condition on two robots in controlled outdoor settings, with arrival on the image-only goal judged from the operator's mark and onboard video rather than against a metric goal. The failure modes are consistent across our runs and across two platforms running one checkpoint, but we do not quantify the relative frequency of each mode or how the modes interact. We make no claim that any direction above resolves the failure; on our diagnosis a training-side change to the goal representation is the necessary step, and demonstrating one is future work.

References

[1] Peter Bohm, Saimunur Rahman, Abdelwahed Khamis, Sagun Man Singh Shrestha, Chris McCool, and Peyman Moghadam. GemNav: Discrete-token visual robot navigation using a multimodal large language model. *arXiv preprint arXiv:2607.06882*, 2026.

[2] Gemma Team. Gemma 4 model. https://ai.google.dev/gemma/docs/core/model_card_4, 2026. Google DeepMind.

[3] Noriaki Hirose, Catherine Glossop, Dhruv Shah, and Sergey Levine. Omnivla: An omni-modal vision-language-action model for robot navigation. *IEEE International Conference on Robotics and Automation*, 2026.

[4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 1(2):3, 2022.

[5] Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7(4):11807–11814, 2022.

[6] Lazar Milikic, Manthan Patel, and Jonas Frey. VLD: Visual language goal distance for reinforcement learning navigation. *arXiv preprint arXiv:2512.07976*, 2025.

[7] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Ving: Learning open-world navigation with visual goals. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13215–13222. IEEE, 2021.

[8] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

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

[10] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024.

[11] Lauri Suomela, Naoki Takahata, Sasanka Kuruppu Arachchige, Harry Edelman, and Joni-Kristian Kämäräinen. Data scaling for navigation in unknown environments. *IEEE Robotics and Automation Letters*, 2026.

[12] Qiaoyun Wu, Jun Wang, Jing Liang, Xiaoxi Gong, and Dinesh Manocha. Image-goal navigation in complex environments via modular learning. *IEEE Robotics and Automation Letters*, 7(3):6902–6909, 2022.

[13] Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, Xing Wei, and Ning Guo. Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. *International Conference on Learning Representations*, 2025.

[14] Jiazhao Zhang, Anqi Li, Yunpeng Qi, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025.