STOPWATCH: GOAL GROUNDING FOR OBJECT-GOAL NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained VLMs bring rich world knowledge and semantic priors to object-goal navigation, and fine-tuning a VLM into a navigation policy has become a common recipe. Despite this progress, success rates remain low, and it is unclear which capability these fine-tuned policies lack. Our diagnosis identifies goal grounding as the dominant failure. These policies explore and remember well, yet often fail to notice that the goal is already in view or to estimate how far away it is. To address this failure, we let our agent ground the goal explicitly before each action by stating the semantics in view and the distance to the goal. Unlike chain-of-thought reasoning, this grounding comes from a separate perception model supervised by scene annotations. Explicit grounding substantially improves success and reaches state-of-the-art results on six object-goal navigation benchmarks across three scene collections. Our analysis shows that, with ground-truth grounding, the same policy nearly solves the task, leaving the headroom in perception rather than policy learning. Our work establishes goal grounding as the central problem of object-goal navigation and offers a simple yet effective way to address it.

1 INTRODUCTION

Pretrained vision-language models (VLMs) bring web-scale visual and semantic knowledge to embodied navigation (Zhang et al., 2024; Cheng et al., 2025; Zhang et al., 2025; Xue et al., 2026; Chu et al., 2026; Zhang et al., 2026b; Wang et al., 2026; Zhang et al., 2026a). In object-goal navigation, where an agent must find an instance of a named object category in an unseen scene, fine-tuning them on expert demonstrations has become a common recipe (Zhang et al., 2025; Xiang et al., 2025; Chu et al., 2026; Zhang et al., 2026b; Wang et al., 2026; Zhang et al., 2026a). Progress has come from larger demonstration sets (Zhang et al., 2025; 2026a;b; Chu et al., 2026) and better modeling of the observation history (Zhang et al., 2024; 2025; 2026b), yet these policies still generalize poorly to unseen goal categories. Further scaling brings diminishing returns (Zhang et al., 2025; Wang et al., 2026), and some observe that perception and grounding, rather than the navigation policy, limit performance (Xue et al., 2026; Chu et al., 2026). What is missing is a diagnosis of which capability these policies lack. In this paper we give that diagnosis and build a system around it.

Our diagnosis indicates these policies fail at **goal grounding**: determining whether the goal is in view and how far away it is. They learn search well from demonstrations. On novel goal categories, they reach the goal far more often than they succeed (Fig. 1e). They can come within 1 m of the goal in 84% of episodes yet succeed in only 52%. In contrast, given the ground-truth semantics in view and distance to the goal, the same policy can have a success rate of 89%.

The diagnosis suggests a direct remedy. We make goal grounding explicit, so that before each action the agent states which objects are in view and how far the goal is, much like chain-of-thought reasoning in language models and embodied agents (Wei et al., 2022; Zawalski et al., 2024; Lin et al., 2025). Unlike free-form reasoning traces, which are generated by large VLMs and prone to noise (Zawalski et al., 2024; Xiang et al., 2025; Wang et al., 2025; Gao et al., 2026), our goal grounding is derived from the semantic and geometric labels of the training scenes. Every demonstration frame comes with its own ground-truth goal grounding at no extra labeling cost.

At test time, the grounding text must be predicted, so we build a pipeline of two models with this text as their interface. A perception model predicts the grounding from RGB observations, and a policy model reads it alongside the observations to choose the next action. The two are trained

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

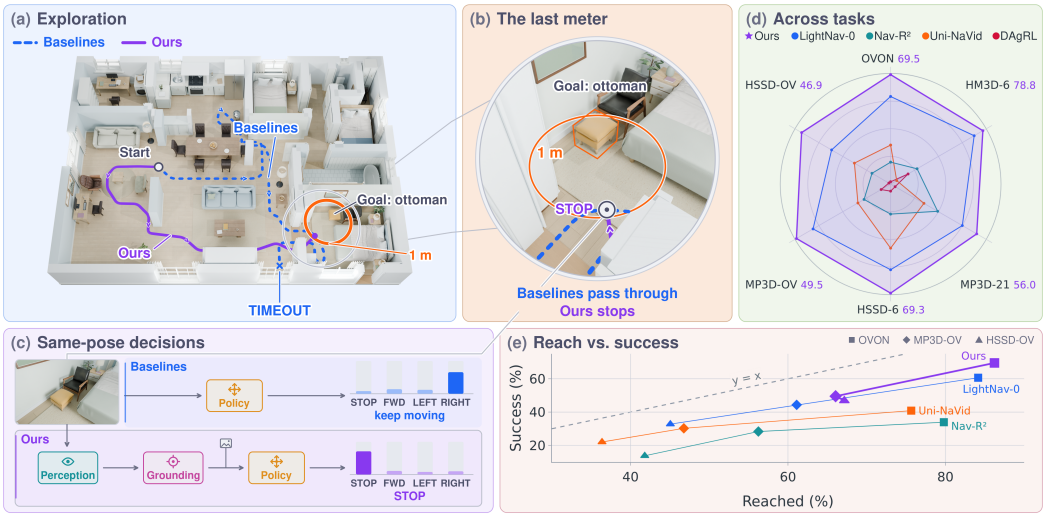


Figure 1: **Goal grounding is key to object-goal navigation. (a, b)** Baselines explore as well as ours yet walk past the goal; ours stops within 1 m. **(c)** Our system has explicit goal grounding, which enables the agent to stop at the right place. **(d)** Ours has the highest success rate of all compared systems on six benchmarks across three scene collections. **(e)** Success rate against the rate of reaching the goal vicinity. All systems reach the vicinity far more often than they succeed.

separately, the policy on ground-truth grounding and the perception model on the same annotations. The pipeline reaches state-of-the-art success on six object-goal navigation benchmarks, 8.9 points above the strongest baseline on HM3D-OVON val_unseen with corrected annotations (Tab. 1). These results reduce a long-standing failure of object-goal navigation to a concrete perception problem.

Our contributions are threefold:

- We identify **goal grounding** as the dominant failure of VLM policies for object-goal navigation, and show that correct grounding turns most of their near-goal failures into successes.
- Guided by this, we build a two-model pipeline that makes goal grounding explicit, reaching state-of-the-art results on all benchmarked object-goal navigation tasks.
- We guide future work with two findings. The remaining headroom lies in perception rather than policy learning, and current evaluations reject valid stops because of annotation defects. We audit and repair these defects, and release the corrected evaluation.

2 Related Work

Embodied navigation tasks. Navigation tasks differ in how they specify the goal. Early tasks give the goal as coordinates or an image (Anderson et al., 2018; Liu et al., 2025; Zhu et al., 2017; Krantz et al., 2023; Liu et al., 2026). These tasks are less complex and addressed by large-scale reinforcement learning (Wijmans et al., 2020). Object-goal navigation names an object category and asks the agent to find any instance of it (Batra et al., 2020; Yokoyama et al., 2024b). The agent must search the scene efficiently, remember where it has been over long horizons, and recognize the goal once it comes into view. In the open-vocabulary setting, test goals include categories never used as goals in training. Like most recent work, we study navigation in continuous environments (Zhang et al., 2024; 2025; Cheng et al., 2025), and focus on open-vocabulary object-goal navigation.

Modular navigation systems. Modular systems decompose navigation into perception and planning modules (Chaplot et al., 2020; Ramakrishnan et al., 2022). Each module can be inspected on its own, and such systems transfer to real homes far better than end-to-end policies (Gervet et al., 2023). Open-vocabulary models (Radford et al., 2021; Liu et al., 2024) made this line zero-shot. Detectors localize the goal, and image-text similarity or LLM commonsense scores the frontiers to explore (Gadre et al., 2023; Zhou et al., 2023; Yokoyama et al., 2024a). LLMs also plan in language,

parsing instructions into landmarks or reading captions of the observations (Shah et al., 2023; Zhou et al., 2024). Similarly, we make perception explicit, but only for the two cues our diagnosis finds missing. Unlike their planners, which act only on this interface by fixed rules, our policy sees every frame and learns search and stopping from demonstrations.

VLM for navigation. Rather than composing modules, recent work fine-tunes a single pretrained VLM into the navigation policy, relying on its world knowledge for what these tasks demand. This is now the dominant approach (Zhang et al., 2024; Cheng et al., 2025; Zhang et al., 2025; Xue et al., 2026; Chu et al., 2026; Zhang et al., 2026b; Wang et al., 2026; Zhang et al., 2026a). These policies follow the vision-language-action (VLA) model paradigm (Brohan et al., 2023; Kim et al., 2024), where the model is trained end-to-end by behavior cloning on expert demonstrations. However, such training has been shown to harm the model's semantic priors and language understanding (Fang et al., 2025; Driess et al., 2025; Zhou et al., 2025). We follow this line of work, but supervise perception explicitly with structured text instead of relying on action labels alone.

Intermediate reasoning. Writing out intermediate steps before the output improves both language models (Wei et al., 2022) and VLA models (Zawalski et al., 2024). In navigation, policies likewise generate reasoning text before acting, either at inference or as training-time supervision (Lin et al., 2025; Gao et al., 2026; Wang et al., 2025). The content of this text is chosen by design, typically plans, subtasks or instruction progress written by a larger model, without evidence of what the policy lacks. Our grounding text instead holds only the two cues our diagnosis finds missing, the semantics in view and the distance to the goal, both read from the scene annotations of every training frame.

3 Where Do VLM Navigators Fail?

A policy fine-tuned from a VLM should be well equipped for object navigation. The base model supplies general visual knowledge and millions of expert demonstrations supply the policy. Yet success on novel goals stays low, and further scaling returns little. Counterintuitively, the agent explores an unfamiliar scene like an expert, nears the goal, and does not stop. The pretrained knowledge does not turn into the right decisions. This section locates the root cause, derives our system from it, and repairs how success is scored.

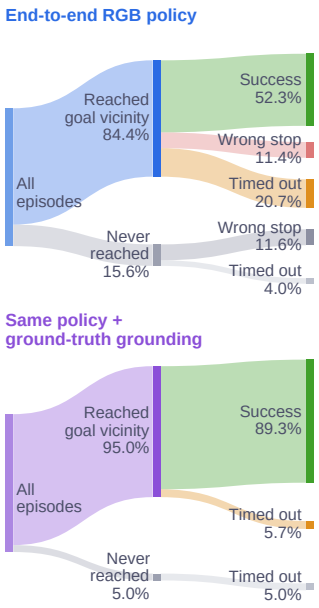


Figure 2: **Where episodes end.** We trace novel-goal episodes of an end-to-end RGB policy (top) and the same policy with ground-truth grounding (bottom).

3.1 Problem Setup

We study object-goal navigation (Zhu et al., 2017; Batra et al., 2020). In the task, an agent is dropped into an unfamiliar environment. It must find an instance of a goal category and stop near one within a step budget. In the open-vocabulary form of the task (Yokoyama et al., 2024b), the goal at test time may be unseen in training. The task couples open-ended semantics with embodied spatial search. A household robot faces this problem when asked to fetch a towel in a home it has never seen. Following the dominant approach (Zhang et al., 2024; Cheng et al., 2025; Zhang et al., 2025), we train the policy by behavior cloning (BC), where the demonstration is generated by a frontier-exploration expert (Yamauchi, 1997). We fine-tune a pretrained VLM to predict the expert's next action from the goal and the history of observations.

3.2 Failure Analysis

Reaching without stopping. Success in object-goal navigation has two requirements. The agent must reach the goal, and it must `STOP` there. Fine-tuned VLM policies mostly meet the first requirement and miss the second, which has long been noted as a distinct difficulty (Wasserman et al., 2022; Song et al., 2023). On novel goal categories, an end-to-end policy comes within 1 m of a goal in 84% of episodes, yet only succeeds in 52% under our criterion (Fig. 2 and Sec. 3.4). Near the goal it walks past without stopping. Most failures come from the model not knowing that it has arrived.

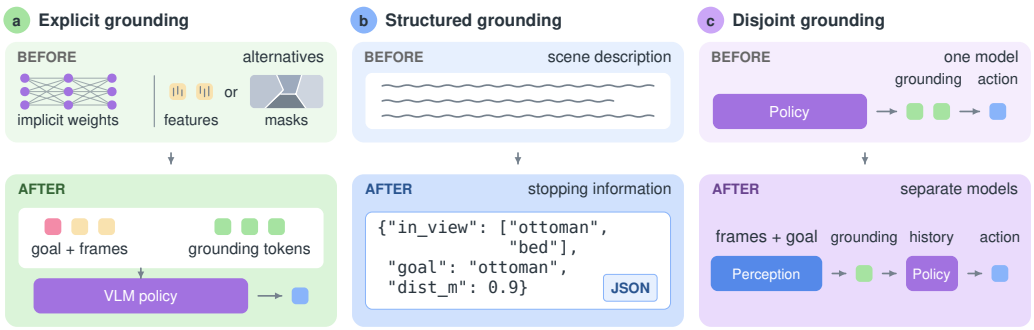


Figure 3: **Design choices.** Three decisions turn an end-to-end policy into our system. Each column shows the alternative we start from and the choice we make.

Why stopping fails. The two requirements pose very different problems for behavior cloning. The expert explores each scene in the same way until the goal comes into view, so the cloned search works just as well for a new category. Stopping, in contrast, requires recognizing the goal, and its supervision is sparse. A demonstration has a single `STOP` at its end, and only for training goal categories. From this one action per episode, the policy must learn goal grounding, namely whether the goal is in view and how far it is. Imitation alone does not make the policy use the VLM's pretrained knowledge, so it fails to stop for unseen goals.

Grounding restores stopping. To test whether goal grounding is what the policy lacks, we give the same policy the ground-truth grounding text before each action. On the same episodes, success rises from 52% to 89%, and failures after reaching the goal fall from 32% to 6% (Fig. 2, bottom). Wrong stops disappear entirely, and every remaining failure is a timeout. The end-to-end policy therefore fails at goal grounding. Our design must predict the grounding at test time (Sec. 3.3).

3.3 Design Choices

Explicit grounding. A BC-trained policy grounds the goal implicitly in its weights, and Sec. 3.2 shows that this implicit grounding fails on novel categories. Our first choice is therefore to make the grounding explicit, written out as plain text that the policy reads before acting (Fig. 3a). The policy then receives the grounding as an input at every step, instead of inferring it from the single `STOP` of each demonstration. An explicit grounding can also be inspected, which shows whether a failure comes from grounding or from acting. Modular methods carry such explicit signals as semantic maps, fused features, or segmentation masks (Chaplot et al., 2020; Gervet et al., 2023). We are instead inspired by reasoning in VLMs (Wei et al., 2022; Zawalski et al., 2024), and carry the signal as text, the input a VLM already reads, so the policy needs no new encoder.

Structured grounding. Free-form reasoning data costs extensive labeling and quality control. Instead, we take advantage of the semantic and geometric labels from the scenes themselves. The labels compile into a structured grounding text with two fields, the object categories in view and the distance to the goal (Fig. 3b). These are the two quantities that stopping needs (Sec. 3.2), and their ground-truth values close most of the failures after reaching the goal (Fig. 2). Because the text is compiled from annotations, every frame of every demonstration carries it at no labeling cost, so goal grounding is supervised at every frame rather than once per episode. The category list also names objects other than the goal, so its supervision covers categories that never serve as training goals.

Disjoint grounding. At test time no annotation exists, so the grounding text must be predicted. A single model could narrate and act in the chain-of-thought manner. However, we find that such a model navigates worse (Tab. 2), so we split the system into a perception model and a policy model (Fig. 3c). The perception model predicts the grounding text from recent frames and the goal word. The policy acts on the observations together with the grounding text, and is trained on annotated groundings. Each model thus learns one task from its own supervision. Perception becomes a

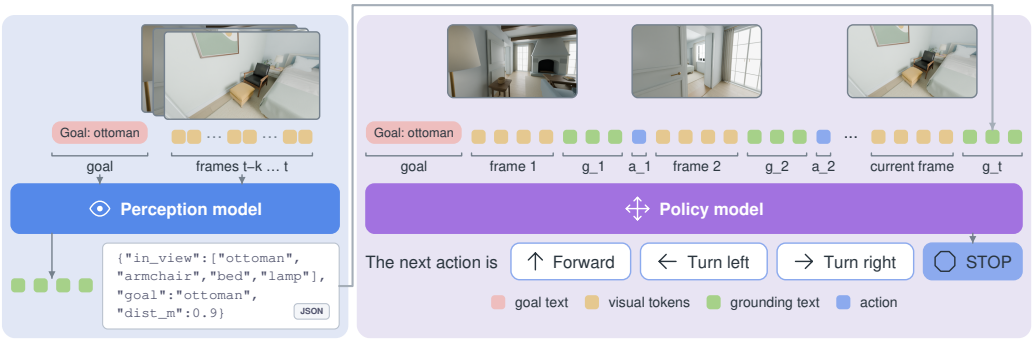


Figure 4: **System overview.** The perception model writes grounding text for each frame. The policy reads the frames interleaved with this text and outputs the next action. The two models are trained separately, both on grounding text from the scene annotations.

standalone prediction problem, trained and scaled independently of the policy, and more accurate grounding text improves the system without retraining the policy (Sec. 4.3).

The system. Fig. 4 shows the assembled system. The perception model turns each observation into grounding text. The policy reads the interleaved history of frames and grounding text and outputs the next action. Swapping the predicted text for ground truth gives the upper bound of the design, and the gap between the two runs measures the cost of imperfect perception.

3.4 Unreliable Evaluation

Inconsistent conventions. Comparing systems relies on evaluating each episode against the object names in the scene annotations. In HM3D, MP3D, and HSSD (Yadav et al., 2023; Chang et al., 2017; Khanna et al., 2024), these names are free-form strings with no shared convention. One object type appears under spelling variants and synonyms, and different scenes choose different names for it (Fig. 5a). The diversity helps training by exposing the model to a broader vocabulary, but it harms evaluation. Since the benchmark (Yokoyama et al., 2024b) credits a stop only when the annotation matches the goal word, we find agents scored a failure at perfectly valid goal instances that carry another name, such as a couch annotated as *sofa*, or a picture annotated as *painting*.

Wrong annotations. The annotations also contain two kinds of errors (Fig. 5b). In a false negative, a visible goal object has no annotation, so a correct stop at it is scored a failure. In a false positive, an annotation gives an object the wrong name, so the agent is sent to an object that is not the goal. Our manual audit filters out about a quarter of HM3D val_unseen episodes for either error.

Correcting the evaluation. Both defects make evaluations on some episodes incorrect, so we repair them. For the naming, we merge 2,408 distinct raw labels into 540 object classes and credit a stop at any instance of the goal's class. For the errors, we manually audit the evaluation episodes and filter out the affected ones. We report both the official and the corrected evaluation. The corrected evaluation separates methods more clearly. On OVON val_unseen, it raises LightNav-0 by 13 points, our system by 16, and the policy with ground-truth grounding by 19 (Tab. 1), because a stronger agent makes more of the valid stops that the official criterion rejects. Optimizing the official number would reward matching the annotators' naming rather than finding the goal.

4 Experiments

4.1 Setup

Benchmarks. We evaluate on an open-vocabulary and a closed-vocabulary task in HM3D, MP3D, and HSSD (Yadav et al., 2023; Chang et al., 2017; Khanna et al., 2024). The open-vocabulary tasks are HM3D-OVON (Yokoyama et al., 2024b) with its val_seen and val_unseen splits, MP3D-OV, and

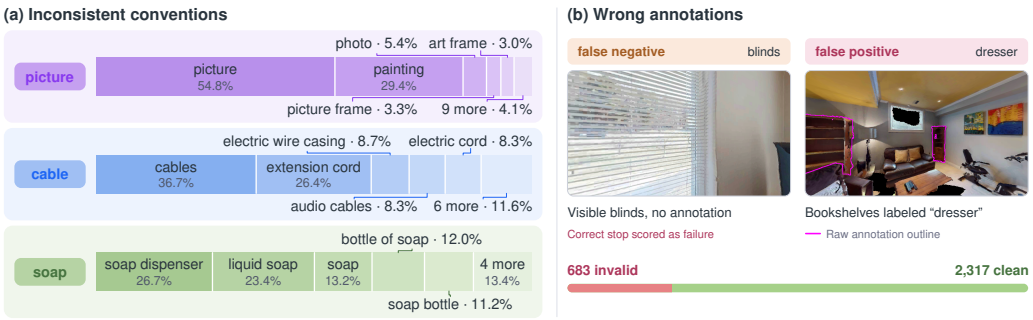


Figure 5: **Two defects of the evaluation.** (a) The same object type is annotated under many names, and only the exact annotated name earns success credit. (b) The annotations contain errors of two kinds, and a manual audit filters out 23% of the val_unseen episodes.

HSSD-OV. We compose MP3D-OV and HSSD-OV ourselves on the MP3D and HSSD validation scenes, following HM3D-OVON. The closed-vocabulary tasks are ObjectNav (Batra et al., 2020) with 6, 21, and 6 categories. The metric is success rate (SR). We omit SPL, since the shortest path from the start is no fair reference for a task that requires search. Existing benchmarks are scored under two protocols. *Official* uses the benchmark's own criterion. *Clean* uses our corrected evaluation (Sec. 3.4) on the episodes that pass the audit. MP3D-OV and HSSD-OV use only the clean protocol, marked *ours*. The appendix lists episode counts, categories, and simulator settings.

Models and training. Our perception model is fine-tuned from Qwen3-VL-4B and the policy from Qwen3-VL-2B (Bai et al., 2025). The perception model trains on 2.4M demonstrations (174M frames) from the 145 HM3D and 61 MP3D training scenes. The policy trains on 3.1M demonstrations (253M frames) from these scenes and 123 HSSD training scenes. No evaluation scene is used. The grounding text of every frame comes from the scene annotations. The policy learns actions from it, and the perception model learns to predict it. Training details are in the appendix.

Baselines. We compare open-source systems on the same episodes and criteria. Uni-NaVid, LightNav-0, and Nav-R^2 (Zhang et al., 2025; Wang et al., 2026; Xiang et al., 2025) are end-to-end VLM policies. DAgRL is the HM3D-OVON baseline (Yokoyama et al., 2024b). VLFM (Yokoyama et al., 2024a) is a zero-shot modular method. We exclude recent systems with different setups or without released checkpoints (Ren et al., 2026; Chu et al., 2026; Zhang et al., 2026b).

Does explicit goal grounding beat existing systems? *Our system leads on all six benchmarks under every protocol, and its lead lies mainly in stopping at the goal.*

4.2 Comparison with baselines

Overall comparison. Table 1 evaluates our system and the baselines on all six benchmarks under the same protocols. Our system leads on every benchmark under every reported protocol. The clean one widens the lead, from 5.8 to 8.9 points over LightNav-0 on HM3D-OVON val_unseen. The clean protocol corrects the annotation defects of Sec. 3.4, so it measures each method more faithfully. Explicit goal grounding pays off even without navigation training. Under the clean protocol, the zero-shot modular VLFM outperforms the fine-tuned end-to-end Uni-NaVid on almost all tasks. On HM3D-OVON, our own lead lies mainly in stopping. Once within 1 m of the goal, our system succeeds 81% of the time on clean val_unseen and LightNav-0 72% (Fig. 1e). This is the decision that Sec. 3.2 identifies as the main failure of end-to-end policies.

Deciding to stop. Figure 6 follows our system and LightNav-0 through one episode of HM3D-OVON val_unseen, with the goal "flower vase". Both reach the vase, coming within 0.12 m of it at steps 59 and 40. From step 57, our perception reports the vase in view within 0.5 m. The stop probability then rises, and the policy stops at step 60. LightNav-0 has the vase in view at step 40 but

Table 1: Main results, success rate in %, all systems under one protocol. *Official* columns use each benchmark's own success criterion; *clean* columns use our criterion on the audited clean subsets (Sec. 4.1). *Ours* marks the two benchmarks we build, which use the clean protocol. GT grounding is the same policy reading ground-truth grounding text, shown for reference. Shading marks the first and second best values per column, excluding GT grounding.

	HM3D						MP3D			HSSD		
	OVON val_seen		OVON val_unseen		6 cat.		OV	21 cat.		OV	6 cat.	
	official	clean	official	clean	official	clean	ours	official	clean	ours	official	clean
VLFM (Yokoyama et al., 2024a)	40.0	53.3	39.0	52.6	64.7	64.2	30.7	33.6	37.2	36.3	59.0	60.8
Nav-R^2 (Xiang et al., 2025)	39.6	51.3	27.4	33.9	58.1	54.7	28.3	36.1	39.7	13.8	35.7	33.6
DAgRL (Yokoyama et al., 2024b)	44.8	60.3	21.0	25.9	54.0	51.4	23.0	23.2	21.9	5.8	23.6	23.2
Uni-NaVid (Zhang et al., 2025)	35.4	46.1	32.1	40.9	62.5	47.2	30.3	32.2	33.9	22.0	51.4	49.0
LightNav-0 (Wang et al., 2026)	52.0	67.0	47.5	60.6	80.6	75.6	44.3	49.0	50.0	32.8	59.9	58.7
Ours	**58.7**	**77.1**	**53.3**	**69.5**	**81.5**	**78.8**	**49.5**	**50.1**	**56.0**	**46.9**	**70.0**	**69.3**
GT grounding	78.5	90.8	72.2	90.7	84.2	85.6	71.8	65.9	73.4	74.1	72.8	83.2



Figure 6: **Same episode, different decisions.** (a) Paths of both systems through the scene. (b) Views at each system's closest approach to the goal and at its STOP, labeled with the step and the action taken. Our frames also show the predicted grounding. (c) Goal distance of both systems, with our predicted goal visibility, distance bin, and stop probability. Steps 30 to 64 are enlarged.

turns away, and stops 7.4 m from it at step 280. Figure 1e shows that the end-to-end baselines often fail this way, like our end-to-end policy in Sec. 3.2.

Where does the remaining headroom lie? *Every design choice contributes, disjoint grounding the most. With ground-truth grounding, the policy nearly matches the expert, so the remaining headroom lies in perception rather than policy learning.*

4.3 Analysis

Ablation study. We ablate our design choices in Tab. 2. Disjoint grounding matters most. When one model writes the grounding text and then acts, success on val_unseen falls to 50.5%, even below the policy without any grounding. A separate perception model writing the same text raises it to 69.5%. Grounding text thus helps only when a separate model writes it. Either field alone recovers most of this gain. Each field tells the policy whether it has found the goal, which the decision to stop depends on (Sec. 3.2). Both fields together do best.

Perception as the bottleneck. Most of the remaining failures in our system come from perception. With ground-truth grounding, the same policy's success on clean HM3D-OVON val_unseen rises by 21 points (Tab. 1). To find which perception errors cost these points, we replace the predicted

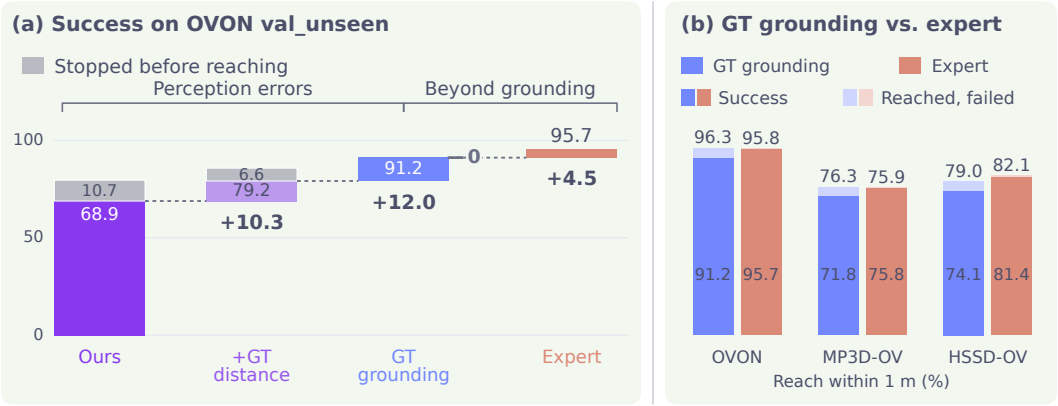


Figure 7: **From our system to the expert.** (a) Success on HM3D-OVON val_unseen as predicted grounding fields are replaced with ground truth. Gray marks episodes where the policy stops before ever coming within 1 m of the goal. (b) The policy with ground-truth grounding against the expert. Bar height is reach within 1 m of the goal, and the solid part is success. Clean subsets throughout.

grounding with ground truth in two steps (Fig. 7a). The first step corrects only the goal distance, on frames where the perception model recognizes the goal. The second step also corrects the category list. Each step raises success by a similar amount. Better goal recognition and better distance estimation both can carry over to navigation without retraining the policy.

Search with correct grounding. We next ask whether the policy itself still limits the system once grounding is correct. We compare the policy reading ground-truth grounding with the frontier-exploration expert that generated its demonstrations, on the same episodes (Fig. 7b). On all three benchmarks, the policy comes within 1 m of the goal about as often as the expert, and most of its remaining gap arises after arrival, when it fails to stop. With correct grounding, the policy searches nearly as well as its teacher in new scenes of all three collections.

Table 2: **Ablation study.** Success (%) of the design choices in Fig. 3 on the clean OVON val_seen and val_unseen splits. The last row is our design.

Disjoint	Semantic grounding	Distance grounding	Seen	Unseen
			72.3	63.4
	✓	✓	64.0	50.5
✓	✓		76.0	68.5
✓		✓	75.9	68.8
✓	✓	✓	77.1	69.5

Evaluation ceiling. We last ask how much headroom remains above the policy with ground-truth grounding. We evaluate the frontier-exploration expert that generated the demonstrations on HM3D-OVON val_unseen. The clean protocol credits the expert in 96% of the episodes. Nearly all of its failures come from running out of frontiers or steps before it finds the goal. The official protocol credits it in 72%, about the same as the policy with ground-truth grounding. The difference consists of expert stops at objects that our vocabulary accepts as the goal but the official protocol rejects (Fig. 5). Under the official protocol, most of the remaining headroom therefore reflects the evaluation rather than navigation.

Which perception errors cause the remaining failures? *Wrong goal visibility and distance cause most failures, and per-category success rises with how often perception reports the goal.*

4.4 Perception errors

Error types. Sec. 4.3 places most of the remaining gap in perception. We now ask which perception errors cause these failures. For each failed episode on HM3D-OVON val_unseen, we check what the perception model reported when the outcome was decided, at the STOP step or at a timeout near the goal. As shown in Fig. 8a, three errors account for two thirds of the failures. In a false positive, the perception model reports the goal although it is not in view. In a wrong distance, it predicts a visible goal within 0.5 m although the goal is meters away. In a false negative, it does not report the goal when it is clearly in view (Fig. 8, top). Wrong goal visibility causes 45% of the failures, and wrong goal distance causes 23%. Both parts of goal grounding still limit the system.

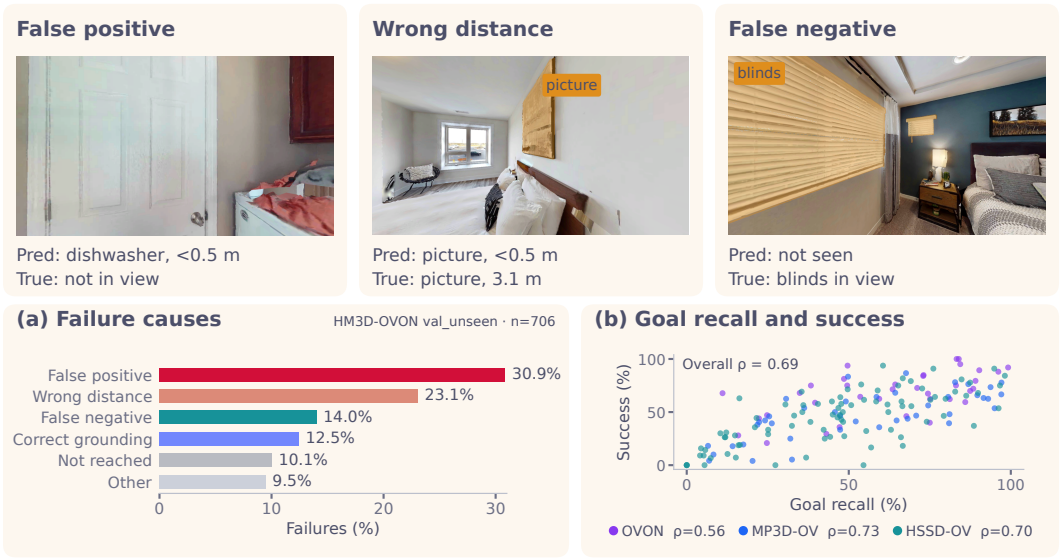


Figure 8: **Perception errors of our system.** Top, one example of each error on HM3D-OVON val_unseen, with visible true goals masked. (a) Failures on HM3D-OVON val_unseen by cause. (b) Success per goal category against how often the perception model reports the goal in view.

Correlation across categories. We last ask whether perception also explains why success differs across goal categories. For each category, we compare its success rate with how often the perception model reports the goal when it is in view. On all three benchmarks, success rises with this rate (Fig. 8b). The least reported goals are small or thin objects, such as hangers, bottles, and books, while large furniture such as beds and couches is reported reliably. With the ground-truth substitution in Sec. 4.3, this correlation indicates that higher success needs better perception.

5 Discussion and Conclusion

Broader impact. Our work offers three things. First, it identifies goal grounding as the capability that limits VLM navigation policies, and it shows that grounding can be supervised from scene annotations and improved apart from the policy. Second, it shows that the official evaluations reject valid stops, and more of them for stronger agents (Sec. 3.4). Comparisons between methods therefore need a clean evaluation, and we release the corrected one. Third, we release two new open-vocabulary benchmarks, MP3D-OV and HSSD-OV, which extend evaluation beyond HM3D.

Limitations. Our work has two main limitations. First, the grounding supervision comes from the semantic and geometric annotations of 3D scenes, so training data is limited to annotated scene collections. Learning grounding from real videos without such annotations would open much larger data sources. Second, we evaluate only in simulation, on object-goal navigation in static indoor scenes. We expect our gains to carry over to real robots, where VLM policies trained mainly on the same simulated scenes, such as LightNav-0, already run without adaptation (Wang et al., 2026).

Conclusion. Recent VLM navigation policies have grown through more demonstrations and better history modeling, yet generalization to unseen goals still lags. We traced this gap to goal grounding. The agent learns to search from demonstrations and fails at the final step, when it must tell whether the goal is in view and how far away it is. Explicit grounding splits navigation into a policy that nearly matches its teacher once grounding is correct and a perception model that accounts for most of the remaining failures. Improvements to that perception model carry over to navigation without retraining the policy. We expect object-goal navigation to advance with goal grounding.

AI USE STATEMENT

We used generative AI tools to help write code and write the paper. We also used a vision-language model to flag annotation errors when cleaning the evaluation episodes (Appendix C). We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We will release the code, the model weights, and the benchmark data, including the corrected evaluation and the MP3D-OV and HSSD-OV episodes. Sec. 4.1 and Appendices A–E describe the benchmarks, the corrected evaluation, the grounding text, the expert demonstrations, and the training details.

REFERENCES

Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *International Conference on 3D Vision (3DV)*, 2017.

Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4247–4258, 2020.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. In *RSS*, 2025.

Zedong Chu, Shichao Xie, Xiaolong Wu, Yanfen Shen, Minghua Luo, Zhengbo Wang, Fei Liu, Xiaoxu Leng, Junjun Hu, Mingyang Yin, et al. Abot-n0: Technical report on the vla foundation model for versatile embodied navigation. *arXiv preprint arXiv:2602.11598*, 2026.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. In *Advances in Neural Information Processing Systems*, volume 38, pp. 102867–102888, 2025.

Irving Fang, Juexiao Zhang, Shengbang Tong, and Chen Feng. From intention to execution: Probing the generalization boundaries of vision-language-action models. *arXiv preprint arXiv:2506.09930*, 2025.

Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. CoWs on Pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23171–23181, 2023.

Chen Gao, Liankai Jin, Xingyu Peng, Jiazhao Zhang, Yue Deng, Annan Li, He Wang, and Si Liu. Octonav: Towards generalist embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40074–40084, 2026.

Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79):eadf6991, 2023.

Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel Xuan Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2024.

Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10916–10925, 2023.

Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5945–5957, 2025.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, pp. 38–55, 2024.

Xinhao Liu, Jintong Li, Yicheng Jiang, Niranjan Sujay, Zhicheng Yang, Juexiao Zhang, John Abanes, Jing Zhang, and Chen Feng. Citywalker: Learning embodied urban navigation from web-scale videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6875–6885. IEEE, 2025.

Xinhao Liu, Jiaqi Li, Youming Deng, Ruxin Chen, Yingjia Zhang, Yifei Ma, Li Guo, Yiming Li, Jing Zhang, and Chen Feng. Wanderland: Geometrically grounded simulation for open-world embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1041–1052, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763. PMLR, 2021.

Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18868–18878. IEEE, 2022.

Botao Ren, Junjun Hu, Xinda Xue, Minghua Luo, Jintao Chen, Haochen Bai, Liangliang You, and Mu Xu. Astranav-memory: Contexts compression for long memory. In *CVPR*, 2026.

Dhruv Shah, Błażej Osiński, Brian Ichter, and Sergey Levine. LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 492–504. PMLR, 2023.

Yuhang Song, Anh Nguyen, and Chun-Yi Lee. Learning to terminate in object navigation. In *Asian Conference on Machine Learning (ACML)*, 2023.

Shaoan Wang, Aocheng Luo, Fei Huang, Jingyi Xu, Xiaoyang Wang, Yueyu Wang, Qianli Ma, Fan Yang, Ran Mei, Jia Wei, et al. Lightnav-0: Eliciting vlm spatial intelligence for generalist embodied navigation. *arXiv preprint arXiv:2608.30935*, 2026.

Shuo Wang, Yongcai Wang, Wanting Li, Xudong Cai, Yucheng Wang, Maiyue Chen, Kaihui Wang, Zhizhong Su, Deying Li, and Zhaoxin Fan. Aux-think: Exploring reasoning strategies for data-efficient vision-language navigation. In *Advances in Neural Information Processing Systems*, 2025.

Justin Wasserman, Karmesh Yadav, Girish Chowdhary, Abhinav Gupta, and Unnat Jain. Last-mile embodied visual navigation. In *Conference on Robot Learning (CoRL)*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Erik Wijmans, Abhishek Kadian, Ari S. Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2020.

Wentao Xiang, Haokang Zhang, Tianhang Yang, Zedong Chu, Ruihang Chu, Shichao Xie, Yujian Yuan, Jian Sun, Zhining Gu, Junjie Wang, Xiaolong Wu, Mu Xu, and Yujiu Yang. Nav-r^2: Dual-relation reasoning for generalizable open-vocabulary object-goal navigation. *arXiv preprint arXiv:2512.02400*, 2025.

Xinda Xue, Junjun Hu, Minghua Luo, Shichao Xie, Jintao Chen, Zixun Xie, Quan Kuichen, Wei Guo, Zedong Chu, Mu Xu, et al. Omninav: A unified framework for prospective exploration and visual-language navigation. In *International Conference on Learning Representations*, volume 2026, pp. 106216–106231, 2026.

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Brian Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1997.

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM: Vision-language frontier maps for zero-shot semantic navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 42–48, 2024a.

Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. HM3D-OVON: A dataset and benchmark for open-vocabulary object goal navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5543–5550, 2024b.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *Robotics: Science and Systems*, 2024.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *Robotics: Science and Systems*, 2025.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. In *International Conference on Learning Representations*, volume 2026, pp. 127293–127322, 2026a.

Jiazhao Zhang, Gengze Zhou, Hale Yin, Yiyang Huang, Zixing Lei, Qihang Peng, Haoqi Yuan, Jie Zhang, Xudong Guo, Xiaoyue Chen, et al. Qwen-robotnav technical report: A scalable navigation model designed for an agentic navigation system. *arXiv preprint arXiv:2606.18112*, 2026b.

Gengze Zhou, Yicong Hong, and Qi Wu. NavGPT: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024.

Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. ESC: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42829–42842. PMLR, 2023.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, Feifei Feng, and Yi Xu. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5377–5395. Association for Computational Linguistics, 2025.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3357–3364, 2017.

A BENCHMARKS AND SIMULATION

Tab. 3 lists the evaluation sets, all on scenes held out from training. HM3D-OVON and the closed-vocabulary tasks use official validation episodes. MP3D-OV and HSSD-OV use episodes generated on publisher validation scenes, with category goals from our vocabulary.

Table 3: Evaluation sets. Episodes, scenes, goal names, and goal groups describe the raw sets before exclusions. Goal groups count equivalence groups under our vocabulary. Clean counts give the adopted evaluation subsets.

Evaluation set	Episodes	Scenes	Goal names	Goal groups	Clean
HM3D-OVON val_seen	3,000	36	79	66	1,949
HM3D-OVON val_unseen	3,000	36	49	40	2,317
HM3D ObjectNav six-class	1,000	36	6	n/a	938
MP3D-OV	3,000	11	62	59	1,193
MP3D ObjectNav 21-class	2,195	11	21	n/a	1,096
HSSD-OV	3,000	43	110	104	1,751
HSSD ObjectNav six-class	1,248	42	6	n/a	974

Generated goals must have annotated instances and valid starts in multiple validation scenes. We sample navigable starts and uniform yaw angles, with the nearest accepted target reachable at a geodesic distance between 1 and 30 m.

Simulation. Our system uses the Stretch embodiment in Tab. 4. The action set is `Forward`, `Left`, `Right`, and `Stop`. Episodes that exhaust the action budget without stopping fail.

Table 4: Simulation settings for our system. Image dimensions are width by height.

Parameter	Value
Body height / radius	1.41 m / 0.17 m
Camera height	1.31 m
Rendered RGB / horizontal field of view	640×360 / 120°
Forward step / turn angle	0.25 m / 30°
Action budget / sliding	500 / enabled

B QUALITATIVE COMPARISONS

Fig. 9 shows four more episodes on which our system and LightNav-0 diverge. We selected one episode per open-vocabulary benchmark and one failure of ours from candidates ranked by their distance curves. In the rug episode (HM3D-OVON val_unseen), LightNav-0 comes within 0.01 m of the rug, continues forward, and stops 1.3 m away at step 136. Our policy stops at step 77, 0.04 m from it. In the carpet episode (MP3D-OV), both systems pass within 1 m of the carpet early. Our system returns and stops at step 149, and LightNav-0 stops 3.9 m away at step 328. In the kitchen cabinet episode (HSSD-OV), LightNav-0 comes within 0.02 m, continues, and stops 9.2 m away at step 275. Our policy stops at step 172, 0.11 m from it. In the book episode (HM3D-OVON val_unseen), our perception reports the book in view within 0.5 m at step 450 while it is not in view. Our policy stops there, 13.2 m from the goal. LightNav-0 stops 0.06 m from it at step 36.

C CORRECTED EVALUATION

Our vocabulary maps scene labels to shared object classes and groups equivalent goal names. Our success criterion requires `Stop` below 0.25 m geodesic distance to any accepted instance viewpoint, measured on the Stretch navigation mesh. Official success uses the benchmark's own goal viewpoints. MP3D-OV and HSSD-OV use only our criterion.

A vision-language model serves as an annotation auditor, comparing rendered frames with their semantic annotations. An annotation false negative is a visible goal object with no annotation; an

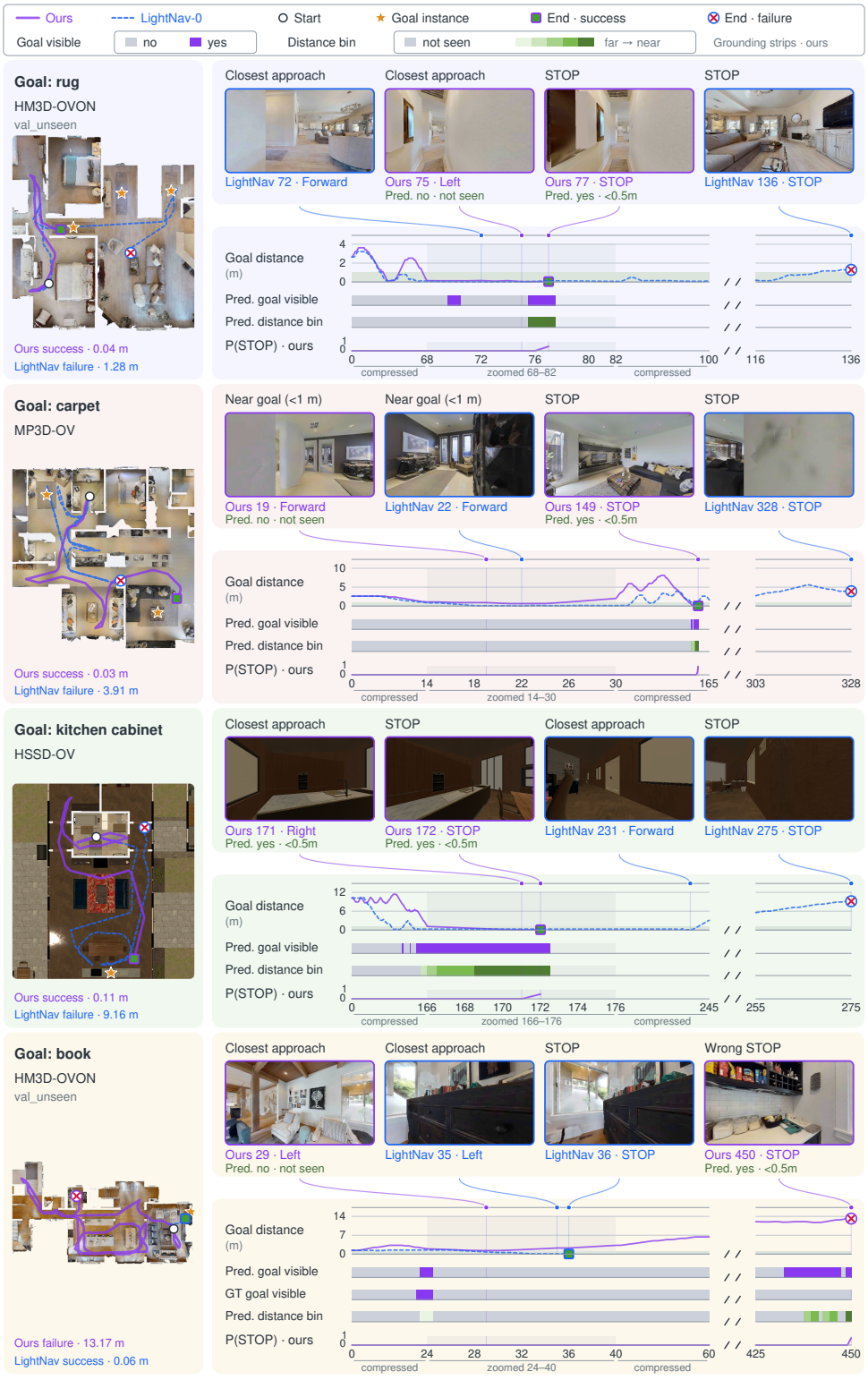


Figure 9: Same-episode comparisons of our system and LightNav-0. Each row follows the layout of Fig. 6. The last row is a failure of ours and adds the ground-truth goal visibility.

annotation false positive is an object annotated as a goal that it is not. We flag scene/category pairs with repeated evidence of either error and exclude episodes whose goal is flagged in their scene. MP3D-OV and HSSD-OV also exclude broad goal names that do not specify an object type. The clean protocol combines these exclusions with our success criterion; retained counts appear in Tab. 3.

D GROUNDING TEXT AND EXPERT DEMONSTRATIONS

Grounding text. For ObjectNav, the policy reads the fields in Tab. 5 at each observation. The inventory contains at most 16 categories ordered by decreasing image area. Categories pass area thresholds derived from training-instance geometry, with the goal retained whenever it passes its threshold. Goal visibility is inventory membership, and distance is supplied only when the goal is visible in the current image.

Table 5: Grounding text read by the policy. The action prompt follows these fields.

Field	Format
`Objects:`	Category names separated by `|`
`Goal visible:`	`true` or `false`
`Distance:`	`<0.5m`, `0.5-1m`, `1-2m`, `2-4m`, `>4m`, or `not seen`

Perception predicts the inventory and distance bin used to construct these inputs.

Frontier-exploration expert. The expert explores until it can approach an accepted goal instance. Its destination set contains viewpoints adjacent to accepted goal instances, and the distance label is the geodesic distance to the nearest destination. The expert stops within 0.15 m of its destination after facing the object.

Training corpora. The policy's corpora provide 3,138,329 demonstrations and 252,526,115 frames from 145 HM3D, 61 MP3D, and 123 HSSD training scenes. The perception model uses HM3D-OVON and MP3D object navigation plus VLN-CE instruction demonstrations, totaling 2,405,532 demonstrations and 174,482,411 frames from the HM3D and MP3D training scenes. These totals precede training-index filtering.

E TRAINING DETAILS

Tab. 6 summarizes training from the Qwen3-VL initializations (Bai et al., 2025).

Table 6: Training hyperparameters. Frame budgets are global caps per update.

Parameter	Perception	Policy
Initialization	Qwen3-VL-4B-Instruct	Qwen3-VL-2B-Instruct
Model RGB input	640×352	320×160
Maximum history frames	128	500
Updates	20,000	40,000
Frames per update, cap	16,384	8,192
Peak learning rate	10^{-5}	10^{-5}
Learning-rate schedule	Cosine	Cosine

Perception reads RGB and expert-action history during training. It learns to predict grounding text, while the policy learns expert actions from RGB, ground-truth grounding text, and prior actions. The models are trained separately.

F FAILURE ANALYSIS PROTOCOL

Figure 2 compares the same Qwen3-VL-2B policy trained twice on HM3D-OVON expert demonstrations for 20,000 updates. Both runs use 320×160 model inputs and up to 500 frames of history.

One reads RGB and prior actions. The other also reads ground-truth grounding text with distance-bin edges at 0.15, 0.25, 0.5, 1, 2, and 4 m. Both cover all 3,000 HM3D-OVON val_unseen episodes.

Reaching a goal. An episode reaches the goal vicinity when any saved trajectory position has geodesic distance below 1 m to an accepted instance viewpoint. Distances are measured on the Stretch navigation mesh over all equivalent goal instances under our vocabulary. The same target set determines success; the grounding distance instead uses the adjacent viewpoints in Appendix D.

Outcomes. Success requires `Stop` below 0.25 m geodesic distance to the accepted viewpoint set. A wrong stop is a stop that fails this criterion. A timeout executes 500 actions without stopping. Tab. 7 partitions these outcomes by reach, with all percentages using 3,000 episodes as the denominator. Failure after reaching includes both wrong stops and timeouts in the reached branch.

Table 7: Counts behind Figure 2. The ground-truth-grounding policy is the same single-source policy described above.

Branch or outcome	RGB-only	Ground-truth grounding
All episodes	3,000	3,000
Reached vicinity	2,532	2,850
Success	1,569	2,679
Wrong stop	343	0
Timeout	620	171
Never reached vicinity	468	150
Wrong stop	347	0
Timeout	121	150
Reached vicinity / all episodes	84.40%	95.00%
Success / all episodes	52.30%	89.30%
Failed after reaching / all episodes	32.10%	5.70%

Grounding ladder and expert reference. Figure 7 uses the same 4B perception model and HSSD-trained 2B policy as Tab. 1. The middle rung keeps predicted categories and substitutes ground-truth distance only when both inventories contain the goal. The full-grounding condition gives the same policy the annotated inventory and distance. The expert is the controller that generated our demonstrations. Tab. 8 gives the clean-subset counts, with reach including positions exactly 1 m from a goal viewpoint. Episodes without usable goal viewpoints count as failures.

Table 8: Clean-subset counts for Fig. 7. Pred. uses predicted grounding, GT uses ground-truth grounding, and Exp. denotes the frontier-exploration expert.

	HM3D-OVON val_unseen 2,317			MP3D-OV 1,193			HSSD-OV 1,751		
Outcome	Pred.	GT	Exp.	Pred.	GT	Exp.	Pred.	GT	Exp.
Success	1,611	2,102	2,218	591	857	904	821	1,298	1,426
Reached, wrong stop	203	64	0	65	17	0	121	24	1
Reached, timeout	184	62	1	132	36	1	234	61	10
Never reached, stop	248	0	63	201	0	98	254	0	104
Never reached, timeout	71	89	27	204	283	190	321	368	210
No usable goal viewpoint			8			0			0
Reached (%)	86.2	96.2	95.8	66.1	76.3	75.9	67.2	79.0	82.1
Success (%)	69.5	90.7	95.7	49.5	71.8	75.8	46.9	74.1	81.4

Failure attribution. Figure 8 assigns each of the 706 failures on the 2,317 clean OVON val_unseen episodes one cause. We inspect the final pre-action frame at a stop, or the interval from first reach through the end of a reached timeout. Reach uses the ladder's rule of at most 1 m. A false positive is a failed stop that reports a goal not visible under the annotation-derived visibility rule. Wrong distance is a failed stop with a visible, reported goal and an underestimated distance bin. A false negative is a reached timeout with a visible goal but no simultaneous true-positive report in the inspected

interval. Correct grounding includes failed stops with correct visibility and distance bins, reached timeouts with a true-positive goal report and predicted distance below 0.5 m, and never-reached stops with the goal neither visible nor reported. Not reached contains never-reached timeouts. Other contains unforeseen patterns and unresolved cases, including reached timeouts with no visible goal. The six counts, in this order, are 218, 163, 99, 88, 71, and 67. For the category-level analysis, we retain categories with at least ten clean episodes and compute report recall over all frames where the annotated goal is visible.