THINK-TO-ACT: GROUNDED REASONING SUPERVISION FOR GENERALIZABLE VISION-AND-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The dominant recipe for continuous vision-and-language navigation (VLN) fine-tunes a pretrained vision-language model (VLM) with an action cross-entropy loss. This loss rewards the correct action but places no semantic constraint on the model. The policy therefore overfits to action and pixel-goal outputs and loses the pretrained semantics that generalisation relies on. We call this failure *action overfitting*. It goes unnoticed in training environments but hurts navigation in unseen ones. We present Think-to-Act (T2A), which uses structured reasoning as a *training-time semantic anchor*. At every decision point, T2A supervises four types of navigation semantics (scene understanding, instruction progress, decision rationale, and stop condition). The supervision comes from T2A-GR, a grounded reasoning dataset that covers nearly the entire R2R-CE training split. An action-first target layout prevents teacher-forced leakage. Reasoning dropout and a path-consistency objective train the reasoning-free path used at deployment, so no reasoning token is decoded at test time. Under matched training, T2A improves val-unseen success rate by **5.7 points** on R2R-CE and by 4.8 points on RxR-CE, whose instructions are never annotated. It is also more effective than data replay and single-field auxiliary objectives under the same protocol. Trained only on three simulated datasets, the reasoning-free semantic system transfers zero-shot to a quadruped robot in unseen physical scenes. It succeeds in **86.7%** of trials, against 38.3% for the InternVLA-N1. Code and data will be released.

1 INTRODUCTION

Continuous vision-and-language navigation (VLN) asks an embodied agent to follow a natural-language instruction through a unseen environment while producing executable low-level motion. This task has become a proving ground for embodied AI, as it forces perception, language grounding, and control to succeed together (Wang et al., 2023; Cheng et al., 2024; Wei et al., 2025). Pretrained vision-language models (VLMs) have driven recent progress. Their scene understanding, instruction following, and visual reasoning transfer through post-training and give end-to-end navigation policies strong generalisation (InternNav Team, 2025; Wei et al., 2026; Cheng et al., 2024).

However, fine-tuning a VLM into a navigation policy tends to erode its pretrained semantics. The action cross-entropy loss only asks whether the predicted action is correct. It places no semantic constraint on the hidden states, so nothing requires them to keep scene understanding, instruction progress, decision rationale, or the stop condition. Under this supervision the model overfits to action and pixel-goal outputs, and the semantics it inherited from pretraining are lost. We call this *action overfitting*. In training environments the damage is hard to see, because action patterns can be memorised. In unseen environments it becomes a major source of error, because generalisation depends on the semantics that were lost. Recent VLN results agree with this view. Several groups have independently added reasoning supervision to training and consistently report gains in unseen environments (Wang et al., 2026b; Sun et al., 2026; Guo et al., 2026; Wang et al., 2026c).

Two families of methods have been used to enhance semantics during navigation training. *Data replay* co-trains on external VQA or video corpora alongside navigation data (Cheng et al., 2024; Wei et al., 2025; Yu et al., 2026). Its gradients target general vision–language ability, not the semantics

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.

that navigation decisions need. *Auxiliary prediction* adds a task-relevant objective such as progress monitoring (Zhu et al., 2020). It stays on the navigation data but covers only one type of semantics. Neither family constrains the whole decision process, from understanding the scene and tracking instruction progress to justifying the next move.

We propose Think-to-Act (T2A), which uses structured reasoning as a *training-time semantic anchor* (Figure 1). At every decision point, T2A supervises four types of navigation semantics. They cover scene understanding, instruction progress, decision rationale, and stop condition. The reasoning is written on the frames the model already trains on, so no external corpus is needed. We place the action first and the reasoning after it. Under causal masking the action tokens cannot read the reasoning, which prevents teacher-forced leakage, and the reasoning tokens still send semantic gradients into the backbone. A *reasoning-dropout* schedule trains the reasoning-free path used at deployment. A *path-consistency* objective aligns the latent of this path with that of the reasoning-bearing path, so the semantic gains carry over and no reasoning is decoded at test time. The same backbone can also be trained jointly with a diffusion control system, without the freezing that prior dual-system navigators rely on (Wei et al., 2026).

Our experiments support three findings. First, T2A improves generalisation to unseen environments. Under matched data, steps, and initialisation, it raises val-unseen SR on R2R-CE by **5.7 points** while val-seen SR stays flat, so the gap between seen and unseen environments narrows. The gain holds on RxR-CE ($+4.8$ SR), whose instructions are never annotated. Second, T2A is more effective than other semantic enhancement methods. Under the same protocol, replaying external VQA data or supervising a single type of semantics recovers only part of the gain, and a shuffled rationale falls below action-only training. The gain therefore comes from the content of the supervision. Third, T2A transfers to the real world. We train only on three simulated datasets and deploy the reasoning-free semantic system zero-shot on a quadruped robot in physical scenes it has never seen. It succeeds in **86.7%** of 60 trials, against 38.3% for the released InternNav slow system (InternNav Team, 2025). Route errors, such as wrong turns and missed side corridors, drop from 19 to 1.

Contributions. (1) **Action overfitting and T2A.** We identify action overfitting as a cause of poor generalisation in VLM-based navigation. We propose T2A, which enhances the lost semantics with grounded reasoning during training and keeps a reasoning-free path for deployment. (2) **The T2A-GR dataset.** A visually grounded reasoning dataset of 166,997 decision points over 10,684 R2R episodes. It covers 98.8% of the R2R-CE training split with four types of navigation semantics. (3) **Extensive experiments.** T2A improves unseen generalisation on R2R-CE and RxR-CE. It is more effective than data replay and auxiliary objectives under matched training. Trained only on three simulated datasets, it transfers zero-shot to a real quadruped robot in unseen physical scenes and leads the InternNav slow system by 48.4 points in success rate.

2 Related Work

VLM-based navigation. End-to-end VLM policies have advanced continuous VLN by scaling data and removing dependencies on depth, panoramic views, or maps (Wang et al., 2023; Cheng et al., 2024; Wei et al., 2025). Several of them mix general VQA or video corpora into training to keep broad VLM capabilities (Cheng et al., 2024; Wei et al., 2025; Yu et al., 2026). Reasoning supervision appears as auxiliary progress losses (Zhu et al., 2020), inference-time rationales (Wang et al., 2026c; Guo et al., 2026; Wang et al., 2026a), self-correction flywheels (Yu et al., 2026), and unified multimodal rationales (Zuo et al., 2026). Aux-Think (Wang et al., 2026b) trains with chain-of-thought but acts without it at test time, and ERVLA (Sun et al., 2026) mixes reasoning-bearing and reasoning-free paths for manipulation. T2A shares this asymmetry but uses reasoning to enhance pretrained semantics rather than to internalise navigation patterns. These works find that reasoning supervision improves unseen generalisation, but none traces the gain back to what action-only supervision lacks. Dual-system navigators separate semantic decision from motor control (InternNav Team, 2025; Wei et al., 2026; Liu et al., 2025; Gong et al., 2026) and typically freeze the slow system to protect generalisation (Wei et al., 2026). T2A removes this need by keeping a semantic gradient on the backbone.

Semantic loss under fine-tuning. Catastrophic forgetting is well studied in sequential learning (Kirkpatrick et al., 2017; De Lange et al., 2021). In multimodal fine-tuning, instruction tuning

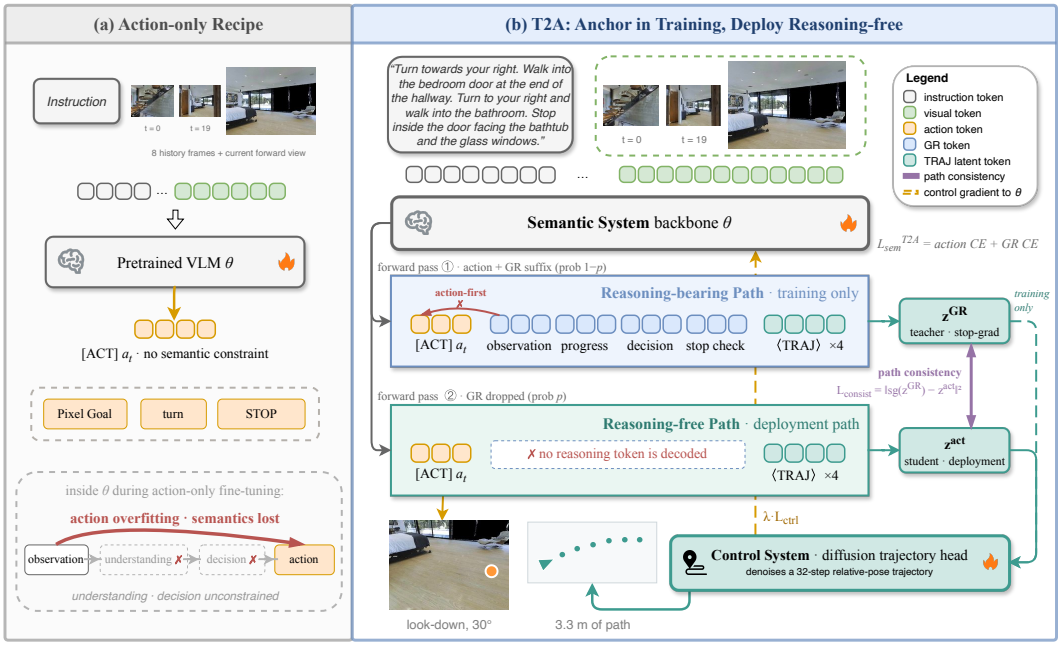


Figure 1: Overview of Think-to-Act. (a) Action-only fine-tuning places no semantic constraint on the backbone, so the policy overfits to action outputs and loses its pretrained semantics. (b) T2A supervises an action-first grounded reasoning suffix at every decision point. Reasoning dropout trains the reasoning-free path, and path consistency ($\mathcal{L}_{\text{consist}}$) aligns the latents of the two paths.

degrades VLM accuracy (Zhai et al., 2024; Luo et al., 2025), action training overwrites scene understanding while structured reasoning partially restores it (Zhou et al., 2025), and diffusion-expert gradients harm the language following of a VLM backbone (Driess et al., 2026). Common remedies such as regularisation, replay, and isolation add external costs. Two precedents outside navigation inform our design. Answer-first ordering fixes rationale leakage in chain-of-thought distillation (Chen et al., 2025), and thinking before a verifiable answer concentrates weight drift in deeper layers (Li et al., 2026). T2A enhances semantics from within the navigation task, without external corpora.

3 Method

3.1 Problem Formulation

We treat continuous VLN as a language-conditioned embodied decision process. Given an instruction $\mathcal{I} = (x_1, \ldots, x_L)$ and a stream of egocentric RGB observations $\mathbf{o}_{1:t}$, the semantic system (the slow system, or System 2, of dual-system navigators) is a policy

$$\pi_{\text{sem}}\big(a_t \mid \mathcal{I}, \mathbf{o}_{1:t}; \theta\big), \tag{1}$$

whose action a_t is a pixel goal, a discrete turn, or STOP. At four trajectory query tokens $\langle\text{TRAJ}\rangle$ appended to the end of the sequence, the backbone produces hidden states $\mathbf{h}_{\text{sem}}^{(\text{traj})}$, projected into a trajectory latent $\mathbf{z} = \text{Proj}\big(\mathbf{h}_{\text{sem}}^{(\text{traj})}\big)$ for the control system, a diffusion transformer that denoises a relative-pose trajectory conditioned on $\mathbf{z}$.

Under the standard recipe the semantic system is trained with an action cross-entropy loss alone,

$$\mathcal{L}_{\text{sem}} = -\sum_{t} \log P_\theta\big(a_t \mid \mathcal{I}, \mathbf{o}_{1:t}\big), \tag{2}$$

which only asks $\mathbf{h}_{\text{sem}}^{(\text{traj})}$ to carry enough information to name the next action. It places no semantic constraint on θ, so the pretrained representation drifts toward whatever fits the training actions. This is where action overfitting comes from. T2A adds a structured-reasoning target that constrains the

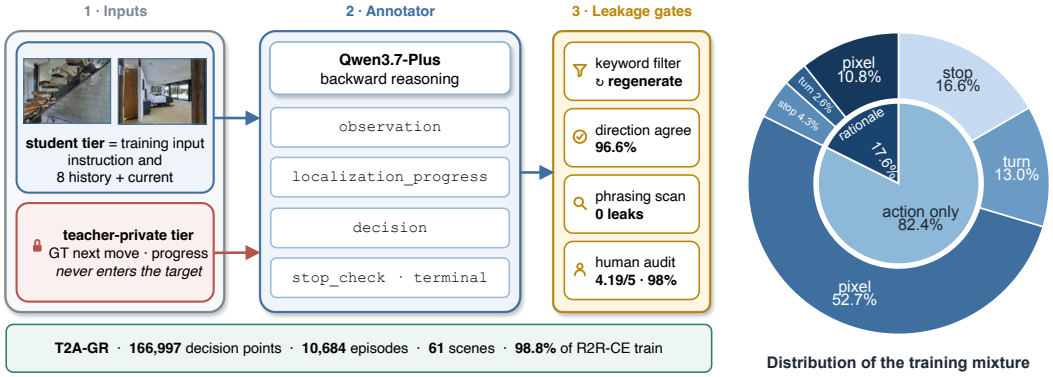


Figure 2: T2A-GR construction. (a) Backward annotation and leakage gates. (b) Training mixture by supervision (inner) and action target (outer). Details are in Appendices B and D.

backbone (§3.2–§3.4) and uses path consistency to carry the resulting representation to the deployment path (§3.4).

3.2 T2A-GR: STRUCTURED VISUAL REASONING AT DECISION POINTS

T2A-GR supplies the semantic gradient that Eq. 2 lacks. It annotates one entry per decision point on the training frames, covering **10,684 R2R episodes, 166,997 decision points, and 61 scenes** (**98.8%** of the R2R-CE training split). Each entry has four fields: `observation` (what is visible and navigable), `localization_progress` (how much of the instruction is complete), `decision` (why the chosen action follows), and `stop_check` (whether the goal has been reached, written only at terminal steps). The entries come from teacher-forced backward reasoning (Figure 2). An annotator VLM sees the student's inputs plus the ground-truth next decision and explains why that move follows. The prompt writes `observation` as if the instruction were unread and forbids directions, coordinates, and progress percentages, and a keyword filter regenerates any entry that leaks them. The direction implied by `decision` agrees with the ground-truth action in **96.6%** of entries. Appendix D gives the prompt, audits, examples, and statistics.

3.3 ACTION-FIRST AUXILIARY REASONING

Since action overfitting arises from a loss without semantic constraint, the natural remedy is to add supervised tokens that require semantic understanding. Placing the rationale before the action, however, opens a new way to lower the loss without learning. Under causal masking the action tokens attend to the gold rationale that precedes them, so the action loss can fall without the competence being acquired. This teacher-forced leakage is the *rationale sensitivity* observed in chain-of-thought distillation, where attention at answer time falls almost entirely inside the preceding rationale (Chen et al., 2025). A rationale of roughly a hundred tokens also dilutes an action of three to five in the sequence-level cross-entropy.

Following the answer-first remedy proposed for that setting (Chen et al., 2025), T2A places the action first and the rationale as a suffix:

$$\underbrace{[\text{ACTION}]\ a_t}_{\text{prefix}} \ \| \ \underbrace{\texttt{observation} \cdot \texttt{progress} \cdot \texttt{decision} \cdot \texttt{stop_check}}_{\text{auxiliary grounded reasoning suffix}} \tag{3}$$

Causal masking then provides three properties (Figure 7). Action tokens cannot see the gold rationale, so leakage cannot occur and action prediction must rest on the instruction and observations. Rationale tokens still attend to the action and remain fully supervised, so their gradients keep acting on the backbone as an auxiliary target rather than a precondition. And because the action occupies the first positions, its loss is no longer diluted by the suffix.

3.4 REASONING DROPOUT

Semantic supervision helps at test time only if the path used at deployment is also trained. T2A therefore mixes two sample types with dropout ratio p: with probability $1 - p$ the full action-first sequence with all four grounded reasoning fields, and with probability p an action-only sample whose prompt matches deployment exactly. The semantic-system objective stays a cross-entropy but now supervises both streams,

$$\mathcal{L}_{\text{sem}}^{T2A} = -\sum_t \Big[\log P_\theta\big(a_t \mid \cdot\big) + \mathbb{1}_{\text{GR}} \cdot \log P_\theta\big(\text{GR}_t \mid a_t, \cdot\big)\Big], \tag{4}$$

with $\mathbb{1}_{\text{GR}}$ set by the dropout draw. Producing the four fields requires the hidden states to encode scene semantics, progress, rationale, and stop condition, so the grounded reasoning term shapes θ whether or not a rationale is decoded later.

Path consistency. Dropout alone does not guarantee that the reasoning-free path inherits these semantics. During training, the trajectory queries can attend to the grounded reasoning suffix, so the model could keep critical semantics in that context, which the deployment path cannot reach. Path consistency closes this gap. For each sample drawn into the reasoning-bearing stream ($\mathbb{1}_{\text{GR}} = 1$), we perform a second forward pass with the grounded reasoning suffix removed, matching the deployment prompt. Let $\mathbf{z}^{\text{GR}}$ and $\mathbf{z}^{\text{act}}$ denote the trajectory latent from the reasoning-bearing and reasoning-free passes respectively. We add

$$\mathcal{L}_{\text{consist}} = \mathbb{1}_{\text{GR}} \cdot \big\|\text{sg}\big(\mathbf{z}^{\text{GR}}\big) - \mathbf{z}^{\text{act}}\big\|^2, \tag{5}$$

where $\text{sg}(\cdot)$ denotes stop-gradient, so the reasoning path, which carries the richer supervision, sets the target. The term forces the backbone to encode what the reasoning path extracts in a form the trajectory queries can read without the suffix.

3.5 FROM SEMANTICS TO CONTROL

Stage 1: semantic-system training. Starting from a pretrained VLM, we train on T2A-GR under reasoning dropout with $\mathcal{L}_{\text{sem}}^{T2A} + \mu\,\mathcal{L}_{\text{consist}}$ (Eqs. 4 and 5), $\mu = 1$.

Stage 2α: frozen transfer. We freeze θ and train only the control system, a diffusion transformer conditioned on the projected latent and on depth features estimated from RGB, with a flow-matching objective $\mathcal{L}_{\text{ctrl}} = \mathbb{E}_{t,\epsilon}\|\epsilon - \hat{\epsilon}_\phi(\mathbf{x}_t, t, \mathbf{z})\|^2$. Freezing isolates representation quality. If the T2A backbone encodes richer semantics, the control system should benefit even when no gradient reaches it.

Stage 2β: joint optimisation. We then unfreeze the semantic system and let both terms reach it,

$$\mathcal{L}_\beta = \underbrace{\mathcal{L}_{\text{sem}}^{T2A}}_{\text{semantic, via the LM head}} + \lambda \underbrace{\mathcal{L}_{\text{ctrl}}}_{\text{control, via the latent}} + \mu \underbrace{\mathcal{L}_{\text{consist}}}_{\text{path consistency}}, \qquad \lambda = \mu = 1. \tag{6}$$

The backbone now receives a semantic gradient through the language-model head and a control gradient through the latent queries. Prior work freezes the slow system (Wei et al., 2026) or cuts the action expert's gradient (Driess et al., 2026), because control gradients are expected to damage VLM generalisation. With a semantic term still acting on the backbone, T2A lets the control term help shape the representation instead. Contrasting α with β tests whether it does.

Deployment. At test time the semantic system emits the action target and latent condition with *no* reasoning tokens decoded, the control system denoises a continuous trajectory from that latent, and the controller executes it. No rationale is generated, so end-to-end latency matches a navigator trained on actions alone.

4 EXPERIMENTS

4.1 SETUP

Benchmark and metrics. We evaluate on R2R-CE in Habitat, reporting navigation error (NE↓), oracle success rate (OSR↑), success rate (SR↑), and success weighted by path length (SPL↑). Val-

Table 1: Comparison on R2R-CE val-unseen. Protocol columns mark panoramic input, depth, DAgger, extra-simulator data, and test-time reasoning. OPFT: on-policy fine-tuning (§4.1).

Method	Pano.	Depth	DAg.	Ext.	TT-R	NE↓	OSR↑	SR↑	SPL↑
ETPNav (An et al., 2024)	✓	✓				4.71	65.0	57.0	49.0
BEVBert (An et al., 2022)	✓	✓				4.57	67.0	59.0	50.0
ScaleVLN (Wang et al., 2023)	✓	✓				4.80	–	55.0	51.0
Uni-NaVid (Zhang et al., 2024)						5.58	53.3	47.0	42.7
Dynam3D (Wang et al., 2026d)		✓				5.34	62.1	52.9	45.7
NaVILA (Cheng et al., 2024)				✓		5.22	62.5	54.0	49.0
StreamVLN (Wei et al., 2025)			✓	✓		4.98	64.2	56.9	51.9
Aux-Think (R2R) (Wang et al., 2026b)						6.01	52.2	46.0	40.5
Aux-Think (scaled) (Wang et al., 2026b)				✓		6.08	60.0	54.8	46.9
InternNav S2 (InternNav Team, 2025)						4.89	60.6	55.4	52.1
InternNav dual (InternNav Team, 2025)		✓				4.83	63.3	58.2	54.0
DualVLN (Wei et al., 2026)			✓			4.05	70.7	64.3	58.5
Progress-Think (Wang et al., 2026c)					✓	4.68	63.6	60.1	53.6
ReflectVLN (Wang et al., 2026a)					✓	4.19	67.3	62.8	58.5
CorrectNav (Yu et al., 2026)			✓	✓		4.24	67.5	65.1	62.3
AwareVLN (Guo et al., 2026)			✓	✓	✓	4.02	73.5	65.4	55.1
Action only						5.24	60.3	53.3	49.4
T2A semantic						4.79	62.4	59.0	55.5
T2A semantic + OPFT			✓			4.22	67.9	64.0	58.3
T2A dual + OPFT		✓	✓			**4.07**	**70.2**	**65.8**	**59.6**

unseen is the primary split; val-seen is reported beside it because the gap between the two measures how much a policy overfits to training environments. On English RxR-CE val-unseen (3,669 episodes) we report nDTW↑ in place of OSR, following the benchmark convention. No evaluation decodes reasoning text. Episode versions, camera geometry, action space, stop rule, and controller are fixed across our runs (Appendix A).

On-policy fine-tuning (OPFT). The OPFT rows of Tables 1 and 3 continue training on the policy's own rollouts, first with DAgger trajectories annotated by the T2A-GR pipeline and then with trajectory-level GRPO (Shao et al., 2024) on simulator rewards under a KL penalty (Appendix C).

4.2 Main results

Table 1 places T2A in context. Without DAgger, the T2A semantic system exceeds the InternNav slow system (InternNav Team, 2025) by 3.6 SR and is the strongest row that uses no DAgger, external data, or test-time reasoning. It also outperforms methods with more data or richer input, such as Uni-NaVid, Dynam3D, and ScaleVLN. With on-policy fine-tuning, the dual system reaches 65.8 SR, the highest in the table, without external data or test-time reasoning. Its 59.6 SPL trails only CorrectNav (62.3), which also trains on external corpora.

4.3 Generalisation to unseen environments

Table 2 and Figure 3a give the central result. Adding grounded reasoning supervision raises val-unseen SR by 5.7 points (paired 95% episode-level bootstrap $[3.3, 8.1]$) and SPL by 6.1. Val-seen SR changes by -1.5, with a paired interval $[-5.3, 2.3]$ compatible with no change. The gap between seen and unseen SR therefore shrinks from 12.5 to 5.3 points. This is the pattern expected when action overfitting is reduced. The action-only policy fits the training environments well, but it relies on action patterns that do not transfer. With the semantics kept by grounded reasoning supervision, the policy performs nearly as well in unseen environments as in seen ones.

The gain cannot come from test-time computation, because both rows decode the same short action target. It also does not depend on the reported checkpoint. The val-seen curves intertwine across training (Figure 3c), and the 5.7-point gap is measured at the only checkpoint where both runs are

Table 2: Matched comparisons on R2R-CE. Same mixture, 15,000 steps, and initialisation.

Group	Variant	Extra corpus	seen SR↑	seen SPL↑	unseen SR↑	unseen SPL↑
	Action only		65.8	61.9	53.3	49.4
Strategies	General VQA replay	✓	64.9	60.9	53.6	49.7
	Embodied QA replay	✓	64.7	60.7	54.9	51.1
	Progress auxiliary		65.0	60.9	56.4	52.8
	Instruction auxiliary		65.3	61.3	54.2	50.4
What	Observation only		66.1	61.0	56.6	51.9
	Shuffled rationale		65.5	61.6	52.1	48.2
	Four-field, no $\mathcal{L}_{\text{consist}}$		64.4	60.2	58.2	54.8
How	Rationale-first		**66.3**	**62.4**	53.4	49.5
	Action-first, no dropout		58.7	51.4	51.1	46.0
	T2A-GR (full T2A)		64.3	60.1	**59.0**	**55.5**

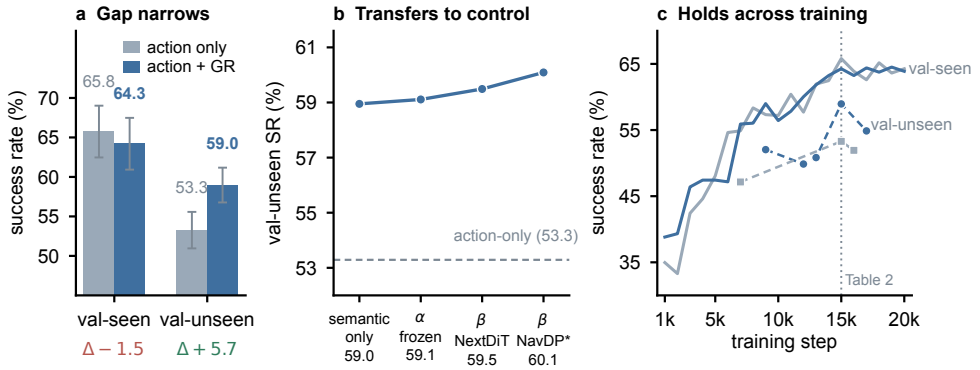


Figure 3: Grounded reasoning supervision narrows the seen/unseen gap. (a) SR with and without grounded reasoning supervision, with 95% episode-level bootstrap intervals. (b) Val-unseen SR with the control system attached, frozen (α) or jointly trained (β). (c) Val-seen SR every 1k steps and val-unseen SR at selected checkpoints (dashed); the dotted line marks the Table 2 checkpoint.

evaluated on both splits, which is also the action-only run's best val-unseen score. Figure 4 shows the reasoning-free deployment path on two val-unseen episodes, where the semantic system follows the named landmarks and stops at the goal. Full strips are in Figure 10. Appendix F gives the full strips and the episodes on which the two systems disagree.

Generalisation beyond the annotated distribution. T2A-GR annotates R2R decision points only. RxR-CE has longer instructions and a different distribution over phrasing and paths, so it lies outside the distribution of the semantic supervision, even though RxR action targets are in the training mixture. The unseen gain persists under this shift (Table 3). T2A reaches 55.6 SR, +4.8 over the matched action-only run (50.8), and exceeds every external method in the table, including both the InternNav slow system (49.5) and its dual system (53.5).

Transfer to the control system. The gain survives attaching the control system. Val-unseen SR rises from 59.0 for the semantic system alone to 59.1 under frozen transfer, and to 59.5 and 60.1 under joint optimisation with NextDiT and NavDP* (Figure 3b). NavDP* also reads depth, so the two joint points are not an ablation of each other (Table 5). On RxR-CE the jointly trained dual system adds 1.8 SR (Table 6). The T2A latent is usable by the control system even when no gradient reaches the backbone, and β exceeds α. While a semantic term keeps acting on the backbone, the control gradient helps rather than harms, so freezing is not required to protect generalisation (§3.5).

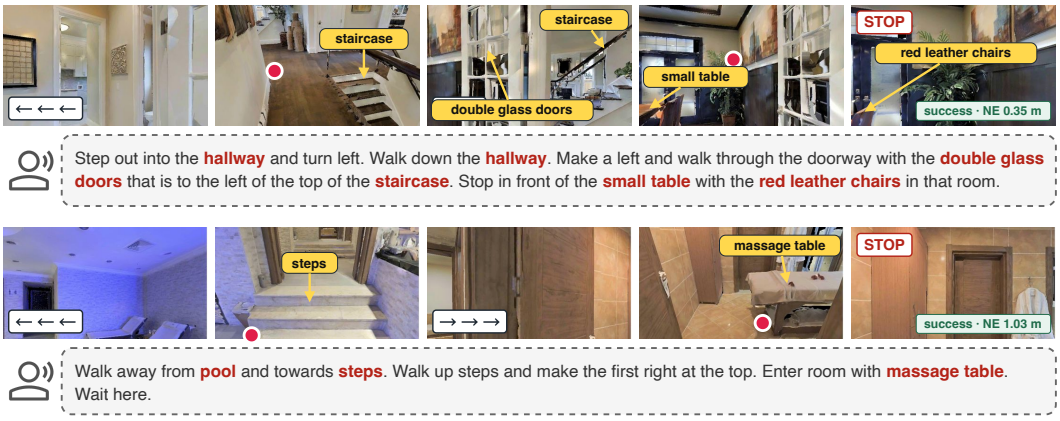


Figure 4: Two val-unseen successes of the T2A semantic system. Red markers are pixel goals and arrows are turns. Instruction landmarks are highlighted and tagged in yellow.

Table 3: Comparison on RxR-CE English val-unseen. T2A-GR annotates no RxR instruction.

Method	Pano.	Depth	DAg.	Ext.	NE↓	SR↑	SPL↑	nDTW↑
ETPNav (An et al., 2024)	✓	✓			5.64	54.7	44.8	61.9
Uni-NaVid (Zhang et al., 2024)					6.24	48.7	40.9	–
NaVILA (Cheng et al., 2024)				✓	6.77	49.3	44.0	58.8
Aux-Think (scaled) (Wang et al., 2026b)				✓	6.24	52.2	40.2	–
InternNav S2 (InternNav Team, 2025)					6.41	49.5	41.8	62.6
InternNav dual (InternNav Team, 2025)		✓			5.91	53.5	46.1	65.3
Action only					6.35	50.8	43.2	62.5
T2A semantic					5.58	55.6	47.8	66.1
T2A semantic + OPFT			✓		4.92	60.0	50.2	67.9
T2A dual + OPFT		✓	✓		**4.71**	**61.5**	**52.0**	**69.9**

4.4 Comparison with other semantic enhancement methods

The strategy rows of Table 2 instantiate the two families of methods from §1 on our recipe, with the same initialisation, steps, and reasoning-free evaluation. The replay rows replace 25% of the mixture with general (ShareGPT4V) or embodied (ScanQA) question answering. The auxiliary rows supervise only instruction progress, or reconstruct the instruction from the trajectory frames. Appendix B gives the details.

The results form a clear ranking. Data replay moves unseen SR only modestly, by $+0.3$ for General VQA and $+1.6$ for Embodied QA. Each spends 25% of the training budget on external corpora whose gradients target general vision–language ability rather than the semantics that navigation decisions need. Single-field auxiliary supervision does better, by $+3.1$ for Progress and $+0.9$ for Instruction, but each covers only one type of semantics. T2A supervises all four fields on the same in-distribution frames and leads by $+5.7$ without external data. Every row keeps seen SR within 1.5 points of the action-only baseline, so the methods differ mainly in how well they generalise.

4.5 Zero-shot transfer to a real robot

The real robot is the strongest test of generalisation in this paper. Trained only on R2R, RxR, and ScaleVLN in simulation, the T2A semantic system is deployed zero-shot on an AgiBot D1 Edu quadruped, with all inference on its on-board NVIDIA Jetson Thor. The released InternNav slow system (InternNav Team, 2025) runs through the identical pipeline, so the two runs differ only in the semantic system. We test six instructions in three office scenes built from portable furniture. The instructions are graded easy, medium, or hard by the number of navigation challenges they combine, and each is run ten times per system. Appendix G gives the platform, scenes, instructions, and protocol, and Figure 11 shows the robot and three trials.

Table 4: **Real-robot success by difficulty tier.** Ten trials per task and system. SR@d: natural stops within d m of the goal. Route err.: failed trials with a wrong turn, a missed corridor, or a lost target.

	InternNav S2 (InternNav Team, 2025)				T2A semantic			
Tier	SR@3↑	SR@1↑	SR@0.5↑	Route err.↓	SR@3↑	SR@1↑	SR@0.5↑	Route err.↓
Easy	30.0	20.0	20.0	0	**95.0**	**90.0**	**90.0**	0
Medium	60.0	20.0	5.0	4	**85.0**	**25.0**	**10.0**	**0**
Hard	25.0	20.0	5.0	15	**80.0**	**35.0**	**20.0**	**1**
All	38.3	20.0	10.0	19	**86.7**	**50.0**	**40.0**	**1**

T2A succeeds in 86.7% of trials against 38.3% (Fisher exact test, $p < 10^{-7}$) and leads on every tier and at every success radius (Table 4). On R2R-CE val-unseen the two semantic systems are 3.6 SR apart (Table 1), but in physical scenes the gap grows to 48.4 points. The failures show where the gain comes from. Of the baseline's 37 failed trials, 19 are route errors such as wrong turns and missed side corridors, against one for T2A. These are failures of instruction progress and decision rationale, two of the semantic types that T2A-GR supervises, and they mirror the failure modes in simulation (Appendix F). The stop condition shows on the shortest task, where the goal stands 2.6 m ahead. The baseline ends 3.5 to 9 m from it in 7 of 10 trials, while T2A stops within 0.4 m in all ten.

4.6 Ablations

The lower blocks of Table 2 decompose the T2A gain. The *what* block varies the supervised target, with the action-first layout and $p = 0.2$ held fixed and path consistency off, so those rows differ only in the target. The *how* block varies how the four-field target is laid out and regularised.

In the *what* block, each single type of semantics recovers part of the gap. Progress reaches 56.4 and Observation 56.6 unseen SR, roughly half the $+5.7$. Instruction reconstruction is weaker (54.2), plausibly because the instruction is already in the input prompt. Shuffled rationale (52.1) falls *below* action only (53.3), so supervision tokens with the wrong semantic content harm generalisation. The gain comes from content, not from extra token count. Four-field supervision without path consistency (58.2) captures most of the gain, and $\mathcal{L}_{\text{consist}}$ adds the final 0.8 to reach 59.0.

In the *how* block, rationale-first ordering reaches the highest seen SR in the table (66.3) but ties action only on unseen (53.4). The action tokens read the gold rationale instead of the observation, so seen-environment loss falls without transferable learning, which is the leakage that action-first ordering blocks (§3.3). Action-first without reasoning dropout collapses on both splits (58.7 seen, 51.1 unseen). Trained with the suffix always present and tested without it, neither route works. Action-first ordering, reasoning dropout, and path consistency are therefore each necessary.

5 Conclusion

We identified action overfitting in VLM-based navigation. An action loss places no semantic constraint on the model, so fine-tuning overfits the policy to action outputs and loses the pretrained semantics that generalisation relies on. Think-to-Act uses structured reasoning as a semantic anchor during training and keeps a reasoning-free path for deployment. Under matched training, it raises val-unseen SR by 5.7 points on R2R-CE and by 4.8 points on RxR-CE, and it is more effective than data replay and auxiliary objectives. Trained only in simulation, it succeeds in 86.7% of real-robot trials in unseen physical scenes, against 38.3% for the InternNav slow system. Future work includes training beyond simulation and English, extending grounded reasoning annotation beyond R2R, and probing the four types of semantics directly in the hidden states.

AI use statement

In this work, we used generative AI tools to generate a synthetic dataset. The T2A-GR reasoning annotations were produced by prompting a vision-language model over training frames, as described in §3.2 and Appendix D. The prompt design, leakage constraints, and the consistency and leakage

audits were specified and reviewed by the authors, and no annotation was accepted without passing the automated filter. Proving mathematical claims is not applicable to this work.

Additionally, we used generative AI tools to aid or polish the writing of this paper. Every number reported in the paper was taken from an evaluation log or result file and cross-checked by the authors against the source. We have reviewed all AI-assisted work and we take responsibility for the final content of this paper, including all text, claims, and artefacts.

Reproducibility statement

We will release the code and data. The release includes training and evaluation configurations, the evaluation harness, the T2A-GR dataset, and the annotation prompts and filters used to build it. The training objective and pipeline are specified in §3, the evaluation protocol in §4.1 and Appendix A, hyperparameters and compute in Appendix B, and the real-robot scenes, instructions, protocol, and per-trial results in Appendix G. Table 2 and Figure 3b report matched comparisons in which data, steps, initialisation, control-system recipe, and evaluation configuration are held fixed; the varied factor is stated with each.

References

Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022.

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5130–5145, 2024.

Xiaoshu Chen, Sihang Zhou, Ke Liang, and Xinwang Liu. Distilling reasoning ability from large language models with adaptive thinking. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

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. *arXiv preprint arXiv:2412.04453*, 2024.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Danny Driess, Jost Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *Advances in Neural Information Processing Systems*, 38:102867–102888, 2026.

Ruiyan Gong, Yingnan Guo, Junjun Hu, Jintao Kong, Xiaoxu Leng, Tianlun Li, Weize Li, Fei Liu, Zhicheng Liu, Jia Lu, et al. Abot-n1: Toward a general visual language navigation foundation model. *arXiv preprint arXiv:2607.10383*, 2026.

Wenxuan Guo, Xiuwei Xu, Yichen Liu, Xiangyu Li, Hang Yin, Huangxing Chen, Wenzhao Zheng, Jianjiang Feng, Jie Zhou, and Jiwen Lu. Awarevln: Reasoning with self-awareness for vision-language navigation. *arXiv preprint arXiv:2605.22816*, 2026.

InternNav Team. InternVLA-N1: An open dual-system navigation foundation model with learned latent plans. Technical report, 2025. URL `https://internrobotics.github.io/internvla-n1.github.io/static/pdfs/InternVLA_N1.pdf`. Code: `https://github.com/InternRobotics/InternNav`.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Ming Li, Jike Zhong, Shitian Zhao, Yuxiang Lai, Haoquan Zhang, Wang Bill Zhu, and Kaipeng Zhang. To think or not to think: A study of thinking in rule-based visual reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38:167356–167407, 2026.

Qingxiang Liu, Ting Huang, Zeyu Zhang, and Hao Tang. Nav-r1: Reasoning and navigation in embodied scenes. *arXiv preprint arXiv:2509.10884*, 2025.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. The marathon 2: A navigation system. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2718–2725. IEEE, 2020.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Nan Sun, Yuan Zhang, Yongkun Yang, Wentao Zhao, Peiyan Li, Jun Guo, Wenxuan Song, Pengxiang Ding, Runze Suo, Yifei Su, et al. Revisiting embodied chain-of-thought for generalizable robot manipulation. *arXiv preprint arXiv:2606.03784*, 2026.

Jiahang Wang, Yirong Yang, Yanqing Zhu, Minghua Luo, Shichao Xie, Fei Liu, and Mu Xu. Reflectvln: Training vision-language navigation agents with reflective reasoning. *arXiv preprint arXiv:2607.12680*, 2026a.

Shuo Wang, Yongcai Wang, Wanting Li, Xudong Cai, Yucheng Wang, Maiyue Chen, Zhizhong Su, Deying Li, and Zhaoxin Fan. Aux-think: Exploring reasoning strategies for data-efficient vision-language navigation. *Advances in Neural Information Processing Systems*, 38:30892–30916, 2026b.

Shuo Wang, Yucheng Wang, Guoxin Lian, Yongcai Wang, Maiyue Chen, Kaihui Wang, Bo Zhang, Zhizhong Su, Yutian Zhou, Wanting Li, et al. Progress-think: Semantic progress reasoning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4076–4086, 2026c.

Zihan Wang, Seungjun Lee, and Gim Hee Lee. Dynam3d: Dynamic layered 3d tokens empower vlm for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 38:153522–153544, 2026d.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986. IEEE, 2023.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

Meng Wei, Chenyang Wan, Peng Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation. In *International Conference on Learning Representations*, volume 2026, pp. 12380–12396, 2026.

Zhuoyuan Yu, Yuxing Long, Zihan Yang, Chengyan Zeng, Hongwei Fan, Jiyao Zhang, and Hao Dong. Correctnav: Self-correction flywheel empowers vision-language-action navigation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18737–18745, 2026.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language model fine-tuning. In *Conference on Parsimony and Learning*, pp. 202–227. PMLR, 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. *arXiv preprint arXiv:2412.06224*, 2024.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Yaxin Peng, Chaomin Shen, Feifei Feng, et al. 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, 2025.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10009–10019. IEEE, 2020.

Jing Zuo, Lingzhou Mu, Fan Jiang, Chengcheng Ma, Mu Xu, and Yonggang Qi. Fantasyvln: Unified multimodal chain-of-thought reasoning for vision-language navigation. *arXiv preprint arXiv:2601.13976*, 2026.

A Evaluation protocol

We evaluate on R2R-CE (`R2RVLN-v1` episodes) in Habitat over every episode of each split: val-seen is 778 episodes across 53 scenes, val-unseen 1,839 episodes across 11 scenes. The agent carries one RGB-D camera at 640×480 with a 79° horizontal field of view and depth clipped at 10 m, and episodes are cut off at 5,000 simulator steps.

The action space has the three target types of T2A-GR: a pixel goal, a discrete turn, or STOP. Habitat realises them with a 0.25 m forward step and a 15° turn, and the 30° down-pitched view that pixel goals index (Appendix D.3) is reached by two successive 15° tilts. An episode succeeds when the agent calls STOP within 3.0 m of the goal point.

Deployment closes the loop at two cadences, which is what makes a single semantic-system pass affordable. Within one pixel goal the controller executes four waypoints before the control system re-solves the trajectory from that same goal, and the semantic system is re-run only once eight forward steps have been spent against it.

All semantic-system evaluations decode no reasoning text, matching the deployment path of §3.5. Where a configuration reports both a val-seen peak and a val-unseen peak at different checkpoints, we report a single checkpoint per row so that seen and unseen figures in the same row always come from one model.

B Hyperparameters and compute

The semantic system sees eight history frames sampled uniformly over the episode so far together with the current forward view, each resized to 384×384. Its interface to the control system is four learned query tokens appended at the end of the decoded sequence, so the latent they carry is conditioned on everything decoded before them: the action target and the grounded reasoning suffix while training, the action target alone at deployment. That asymmetry is the sequence reasoning dropout trains for.

Semantic-system training. All semantic systems start from `Qwen2.5-VL-7B-Instruct` (8.29B parameters in bfloat16, of which 0.68B are the vision tower; hidden width 3584, which fixes the input width of the latent projection), never from a navigation-pretrained checkpoint, so every row of Table 2 is a fine-tune of the same public initialisation. The mixture draws on three corpora rendered under several camera configurations: four for R2R, four for RxR, and two for ScaleVLN, the last subsampled to 30%, for 4,008,478 samples in total. grounded reasoning annotation covers the R2R portion only; RxR and ScaleVLN samples carry no rationale and are therefore trained as plain action targets regardless of the dropout draw, which is why the reasoning-free fraction of an epoch is considerably larger than p alone would suggest. We use $p = 0.2$, so the rationale is kept on 80% of R2R samples, 17.6% of all samples (Figure 2b). Optimisation is AdamW under DeepSpeed ZeRO-2 with a cosine schedule, learning rate 2×10^{-5} for the language model and 5×10^{-6}

for the vision tower, warmup ratio 0.003, no weight decay, and gradient clipping at 1.0; the global batch is 128 (4 per device $\times$ 4 accumulation steps $\times$ 8 GPUs). Runs are capped at 20,000 steps and the action-only and T2A rows of Table 2 are read at the matched checkpoint used throughout §4.3. The action-only row is the same recipe with rationales switched off entirely, on the same mixture, steps, and initialisation. The strategy rows of Table 2 keep that budget and initialisation. The replay variants replace 25% of the mixture with external question-answering samples, the ratio used when replay is studied as a forgetting remedy (Zhou et al., 2025): ShareGPT4V for General VQA replay, and ScanQA, upsampled to the same slot, for Embodied QA replay. Progress auxiliary keeps the action-first layout and $p = 0.2$ of T2A but supervises only `localization_progress`, without path consistency. Instruction auxiliary converts 25% of R2R steps into a trajectory-to-instruction target. Isolation, which freezes the backbone (Wei et al., 2026; Driess et al., 2026), is not a row, since it corresponds to the frozen-transfer regime α of Figure 3b. Each run takes roughly 57 hours on 8×H100, about 450 GPU-hours. Every run in this paper uses seed 42, so the comparisons of Table 2 are matched on seed rather than averaged over seeds; the intervals in Figure 3a are episode-level bootstrap intervals.

Control-system training. Both dual configurations attach the control system to the semantic system of Table 2, so the points of Figure 3b share one T2A backbone. Joint optimisation (β) uses 5×10^{-6} for the backbone and vision tower and 1×10^{-4} for the control system, with $\lambda = 1$ and $p = 0.2$ carried over; frozen transfer (α) trains the control system alone at 1×10^{-4} with the backbone and vision tower held fixed and rationales switched off, since no semantic gradient can reach θ in that regime. Schedule, warmup, and weight decay follow the semantic-system stage; the global batch is 64 ($2 \times 4 \times 8$). The NextDiT rows run 20,000 steps on 8×H100 and are read at step 14,000 for β and 20,000 for α, costing about 34 and 21 hours; the NavDP* row runs 30,000 steps on 8×H20, is read at its final step, and costs about 127 hours.

The two control systems. Both variants predict the same object: 32 waypoints, each a relative pose $(\Delta x, \Delta y, \Delta\theta)$ in the robot frame, supervised on the demonstrated path resampled at 0.1 m intervals to 33 points so one target spans 3.3 m. Both read two 224×224 frames, the pixel-goal frame and the current one, and both average 32 trajectories sampled in parallel at inference under $\lambda = 1$. Almost everything else differs, which is why Figure 3b reports them as separate points rather than one control-system number.

Table 5: **The two control-system implementations.** Beyond the depth input marked in Table 1, the two heads also differ in objective, conditioning width, and sampling cost.

	NextDiT	NavDP*
Backbone	Lumina-NextDiT, DiT with cross-attention	Transformer decoder
Layers / width / heads	12 / 384 / 6	16 / 384 / 8
Parameters	91M	99M
Visual input	2×224^2 RGB	2×224^2 RGB and depth
Depth source	estimated from RGB	simulator depth channel
Latent interface	$3584 \rightarrow 768$, 4 tokens kept	$3584 \rightarrow 384$, compressed to 1
Objective	flow matching	DDPM, ϵ-prediction
Denoising steps	10	20
Guidance scale	1.0	none

The depth row is the difference that the protocol columns of Tables 1 and 3 record. NextDiT never sees a depth channel: it passes RGB through a DepthAnything-V2 ViT-S to obtain depth-like features, so it claims no depth input. NavDP* consumes the simulator's depth through a second, separately trained ViT-S, so its 60.1 in Figure 3b is not matched to the NextDiT point on input modality. NavDP* also compresses the four query tokens into a single conditioning token, so the wider latent interface is not what separates the two.

C ON-POLICY FINE-TUNING

On-policy fine-tuning (OPFT) continues training a converged model on its own rollouts, first with DAgger and then with reinforcement learning. Rollouts in both steps decode no reasoning, as at deployment.

DAgger. We roll out the trained semantic system on the R2R-CE training episodes. When the cumulative deviation from the reference path exceeds a tolerance, a shortest-path expert steers the agent back. We keep trajectories that reach the goal with suboptimal efficiency, annotate them with the T2A-GR pipeline (§3.2), and continue training from the Stage-1 checkpoint on the augmented mixture.

Reinforcement learning. Starting from the DAgger checkpoint, we fine-tune the semantic system with trajectory-level GRPO (Shao et al., 2024). For each instruction, the current policy samples $G = 4$ complete trajectories $\{\tau_i\}_{i=1}^{G}$ with temperature 0.7 and top-p 0.9. Each trajectory receives a scalar return R_i, normalised within its group,

$$\hat{A}_i = \frac{R_i - \mathrm{mean}_j\, R_j}{\mathrm{std}_j\, R_j + \epsilon}. \tag{7}$$

Let $\ell_\theta(\tau_i)$ be the mean log-probability of the action tokens that the semantic system emits along τ_i, computed on up to eight evenly spaced decision steps. With $\rho_i = \exp\big(\ell_\theta(\tau_i) - \ell_{\theta_{\mathrm{old}}}(\tau_i)\big)$ and $\delta_i = \ell_{\mathrm{ref}}(\tau_i) - \ell_\theta(\tau_i)$, the objective is

$$\mathcal{L}_{\mathrm{RL}} = -\frac{1}{G}\sum_{i=1}^{G} \min\Big(\rho_i \hat{A}_i,\ \mathrm{clip}(\rho_i, 1-\varepsilon, 1+\varepsilon)\,\hat{A}_i\Big) \ +\ \beta\,\frac{1}{G}\sum_{i=1}^{G}\big(e^{\delta_i} - \delta_i - 1\big), \tag{8}$$

with $\varepsilon = 0.2$ and $\beta = 0.04$. The second term is the non-negative estimator of $\mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})$, and the reference π_{ref} is the frozen DAgger checkpoint. We take one optimisation step per group.

The return sums six terms computed from simulator metrics,

$$R = 3\,\mathbb{1}[\text{success}] + \mathrm{SPL} + 0.5\, r_{\mathrm{approach}} + 0.2\,\mathbb{1}[\text{no collision}] + 1.2\, r_{\mathrm{stop}} + 0.2\, r_{\mathrm{format}}. \tag{9}$$

Success uses the 3 m criterion of Appendix A, with no partial credit for passing near the goal. r_{approach} averages $2\sigma(\Delta d_t) - 1$ over steps, where Δd_t is the decrease in geodesic distance to the goal at step t and decreases above 3 m are down-weighted as implausible jumps. r_{stop} is $+1$ when the episode ends with STOP in success or within 1.5 m of the goal, and -1 when it ends with STOP more than 3 m away. Otherwise, within 2.5 m of the goal, it is $+0.5$ with STOP and -0.5 without, and it is 0 elsewhere. r_{format} is the fraction of outputs that parse into a valid action.

The vision encoder and all control-system modules stay frozen, and only the language-model layers, embeddings, and output head are updated, with 8-bit AdamW at a constant learning rate of 2×10^{-6}. No supervised term is mixed in, so grounded reasoning is not supervised during this step. For the dual row, the fine-tuned semantic system is then attached to the control system and trained jointly with the Stage-2β recipe.

Table 6: **Contribution of each OPFT step** on R2R-CE and RxR-CE val-unseen. No reasoning is decoded.

	R2R-CE				RxR-CE			
	NE↓	OSR↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
T2A semantic	4.79	62.4	59.0	55.5	5.58	55.6	47.8	66.1
+ DAgger	4.35	67.2	62.7	57.6	5.03	58.9	49.6	67.5
+ DAgger + RL (OPFT)	4.22	67.9	64.0	58.3	4.92	60.0	50.2	67.9
T2A dual (β)	4.51	66.2	60.1	56.4	5.28	57.4	50.4	68.6
+ DAgger	4.18	69.4	64.5	58.8	4.82	60.3	51.2	69.3
+ DAgger + RL (OPFT)	**4.07**	**70.2**	**65.8**	**59.6**	**4.71**	**61.5**	**52.0**	**69.9**

Table 6 separates the two steps. DAgger contributes most of the OPFT gain. RL adds a further 1.3 SR on R2R-CE for both the semantic and the dual system, and 1.1 and 1.2 SR on RxR-CE.

D T2A-GR CONSTRUCTION

D.1 ANNOTATION PROMPT AND TIERED INPUTS

Annotation uses teacher-forced backward reasoning over two input tiers. The student-visible tier contains the natural-language instruction, eight history frames sampled uniformly over the episode so far, and the current forward view; this tier is constructed to be identical to what the training pipeline shows the model, so the annotator never sees more of the scene than the student will. The teacher-private tier contains the ground-truth next decision and the progress fraction, which let the annotator produce a rationale consistent with the demonstrated trajectory but must not appear in the student's target. The annotator is Qwen3.7-Plus throughout; no entry in the released corpus comes from a second model.

The system prompt is reproduced verbatim below. Only two things are ours: the line breaks, since each rule reaches the annotator as one continuous line, and the ASCII rendering of its three em dashes. Three of its rules carry the weight that Figure 5 is evidence for. Requiring `observation` to be written as if the instruction had not been read is what makes that field an independent visual record rather than a restatement of the instruction, which is the property the human audit scores. The visual-grounding rule forbids naming an instruction landmark that is not currently visible and requires saying so explicitly when it is absent, so a rationale cannot be produced by paraphrasing the instruction forward. And the varied-phrasing rule exists because an annotator left to itself converges on a handful of sentence openings, which would let a student reproduce the target's surface form without consulting the image.

```
You are a first-person indoor navigation agent, executing a natural-language instruction
    step by step.
You are given: the instruction, several past first-person views (oldest to newest), your
    current first-person view, and --- as a decision you have ALREADY reasoned out --- the
    correct next move.
Your job: produce your INTERNAL reasoning for THIS step, explaining what you see, where you
    are along the instruction, and why you take the next move.

Rules:
- Speak in the first person, like thinking out loud on the spot. Do NOT mention centimeters
    or degrees.
- VISUAL GROUNDING (strict): describe ONLY what is actually visible in the given views. Do
    NOT assume that objects or landmarks named in the instruction (e.g. a bed, a piano, a
    doorway) are present just because the instruction mentions them --- mention such a
    landmark ONLY if you can actually see it now. If an instruction landmark is not yet
    visible, say so explicitly (e.g. "I don't see the bed yet"). Never invent objects.
- OBSERVATION field must be a PURE visual description, written AS IF YOU HAD NOT READ THE
    INSTRUCTION: list only what you literally see (rooms, furniture, doors, windows, walls,
     floor, colors, layout). Do NOT name a target object from the instruction in `
    observation` unless it is unmistakably visible. Relate the scene to the instruction
    ONLY in `localization_progress` and `decision`.
- VARIED PHRASING (strict): vary how each field OPENS across steps. Do NOT start observation
     with "I see..." every time, and do NOT start decision with "I will..." every time.
    Rotate natural alternatives (e.g. observation: "Ahead of me...", "The room opens...", "
    A hallway...", "On my right..."; decision: "Moving forward toward...", "Continuing
    straight...", "Turning left to...", "Heading toward..."). Avoid fixed templates and
    stock phrases.
- Treat the correct next move as YOUR OWN conclusion from observing the scene. NEVER say
    things like "the given answer", "GT", "the system told me", or reference any provided
    decision/map/coordinates.
- Output ONLY a JSON object with these free-text fields (leave stop_check empty unless near
    the destination):
  {"observation": "...", "localization_progress": "...", "decision": "...", "stop_check":
    "..."}
```

The user turn is assembled in a fixed order: the instruction; the marker `[Past views: N frames, oldest first]` followed by the history images; the marker `[Current view]` followed by the current frame; the teacher-private block, opened by a line forbidding the annotator to cite or reference any of it and carrying the progress fraction, a short action history, and the ground-truth next move; and finally a one-line instruction to emit only the JSON, naming the four fields. Every record in the released corpus uses this assembly with eight history frames: 166,997 of 166,997 carry an action-history private context, and none uses a map-based one.

D.2 FIELD SCHEMA

The four fields are stored as separate keys but serialised into the training target as plain prose joined by spaces, in the fixed order `observation`, `localization_progress`, `decision`, `stop_check`. The model never sees field names or delimiters; the $\cdot$ notation of Eq. 3 denotes only this ordering. `stop_check` is a conditional slot: the prompt instructs the annotator to leave it empty away from the goal, so it is written at 9,313 decision points (5.6% of the corpus), all of them steps whose demonstrated action is STOP.

D.3 WHERE THE PIXEL GOAL LIVES

A pixel-goal target is a coordinate in the 30° down-pitched view, not in the 0° forward view, because the navigable floor ahead of the agent is often below the forward camera's frame. The training sequence therefore has two turns at these decision points: the model first emits a look-down token, receives the 30° image in reply, and only then emits the action target and the grounded reasoning suffix. Turn and STOP targets carry no pixel goal and use a single turn. This is a property of the action space rather than of T2A; we record it because Figure 5 shows both cases and the coordinate in panel (a) would otherwise appear to refer to the forward view.

D.4 LEAKAGE CONTROL AND QUALITY AUDIT

Three mechanisms keep teacher-private information out of the target. The system prompt requires `observation` to be written as though the instruction had not been read, and forbids naming directions, coordinates, or progress percentages anywhere in the output. A keyword filter scans generated text for those forms and triggers regeneration on a hit. Finally, we audit agreement between the direction implied by the written `decision` field and the ground-truth action over the full training set, which reaches 96.6%.

Independently of that audit we scanned the whole corpus for seven phrasings that could only originate in the teacher-private tier (for example *the private context*, *the correct next move*, *the provided decision*). The scan flagged four entries, which were removed; re-running it over the released corpus returns no hits.

D.5 CORPUS STATISTICS

Figure 6 summarises the corpus. Of 10,819 episodes in the R2R-CE training split, 10,684 (98.8%) carry annotation, across 61 scenes; annotation density is a median of 14 decision points per episode (interquartile range 11 to 18). Targets divide into 114,531 pixel goals, 41,784 turns, and 10,682 stops; every pixel goal annotates a move-forward decision, and the turns are 21,912 left and 19,872 right.

D.6 HUMAN QUALITY ASSESSMENT

The automated audit checks agreement with the ground-truth action, which is necessary but says nothing about whether a description is faithful to the image. We therefore had a human rater score a stratified sample of 150 entries, drawn with a fixed seed and spanning all 61 scenes and 150 distinct episodes, so that no trajectory is inspected twice. Targets are sampled 92 pixel goals, 33 turns, and 25 stops; STOP is deliberately oversampled above its 6.4% corpus rate because it is the only class in which `stop_check` is written, and 22 of the 25 drawn carry a populated field.

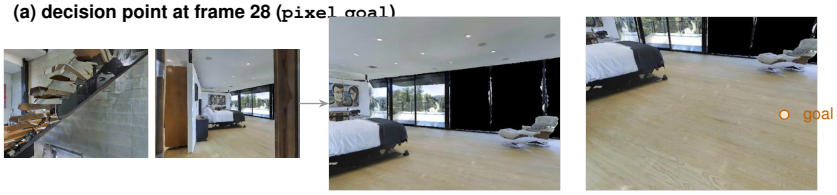
Rating proceeds in two stages, and the order matters. In stage 1 the rater sees the instruction, the eight history frames, the current view, and the four fields, but *not* the ground-truth action: a rater who has already seen the demonstrated move tends to read the image through it, which is precisely the confound that would inflate a visual-grounding score. Stage 2 reveals the ground-truth action and asks only about agreement. That the blinding held is checkable after the fact: the three entries stage 2 later marked as disagreeing with the ground truth had received stage-1 decision scores of 4, 4, and 3 against a sample mean of 4.03, so stage 1 was not reading the answer.

Table 7 reports the outcome. Visual grounding averages 4.19 of 5 with 86% of entries at 4 or above, and a single entry in 150 was flagged as naming something not visible. No entry named an in-

instruction — *"Turn towards your right. Walk into the bedroom door at the end of the hallway. Turn to your right and walk into the bathroom. Stop inside the door facing the bathtub and the glass windows."*

scene `5q7pvUzZiYa`, episode 139 (52 frames, 12 decision points)

(a) decision point at frame 28 (`pixel goal`)



history $t{=}0$ | history $t{=}19$ | current view (0°) | look-down view (30°)

`observation`: The bedroom opens up with light wood flooring and recessed ceiling lights. To my left is a bed with white linens and a dark throw, backed by a large portrait painting. Ahead and to the right, floor-to-ceiling glass windows span the wall, looking out onto a balcony and trees. A white lounge chair sits near the windows on the right side of the room.

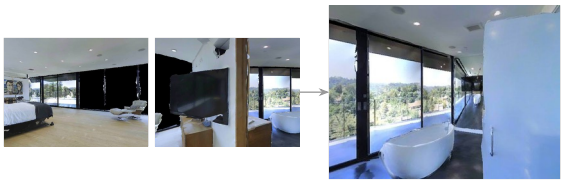
`localization_progress`: I have successfully entered the bedroom at the end of the hallway as instructed. I am now positioned inside the room and need to orient myself toward the right to find the bathroom entrance mentioned in the next part of the instruction.

`decision`: Moving forward into the open space of the bedroom while angling slightly right to better align with the far side of the room where the bathroom access should be located.

`stop_check`: *left empty by design (not a terminal step)*

`target`: `[ACTION] 550 271`

(b) decision point at frame 51 (`stop`)



no look-down turn: a `stop` target carries no pixel goal

history $t{=}28$ | history $t{=}42$ | current view (0°)

`observation`: A freestanding white oval bathtub sits on a dark floor directly ahead, positioned next to a wall of floor-to-ceiling glass windows that look out over a green hillside. To the right is a frosted glass partition or door, and the space appears to be an open bathroom area connected to the bedroom.

`localization_progress`: I have successfully navigated through the hallway, entered the bedroom, turned right into the bathroom area, and am now standing inside the doorway facing the bathtub and the glass windows as instructed.

`decision`: Having reached the final destination described in the instructions—inside the bathroom door facing the tub and windows—I will stop here.

`stop_check`: Destination reached: inside bathroom door facing bathtub and glass windows.

`target`: `[ACTION] STOP`

Figure 5: **Two decision points from one T2A-GR episode**, reproduced verbatim. *(a)* Every object named in `observation` is visible in the current view; the pixel goal is marked on the look-down frame it indexes (Appendix D.3). *(b)* `observation` independently recovers the instruction's terminal landmarks, licensing the `stop_check`. `stop_check` is blank in *(a)* by design, since it is only written at terminal steps.

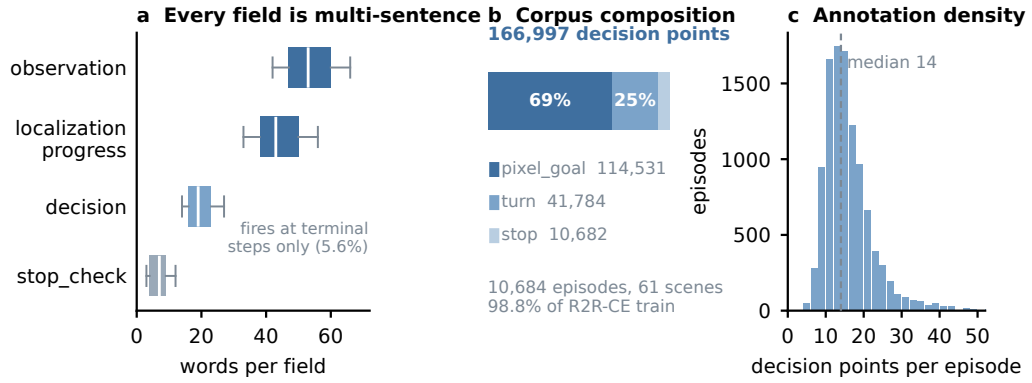


Figure 6: **T2A-GR at a glance.** **(a)** Words per field: `observation` has a median of 53 words and `localization_progress` 43; `stop_check` is written only at terminal steps. Boxes span the interquartile range, whiskers the 10th to 90th percentiles. **(b)** Composition by target type, and coverage of the R2R-CE training split. **(c)** Decision points per episode. All values are computed from the released corpus.

Table 7: **Human quality assessment**, 150 entries stratified over all 61 scenes. Stage 1 is blind to the ground-truth action; stage 2 reveals it. Scored dimensions run 1–5; flags are binary and reported as incidence.

Dimension	Result
Stage 1: ground-truth action withheld	
Visual grounding of `observation`	4.19 / 5 (86.0% scored ≥ 4)
Plausibility of `localization_progress`	4.15 / 5
Reasonableness of `decision`	4.03 / 5
Hallucinated object, room, or landmark	1 / 150 (0.7%)
Instruction landmark named but not visible	0 / 150
Stage 2: ground-truth action revealed	
`decision` agrees with ground-truth action	147 / 150 (98.0%)
`stop_check` correct, on STOP entries	22 / 22
Cites teacher-private context	0 / 150

struction landmark that was absent from the frame, which is the specific failure the `observation` prompt rule is designed to prevent. Agreement between the written `decision` and the demonstrated action is 98.0%, consistent with the 96.6% measured automatically over the whole corpus; the three disagreements are cases where the rationale is locally sensible but stops short of, or overshoots, the demonstrated waypoint. `stop_check` was judged correct on every STOP entry that carried one. The manual scan for teacher-private phrasing returned nothing, corroborating the corpus-wide automated scan.

E Why action-first ordering blocks leakage

F Additional results

On val-unseen the action-only and T2A rows of Table 2 disagree on 512 of 1,839 episodes: the grounded reasoning-supervised semantic system succeeds on 308 of those, the action-only system on 204. Figure 8 decomposes both directions by the failure mode of the losing system. The grounded reasoning-supervised system wins more often on every mode, but the largest absolute share of the net difference comes from episodes the losing system never brings to the goal region at all: 220 of the action-only system's disagreement losses end without the agent ever entering the 3 m goal region, against 152 the other way, which accounts for 68 of the 104-episode net gain. Figure 9 shows one example of each common failure of the action-only run: an early stop in the wrong room, a visit to

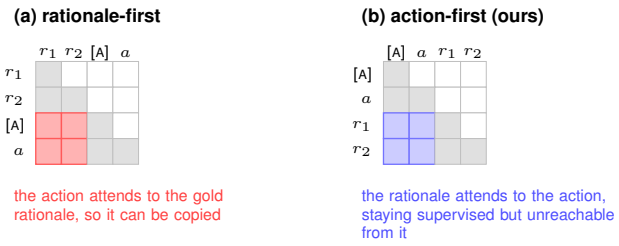


Figure 7: **Action-first ordering blocks rationale leakage.** Causal attention masks under the two orderings; filled cells mark allowed attention. *(a)* Rationale-first: the action attends to the gold rationale (red). *(b)* Action-first: the rationale attends to the action (blue), so it stays supervised but cannot be read by the action.

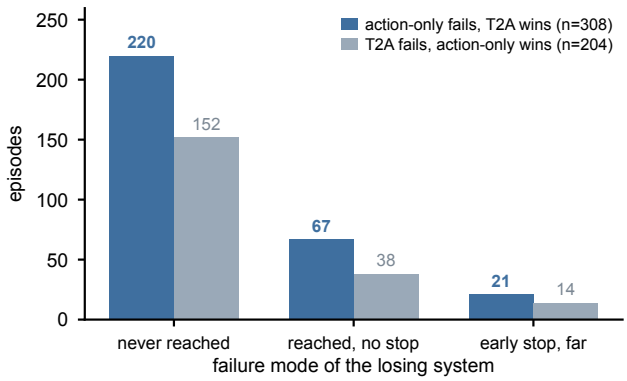


Figure 8: **Failure modes of the 512 disagreement episodes** (val-unseen, no reasoning decoded). Bars count episodes won by one system, grouped by how the losing system failed: *never reached* (oracle success never attained), *reached, no stop* (oracle success but no STOP call), and *early stop, far* (episode ends within 40 steps, final error $\geq$8 m).

the goal region without STOP, and a trajectory that never approaches the goal. Figure 10 gives the ten-frame strips of the two episodes in Figure 4 and a third val-unseen success. Both systems decode no reasoning. The frames come from a separate rendered rollout of the same checkpoints, and the numbers under each panel are from that rollout.

Coverage. T2A-GR already covers 98.8% of the R2R-CE training split. Extending annotation to RxR and retraining from scratch reaches 54.7 val-unseen SR on R2R-CE, above the 53.3 of no grounded reasoning but below the 59.0 of R2R-only annotation. The extra annotation changes the training mixture rather than the coverage of the evaluated split, so this is a question of cross-dataset mixture design; we keep the annotation matched to the evaluation distribution.

G Real-robot evaluation

G.1 Platform and deployment pipeline

The robot is an AgiBot D1 Edu quadruped carrying a ZED 2 stereo camera that publishes aligned colour and depth at 1280×720. The semantic system runs on the robot's on-board NVIDIA Jetson Thor, so no observation leaves the robot during a trial. Frames are centre-cropped and resized to 640×480 before they reach the semantic system, matching the resolution of the simulator camera (Appendix A). The semantic system is the T2A checkpoint of Table 2, trained from `Qwen2.5-VL-7B-Instruct` on simulator data only, and it decodes no reasoning, exactly as in every simulator evaluation. Depth is never an input to the semantic system: it serves only to back-project each pixel goal into a goal pose in the map frame, clipped to at most 3 m from the robot. A

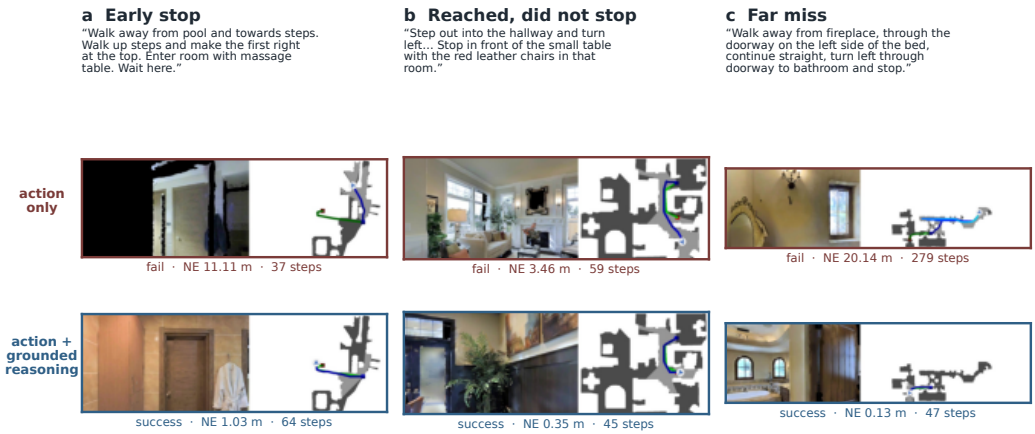


Figure 9: **Qualitative comparison on three val-unseen episodes.** Top: action-only semantic system. Bottom: same recipe with grounded reasoning supervision. Navigation error and step count are from the rendered rollout.

Nav2 local planner (Macenski et al., 2020) drives the robot to that pose at 0.4 m/s, and a STOP from the semantic system halts the planner and ends the trial. Figure 11a shows the platform. Panels b–d follow three successful T2A trials from a third-person view, one each on IF-10, LH-01, and LH-02. Each trial ends with a natural stop at the instructed goal.

G.2 Scenes and tasks

Figure 12 shows the three scenes. Each is laid out in its own metric frame, and every prop is a portable object (folding tables, plastic chairs, trash bins, potted plants, cardboard boxes) placed on a floor-taped footprint, so the layout does not depend on fixed fixtures of the building. Scene A is a 16 m straight corridor, 3 m wide. Scene C is a T-junction: a 4 m stem opens into a 15 m cross corridor whose two arms end in different objects, a chair to the left and a trash bin to the right, so a wrong turn cannot end within 3 m of the goal. Scene D is a 9×3 m hall entered from a 1×1 m alcove whose walls hide the props along the hall from the start.

Table 8 lists the instructions, issued verbatim in English. Difficulty is fixed before testing by counting the challenges a task contains, one point each: a junction decision on the way, a goal not visible from the start, a goal defined by two objects with nothing in front of it to aim at, and a same-category distractor that must be told apart by shape or material. Turning out of the alcove counts under the hidden goal rather than as a junction. A total of zero is easy, one medium, and two or more hard. The six shared tasks cover the three tiers two apiece; three further medium tasks in Scene D were run with T2A.

G.3 Protocol and metrics

Each trial involves an operator, who issues the instruction and times the run, and an independent safety observer, who holds the emergency stop and records collisions and interventions. Before a trial the props are returned to their taped footprints and the robot is aligned to the start marker and heading, and four system checks must pass: the camera-to-body transform is available, the planner is subscribed to the goal topic, colour and depth frames arrive within 5 s, and the instruction text matches the task sheet. A trial that fails a check is voided and rerun, so every task has exactly ten valid trials per system. A run is capped at 180 s, and the observer stops the robot on timeout, on leaving the scene, or on a risk of collision or fall; such trials count as failures. After a natural stop the robot is held in place and the horizontal distance from its body centre to the goal centre is measured by tape to 0.1 m. Every trial is video-recorded and photographed at its end position.

SR@d is the fraction of trials that end in a natural stop strictly within d m of the goal; a stop at exactly 1.0 m does not count toward SR@1. The field record's success flag agrees with SR@3 on

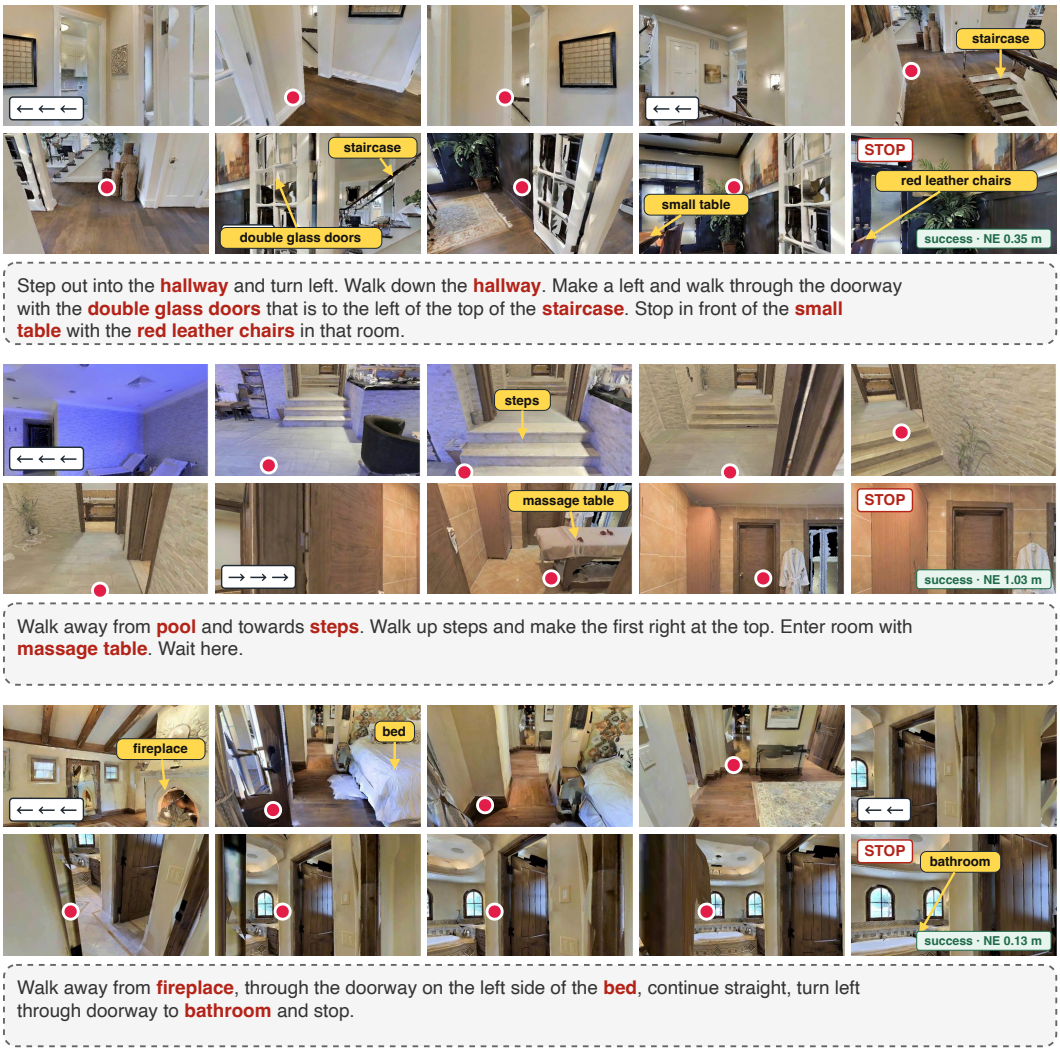


Figure 10: **Full strips of the Figure 4 successes, plus a third.** Each episode is ten frames. Red markers are its pixel goals; arrows are turn actions; yellow tags mark instruction landmarks in view. No reasoning is decoded.

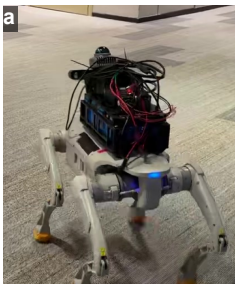
all 150 trials. Intervals are Wilson 95% intervals, and comparisons between the two systems are two-sided Fisher exact tests on the 60 trials of each.

G.4 Per-task results

Table 9 breaks Table 4 down by task. T2A leads at the 3 m radius on all six shared tasks. Pooled over them, its success rate is 86.7% [75.8, 93.1] against 38.3% [27.1, 51.0] for InternNav S2, with $p = 5.8 \times 10^{-8}$; the difference stays significant at 1 m ($p = 1.0 \times 10^{-3}$) and at 0.5 m ($p = 2.5 \times 10^{-4}$). Over all nine tasks T2A succeeds in 82 of 90 trials, 91.1% [83.4, 95.4], and all 30 trials of the three medium tasks that must first leave the alcove toward a hidden goal end in success.

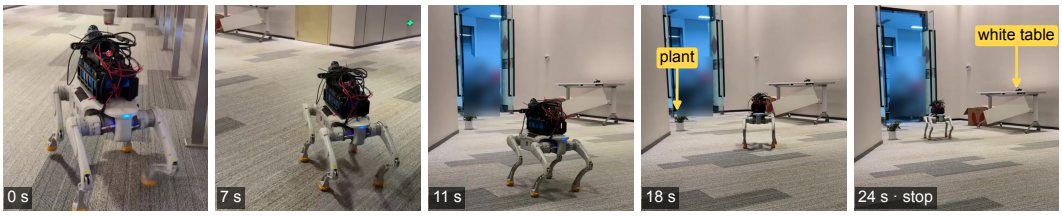
G.5 Failure modes and stop precision

Table 10 sorts the failed trials. The baseline's route errors concentrate on the three tasks that demand a decision the instruction specifies and the scene does not make obvious: four on IF-05, where it turns right instead of left, while it never turns the wrong way on the mirror task IF-04; eight on LH-01, where it keeps to the cross corridor past the stem it should turn into; and seven on LH-02,

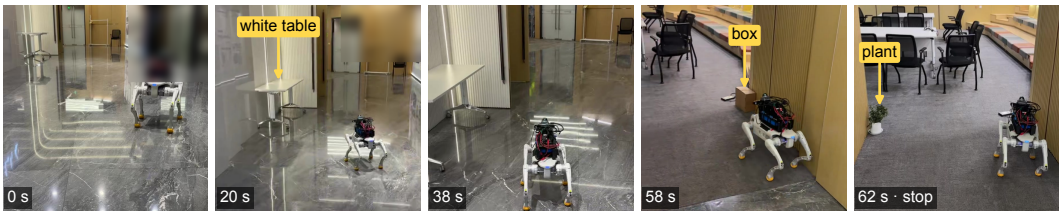


Robot	AgiBot D1 Edu quadruped
Camera	ZED 2 stereo; RGB at 640×480 to the semantic system, depth only to back-project pixel goals
Compute	NVIDIA Jetson Thor on board; no off-board inference
Execution	Nav2 local planner at 0.4 m/s; waypoints at most 3 m ahead
Training data	simulation only (R2R, RxR, ScaleVLN); zero-shot deployment

b **IF-10 · Scene D · medium (hidden goal)**
"Leave the alcove and turn left. Walk past the potted plant on your left, then stop in front of the white table on your right."



c **LH-01 · Scene C · hard (junction, hidden goal, between two objects)**
"Walk forward along the corridor, pass the white table on your left, turn left into the side corridor, and stop between the potted plant on your left and the cardboard box on your right."



d **LH-02 · Scene A · hard (hidden goal, same-category distractor)**
"Turn left and walk down the corridor. Pass the white table and the chair on your left, keep going past the round plastic trash bin on your right, and stop next to the rectangular metal trash bin near the end of the corridor."

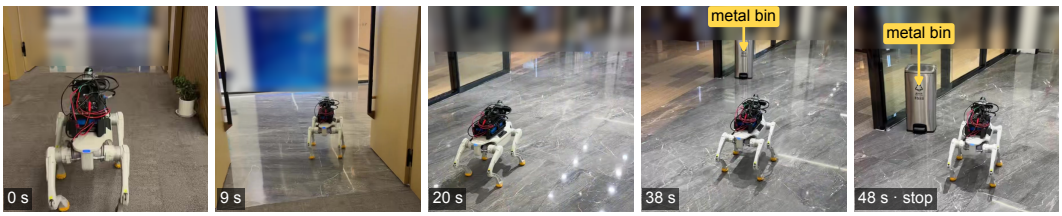


Figure 11: **The robot and three real-robot trials of T2A.** *(a)* The quadruped and its deployment pipeline. *(b)–(d)* One successful trial on each of three tasks, shown as five third-person frames with the instruction issued verbatim above the row. Tags mark landmarks named in the instruction. Times count from the first frame shown, and the last frame is the natural stop. No reasoning is decoded. Displays, signage, and a bystander are blurred for anonymity.

where, starting against a wall, it loses the target and is stopped after taking a wrong branch. All twelve of its far misses fall on the two easy tasks in Scene D, where it ends 5 to 14 m from the goal, although on GS-09 the goal stands 2.6 m ahead of the start. T2A's eight failures are spread over five tasks, at most two on any one.

Figure 13 plots every trial. Successful T2A trials cluster closer to the goal: 24 of its 52 successes end within 0.5 m, against 6 of the baseline's 23. On the three short object-directed tasks, GS-08, GS-09, and IF-11, T2A stops within 0.5 m of the goal in 26 of 30 trials.

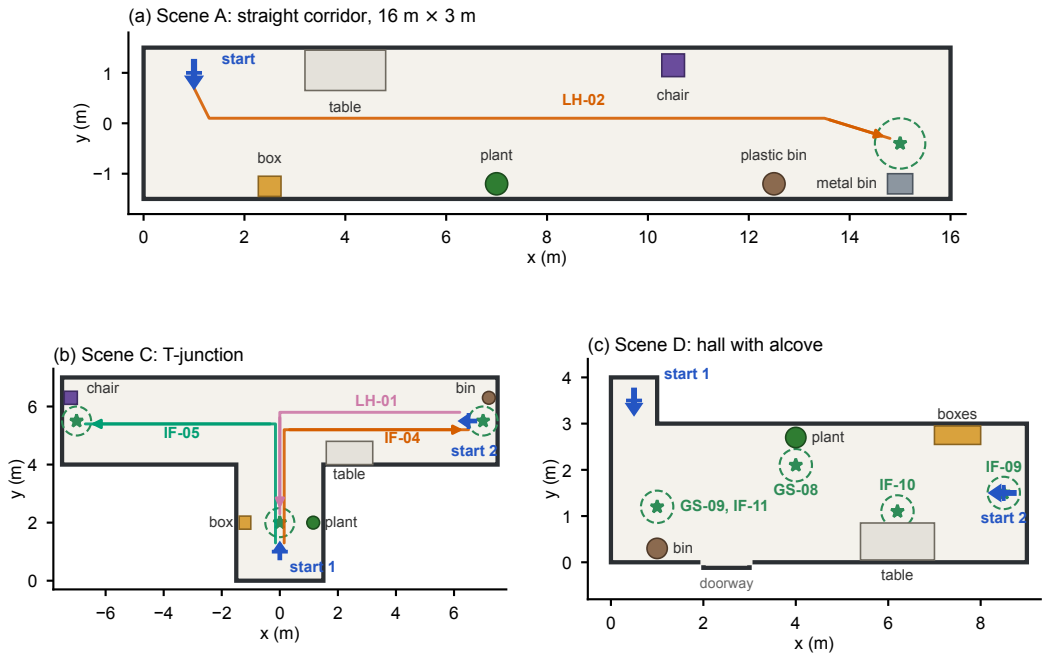


Figure 12: **Real-robot scenes**, top-down, in metres. Blue arrows are start poses; green stars and dashed circles are goals; coloured lines sketch the instructed route. *(a)* Scene A, for LH-02: the robot starts against the north wall facing south, turns left, and must pass the plastic bin to stop at the metal bin. *(b)* Scene C: IF-04 and IF-05 turn right and left at the junction from start 1; LH-01 starts beside the bin and turns into the stem to stop between the plant and the box. *(c)* Scene D: every task except IF-11 starts inside the alcove (start 1); IF-11 runs the hall in the opposite direction from start 2.

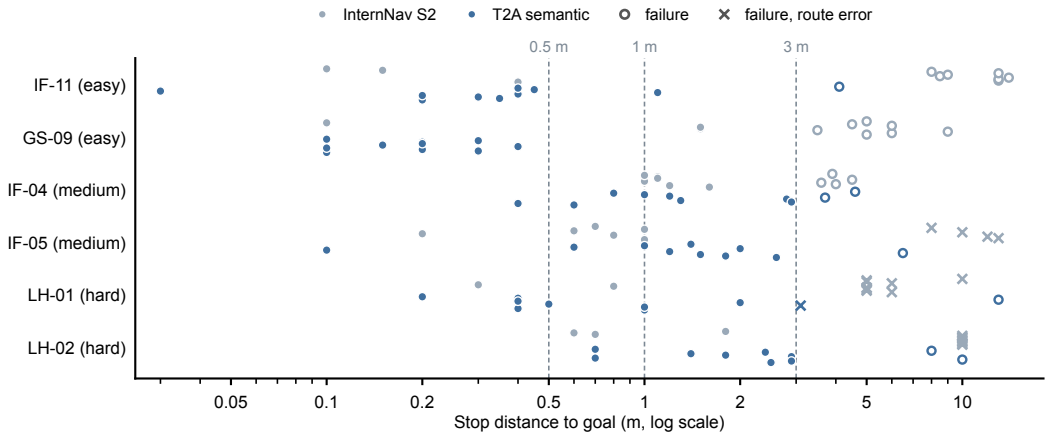


Figure 13: **Stop distance of every trial on the six shared tasks**, log scale. Filled markers are successes at the 3 m radius, open markers failures, and crosses failures with a route error in the field record. Dashed lines mark the 0.5, 1, and 3 m radii.

Table 8: **Real-robot tasks.** Path length is along the instructed route. Challenges: J, junction decision; H, goal hidden at the start; B, goal between two objects; D, same-category distractor.

Task	Scene	Length	Tier (challenges)	Instruction
IF-11	D	7.5 m	easy	Move forward along the room, pass the white table on your left, and stop at the trash bin on your left at the far end.
GS-09	D	2.6 m	easy	Go to the trash bin and stop near it.
IF-04	C	11.5 m	medium (J)	Walk forward to the junction, turn right, and stop at the trash bin at the end of the corridor.
IF-05	C	11.5 m	medium (J)	Walk forward to the junction, turn left, and stop at the chair at the end of the corridor.
LH-01	C	10 m	hard (J,H,B)	Walk forward along the corridor, pass the white table on your left, turn left into the side corridor, and stop between the potted plant on your left and the cardboard box on your right.
LH-02	A	14 m	hard (H,D)	Turn left and walk down the corridor. Pass the white table and the chair on your left, keep going past the round plastic trash bin on your right, and stop next to the rectangular metal trash bin near the end of the corridor.
IF-09	D	9.1 m	medium (H)	Leave the alcove, turn left, walk along the room past the potted plant on your left, and stop at the far end of the room.
IF-10	D	6.9 m	medium (H)	Leave the alcove and turn left. Walk past the potted plant on your left, then stop in front of the white table on your right.
GS-08	D	3.8 m	medium (H)	Walk into the room and turn left, then stop in front of the potted plant.

Table 9: **Real-robot results per task**, successes out of ten trials. Top: the six shared tasks behind Table 4. Bottom: three further medium tasks run with T2A.

		InternNav S2			T2A semantic		
Task	Tier	SR@3	SR@1	SR@0.5	SR@3	SR@1	SR@0.5
IF-11	easy	3	3	3	**9**	**8**	**8**
GS-09	easy	3	1	1	**10**	**10**	**10**
IF-04	medium	6	0	0	**8**	**3**	**1**
IF-05	medium	6	4	1	**9**	2	1
LH-01	hard	2	2	1	**8**	**5**	**4**
LH-02	hard	3	2	0	**8**	2	0
IF-09	medium				10	0	0
IF-10	medium				10	0	0
GS-08	medium				10	9	8

Table 10: **Failed trials on the six shared tasks by mode.** Route error: the field record notes a wrong turn, a missed side corridor, or a lost target followed by a wrong branch. Far and near miss: no route note, and the robot ends at least 5 m, or between 3 and 5 m, from the goal. Interference: a pedestrian crossed the robot's path.

Failure mode	InternNav S2	T2A semantic
Route error	19	1
Far miss ($\geq$ 5 m)	12	3
Near miss (3–5 m)	6	3
Interference	0	1
Total	37	8