SEE WHAT I NEED, DREAM WHAT I WANT: A NAVIGATION FOUNDATION MODEL ON A SELF-CONSTRUCTED REPRESENTATION SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

What visual representation should a navigation policy act on? A central idea of representation learning is that the task's own error should shape the representation. Navigation policies built on vision foundation models largely set this idea aside: the encoders' pretraining objectives decide much of what the vision-language backbone sees. We present RepresentNav, a navigation foundation model built on a *self-constructed representation space*: a visual representation constructed by the policy's own objectives out of the internal structure of vision foundation models. RepresentNav builds this space with Representation Residuals, a compact three-layer transformer that attends over features spanning the full depth of multiple complementary vision foundation models, so that the policy learns to see what it needs for navigation. In the same space it dreams what it wants: before every action a latent chain-of-thought predicts the representation of the scene a moment ahead. The dream is trained end-to-end with the policy as a joint-embedding predictive architecture (JEPA), with a next-representation prediction loss and a regularizer on its targets. RepresentNav sets a new state of the art on VLNVerse, VLN-CE R2R, EVT-Bench, and two point-goal benchmarks, and its gains carry over to a physical robot amid clutter and moving people. On the VLNVerse fine-grained benchmark, RepresentNav achieves a success rate of 83.80%, outperforming the previous state of the art by 20.05 percentage points, and on EVT-Bench distracted tracking it reaches 85.5% with a single camera (+10.1). Analysis shows where they come from. With the same encoders, the self-constructed space clearly outperforms concatenating their outputs and learns to weight their layers unevenly, and a dream in it cuts collisions by a quarter. A single dreamed latent decodes into the depth, semantics, and 3D occupancy of the scene a moment ahead, more accurately than pretrained encoders applied to the real future frame, although occupancy is never supervised, and the rollout tracks the robot's own motion over the next second: the dream emerges as a compact latent world model.

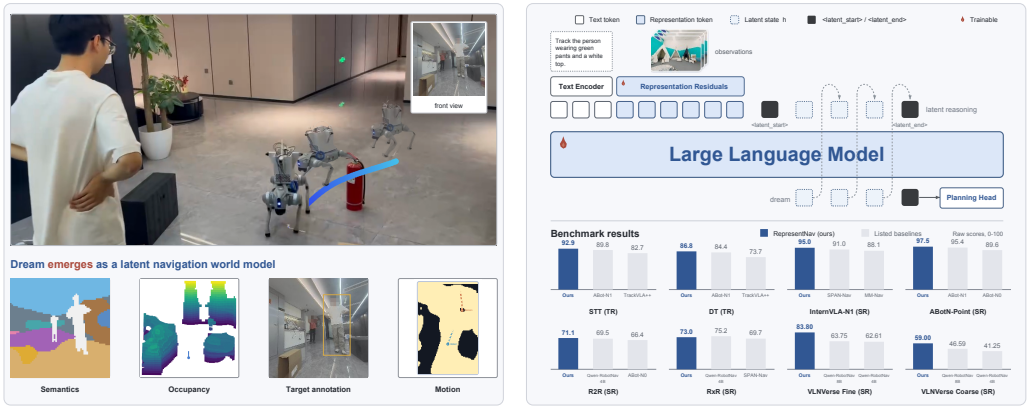


Figure 1: **See what I need, dream what I want.** RepresentNav learns a shared visual interface over frozen visual priors and rolls out latent predictions before generating waypoints. Left: real-world person tracking and post-hoc readouts of semantics, occupancy, and motion. Right: model overview and benchmark results.

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.

1 INTRODUCTION

Consider a robot asked to cross a cluttered hall to a point on its far side. What it needs to see is not every pixel but a representation: a description of the world that makes explicit what the next decision needs and leaves out the rest (Bengio et al., 2013). A person may name three things, how far the nearest surfaces are, what the objects around it are, and how those objects and the robot itself are moving; but what a representation should make explicit is a long-standing question in vision. One tradition answered that a general-purpose description of the scene should come first, built once for any task to read (Marr, 1982). The other answered that perception is purposive: it exists for action and should pick out only what the behavior at hand needs (Gibson, 1979). Deep learning gave the second answer its mechanism: the task's own error, back-propagated into the representation (LeCun et al., 2015).

Vision foundation models have since made the general-purpose tradition practical; navigation, building on them, has largely set the purposive one aside. A vision foundation model is a general-purpose description at a scale no one could have built by hand, but it makes explicit what its own objective needs and by its final layer has left out the rest, including much of the spatial structure it built along the way (Yosinski et al., 2014; Papyan et al., 2020; Raghu et al., 2021); most navigation policies built on vision-language models take that output through an interface fixed before the task exists (Zhang et al., 2026a; 2025b; Cai et al., 2025). Fine-tuning the encoder is only a partial remedy: it adapts one network behind the same output and can erode the general description (Kumar et al., 2022). They follow open-ended instructions and track a person described in words, yet still misjudge free space, collide with obstacles they could describe, and lose a target that turns a corner. The field has more general-purpose visual prior than ever, yet its policies have little say over what they see.

We argue that the two answers need no longer exclude each other in embodied navigation. With vision foundation models a general-purpose description now exists, so the purposive step can be taken on top of it rather than instead of it, and the action gradient is the instrument for taking it. We call the result a *self-constructed representation space*: a representation constructed by the policy's own objectives out of the internal structure of vision foundation models. We present RepresentNav (RepNav), a navigation foundation model that sees and dreams in such a space. It builds the space with Representation Residuals (Fig. 2a), a compact three-layer transformer that attends over layers spanning the full depth of three complementary vision foundation models, semantic, geometric, and dynamic; what it makes explicit is found in training and differs across regions of one view. This is how RepresentNav sees what it needs.

The same question arises in future prediction. Seeing tells the robot where the free space is; crossing the hall requires knowing where it will be a moment later, once the things around it have moved, and an agent that carries a model of the world can anticipate these changes before it acts (Sutton, 1991; Ha & Schmidhuber, 2018; LeCun, 2022). Latent reasoning and navigation world models do this, but in spaces defined by someone else: text, hand-designed spatial tokens, the latent space of a pretrained generative model, or pixels (Hao et al., 2024; Lin et al., 2024; Liu et al., 2026a; Zuo et al., 2026; Bar et al., 2025). A policy that builds its own space for acting has a space of its own to dream in. RepresentNav therefore precedes every action with a latent chain-of-thought (Fig. 2b), trained to predict the representation of the scene 0.4 to 1.6 s ahead in that same space. Its targets are the [CLS] tokens of the future observations, a learnable token that attends alongside the spatial tokens through the three layers of Representation Residuals and gathers each view into one vector; the dream is learned jointly with the policy as a joint-embedding predictive architecture (JEPA), with a next-representation prediction loss and a VISReg regularizer (Wu et al., 2026) on the targets. This is how RepresentNav dreams what it wants: in the space through which it sees.

RepresentNav sets a new state of the art on VLNVerse, VLN-CE R2R, EVT-Bench, and two point-goal benchmarks (Figure 1), including 83.80% success on VLNVerse fine-grained (+20.05) and 85.5% tracking rate on EVT-Bench distracted tracking (+10.1) with a single camera, and the gains carry over to a physical robot (Section 4). Because our claims concern the representation, we then look inside (Section 5) and find:

- **The self-constructed space exploits the internal structure of the priors.** Without a dream, it nearly doubles the gain of concatenating the same three models' outputs, drawing on layers concatenation never sees, such as the dynamic model's layer before its last, and differently across regions of a view (Section 5.1).

- **The dream pays off in the self-constructed space.** Dreaming in the space the policy sees through removes a quarter of the collisions (Section 5.2).
- **One dreamed latent holds a compact world state.** Decoded after the fact, it gives the scene's depth, semantics, and 3D occupancy more accurately than pretrained encoders applied to the real future frame, though occupancy was never a training target, and the rollout gives how the robot and the target move up to 1.2 s ahead (Section 5.3).

In summary, we build a purposive representation on top of a general-purpose one and realize it in RepresentNav with Representation Residuals, a latent chain-of-thought trained as a JEPA in the constructed space, and a two-stage curriculum that first makes the policy act through its dream and then lets it act on what it sees and dreams together.

2 Related Work

Foundation models as the policy's view. Which pretrained representation to hand a policy is a long-standing question in robot learning (Nair et al., 2022; Majumdar et al., 2023; Shang et al., 2024). Since fine-tuning erodes the prior (Kumar et al., 2022), most policies freeze the encoders and fix the interface by hand: the final layer, a fixed set of intermediate layers (Ranftl et al., 2021; Lin et al., 2026; Meng et al., 2024; Bai et al., 2025), or several encoders merged by concatenation, routing, or distillation (Tong et al., 2024; Zong et al., 2024; Ranzinger et al., 2024; Wang et al., 2026). Building on attention residuals (Kimi Team et al., 2026), Representation Residuals construct the interface with the task's own objective instead, over layers spanning the full depth of several vision foundation models.

Predicting before acting. World models let an agent predict before it acts (Ha & Schmidhuber, 2018; Hafner et al., 2020). Navigation has predicted in spaces chosen in advance, pixels, text, occupancy-grounded tokens, or the latent space of a pretrained generative model (Bar et al., 2025; Lin et al., 2024; Gao et al., 2026; Liu et al., 2026a; Zuo et al., 2026), and continuous latent reasoning thinks without words (Hao et al., 2024; Liu et al., 2026c; Chen et al., 2026; Liu et al., 2026b). Joint-embedding predictive architectures learn the space along with the predictor (LeCun, 2022; Assran et al., 2025; Chahe & Zhou, 2026; Zhao et al., 2026) and keep it from collapsing with a moving-average target (Grill et al., 2020) or a regularizer (Bardes et al., 2022; Balestriero & LeCun, 2025; Wu et al., 2026); LeWorldModel trains one end-to-end from pixels, adding only a regularizer (Maes et al., 2026). RepresentNav trains its latent chain-of-thought as one, in the space the policy sees through: the regularizer keeps that space from collapsing, and the task's own objective decides what it holds.

Navigation foundation models. Navigation foundation models unify several navigation tasks on a vision-language backbone (Zhang et al., 2026a; 2025b; Cai et al., 2025; Wang et al., 2025; Zhang et al., 2026b; Chu et al., 2026; AMAP CV Lab, 2026), behind a visual interface fixed before the task exists. RepresentNav keeps this generality and changes what lies beneath it: the space the backbone sees through, and dreams in, is constructed by the task's own objective.

3 Method

RepresentNav plans in a representation space it constructs for itself (Figure 2). Given a language instruction L and RGB observations $I_{1:t}^{1:N}$ from $N \geq 1$ cameras, it predicts eight waypoints $\tau = \{(x_i, y_i, \theta_i)\}_{i=1}^{8}$ in the current robot frame, each a position and a yaw, through one output shared by point-goal navigation, tracking, and vision-and-language navigation (VLN); image and video question answering (QA) is co-trained on the same Qwen3-VL backbone (Bai et al., 2025) and answered autoregressively. Section 3.1 constructs the space with Representation Residuals, which attend over layers spanning the full depth of three frozen vision foundation models (*see what I need*); Section 3.2 rolls out a latent chain-of-thought that predicts the space a moment ahead before the policy acts (*dream what I want*); Section 3.3 trains both in two stages.

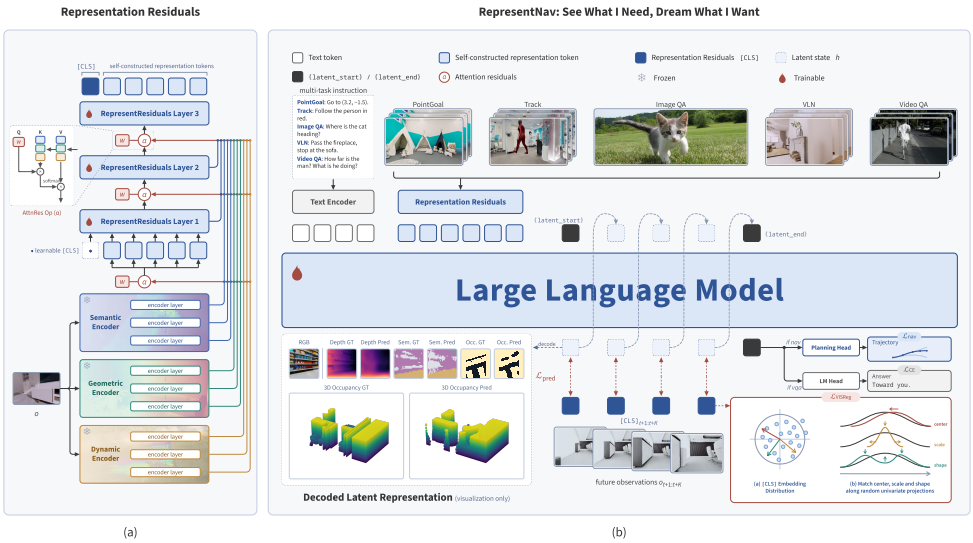


Figure 2: **RepresentNav overview.** (a) Representation Residuals attend over layers spanning the full depth of three frozen vision foundation models and build the self-constructed representation, summarized per view by a learnable [CLS] token. (b) The policy dreams in this space: a latent chain-of-thought predicts the [CLS] tokens of future observations, regularized by VISReg, before the planning head acts; QA shares the backbone, and decoded latents are for visualization only.

3.1 Representation Residuals: See What I Need

Vision foundation models make different things explicit at different depths: the semantic, geometric, and dynamic structure that navigation needs is spread across encoders and across their layers, and the final layer keeps only part of it. Representation Residuals attend dynamically over layers spanning the full depth of several encoders and construct from them a representation of the policy's own; since they are trained with the policy, the task's own objective decides what this space makes explicit.

For each observation $I_{t'}^{v}$ at time t' and view v, DINOv3 (Siméoni et al., 2025) (semantic) and Depth Anything 3 (Lin et al., 2026) (geometric) encode the frame and the video encoder of D4RT (Zhang et al., 2025a) (dynamic), in its open implementation OpenD4RT (RHOS Team, 2026), the same-view clip ending at t', all three frozen; light adapters align features from layers spaced evenly over each encoder's depth to the spatial size of the backbone's native visual input (Appendix A). Representation Residuals are three transformer layers built on attention residuals (Kimi Team et al., 2026). At layer ℓ and position p, with $f_{\ell,0}(p)$ the incoming hidden state and $f_{\ell,j}(p)$, $j = 1, \ldots, J$, the aligned encoder–layer features,

$$\alpha_{\ell,j}(p) = \mathrm{softmax}_j\big[q_\ell^\top \,\mathrm{RMSNorm}\,\big(f_{\ell,j}(p)\big) + b_\ell \mathbf{1}[j=0]\big], \qquad \widetilde{f}_\ell(p) = \sum_{j=0}^{J} \alpha_{\ell,j}(p)\, f_{\ell,j}(p), \quad (1)$$

with a learned query q_ℓ and a bias b_ℓ on the hidden state. The attended spatial tokens and a learnable [CLS] token then pass through residual self-attention and a feed-forward layer. Across the three layers the [CLS] token gathers each view into one vector, the global representation from which Section 3.2 builds its targets. Each view enters the language model as its [CLS] token and a few spatial tokens, finer for the current view and coarser for history, alongside the backbone's own visual tokens (token counts in Appendix A).

3.2 Latent Chain-of-Thought: Dream What I Want

Anticipating the world a moment ahead gives a navigation policy foresight, but what is worth anticipating depends on the task: free space for point-goal navigation, the target's next position for tracking. Prior models choose the prediction target by hand, occupancy in SPAN-Nav (Liu et al., 2026a) and the target's polar position in TrackVLA++ (Liu et al., 2025). RepresentNav predicts in the space it sees through, so the task's own objective decides what the dream holds. At inference,

the rollout is a latent chain-of-thought that gives the planner this foresight before it acts; in training, it is a joint-embedding predictive architecture (JEPA) (LeCun, 2022; Maes et al., 2026) that predicts representations rather than pixels.

Targets. At each future time $t + \Delta_k$, the shared Representation Residuals encode the real observations into per-view [CLS] tokens $c^v_{t+\Delta_k}$, which a learnable projector G maps, concatenated in front–right–back–left order, to the target

$$z_k = G\big(\mathrm{Concat}_{v\in(F,R,B,L)}\, c^v_{t+\Delta_k}\big), \qquad k = 1, \ldots, 4, \tag{2}$$

with $\Delta = (2, 4, 6, 8)$, that is 0.4 to 1.6 s ahead at 5 Hz. Future observations only build targets and never enter the policy's input, and each target uses observations only up to its own time. Because Representation Residuals and G stay trainable, the targets move with the space (dimensions in Appendix A).

Rollout. After the context X of visual tokens, instruction, and time and view identifiers, and a latent-start token, the language model generates four continuous latents, each fed back as the next input embedding, and a projector maps them to the target space:

$$h_k = \mathrm{LLM}(X, h_{<k}), \qquad \widehat{z}_k = \mathrm{Proj}(h_k), \qquad L_{\mathrm{wm}} = \mathrm{MSE}\left(\{\widehat{z}_k\}_{k=1}^4, \{z_k\}_{k=1}^4\right), \tag{3}$$

where LLM returns the hidden state at the last position. The rollout is free-running in training as at inference. The loss averages over target times with available observations, and gradients flow into both the prediction and the target, with no stop-gradient and no moving average, as in LeWorld-Model (Maes et al., 2026). A latent-end token follows the rollout, and the planning head maps its hidden state to the eight waypoints.

Keeping the space from collapsing. Because the target comes from the same Representation Residuals being trained, prediction alone could collapse it. We therefore apply VISReg (Wu et al., 2026) to the targets of each future time, gathered over the global batch into $Z \in \mathbb{R}^{N\times D}$ with mean μ and per-dimension standard deviation σ, and average it over the four times:

$$L_{\mathrm{reg}} = L_{\mathrm{center}} + L_{\mathrm{scale}} + L_{\mathrm{shape}}, \tag{4}$$

$$L_{\mathrm{center}} = \tfrac{1}{D}\|\mu\|_2^2, \quad L_{\mathrm{scale}} = \tfrac{1}{D}\textstyle\sum_{j=1}^{D}(1-\sigma_j)^2, \quad L_{\mathrm{shape}} = \tfrac{1}{NK}\textstyle\sum_{k=1}^{K}\left\|\,\mathrm{sort}(\widetilde{Z}w_k) - \mathbf{q}\right\|_2^2,$$

where $\widetilde{Z} = (Z - \mu)/\,\mathrm{sg}(\sigma)$, w_k are K random unit directions, $q_i = \Phi^{-1}\big(i/(N+1)\big)$ are the quantiles of a standard Gaussian, and the stop-gradient sg keeps the shape term from acting on the scale. The three terms pull the targets toward zero mean, unit variance, and a Gaussian shape along every sampled direction. They fix the scale and shape of the space and nothing else; navigation and prediction decide its content.

3.3 Navigation-Grounded Predictive Curriculum

Training starts from pretrained Qwen3-VL-4B, with Representation Residuals, the [CLS] embedding, the planning head, and both projectors newly initialized, and mixes point-goal, tracking, VLN, and QA data under

$$L = \lambda_{\mathrm{nav}} L_{\mathrm{nav}} + \lambda_{\mathrm{wm}} L_{\mathrm{wm}} + \lambda_{\mathrm{reg}} L_{\mathrm{reg}} + \lambda_{\mathrm{qa}} L_{\mathrm{qa}}, \tag{5}$$

where L_{nav} is the MSE between normalized predicted and expert trajectories and L_{qa} the answer-token cross-entropy. **Stage I** makes the policy act through its dream: following the attention curriculum of LaST-VLA (Luo et al., 2026), the instruction and the observations are isolated from each other, the latents attend to both, and the action position attends only to the instruction and the latents; the loss weights favor prediction. **Stage II** continues from Stage I with full causal attention, so the policy acts on what it sees and what it dreams together, and the weights shift to the trajectory. The training samples are split into two disjoint halves, with one half used in Stage I and the other in Stage II. The three vision foundation models and the backbone's visual encoder stay frozen throughout; loss weights, trainable modules, and the QA path are in Appendix A.

3.4 Training Data

The model is trained on about 5.08M samples (Appendix Figure 6): 4.69M for navigation, namely 2.52M VLN samples from R2R (Anderson et al., 2018) and RxR (Ku et al., 2020), 0.50M fine- and

coarse-grained samples from VLNVerse (Lin et al., 2025), 0.67M tracking samples (Wang et al., 2025), and 1.0M point-goal samples including SAGE-3D (Miao et al., 2025); and 0.39M QA samples from VQA v2 (Goyal et al., 2017) and CLEVR (Johnson et al., 2017). For VLNVerse we follow the training setup of TANGO (Li et al., 2026). All navigation samples share one format: an observation history from one or more cameras, a task instruction or goal, and future waypoints as supervision. The two stages train on disjoint halves of the samples, each drawn evenly from every source and task.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate RepresentNav on vision-and-language navigation, active visual tracking, and point-goal navigation across different environments and observation settings. Section 5 presents component ablations and representation analyses.

Benchmarks and metrics. We evaluate on five benchmarks: VLN-CE (R2R and RxR Val-Unseen) (Krantz et al., 2020; Anderson et al., 2018; Ku et al., 2020) and VLNVerse (Fine- and Coarse-grained, Val-Unseen and Full Test, 3 m success threshold) (Lin et al., 2025) for vision-and-language navigation; EVT-Bench (single-target and distracted tracking) (Wang et al., 2025) for active tracking; and, for point-goal navigation, the InternVLA-N1 protocol (Cai et al., 2025) on InternScenes Home and Commercial (Zhong et al., 2025) and indoor ABotN-PointBench (AMAP CV Lab, 2026), whose success requires collision-free arrival. Metrics follow each benchmark: NE (m), OSR, SR, SPL, and nDTW for VLN; TR, SR, and CR for tracking; SR and SPL for point-goal. Rates are on a 0–100 scale; dashes denote unreported results.

Implementation Details. We train with Muon (learning rate 10^{-3}) for matrix parameters and AdamW (10^{-5}) for the rest, a global batch of 128, and 38,270 steps in total on 16 GPUs. Token counts and the loss weights of each stage are in Appendix A.

4.2 BENCHMARK RESULTS

Active visual tracking. RepresentNav has the highest tracking rate in both observation settings of EVT-Bench (Table 1(a)). With a single camera it reaches 85.5 on distracted tracking, 10.1 points above the best monocular baseline and above every multi-view baseline; with multiple views it reaches 86.8. TrackVLA and TrackVLA++ keep lower collision rates.

Point-goal navigation. On the InternVLA-N1 benchmark, RepresentNav raises mean success to 95.0%, 4.0 points above SPAN-Nav (Table 1(b)). On ABotN-PointBench, where an episode counts only if the robot arrives without a collision, it reaches 97.5% success and 95.8 SPL, 2.1 points above ABot-N1 on both (Table 1(c)).

Vision-and-language navigation. On the VLNVerse test split, RepresentNav reaches 83.80% success with fine-grained instructions and 59.00% with coarse-grained ones, 20.05 and 12.41 points above Qwen-RobotNav-8B; it also leads every reported metric on Val-Unseen (Table 1(e)). Among multi-view methods on VLN-CE, it has the best R2R success, 71.1%, SPL, and navigation error, and ties Qwen-RobotNav-4B on RxR nDTW, 71.9, while Qwen-RobotNav-4B leads the other RxR metrics (Table 1(d)).

4.3 REAL-WORLD RESULTS

We further evaluate RepresentNav in unseen real-world environments across three navigation tasks. As shown in Figure 3, the model avoids obstacles, follows a designated person, and navigates according to language instructions, demonstrating strong cross-environment generalization. Occupancy decoded from the dream latents preserves spatial information relevant to path planning and obstacle avoidance, while bounding-box and semantic readouts reveal the model's perception of targets and scene layout. We also report quantitative real-world evaluations on two robotic platforms, a Go2 quadruped and a Bumi humanoid, in Appendix C.

Table 1: **Benchmark results.** Baselines are from the cited works; **bold**: best per column. (a) EVT-Bench single-target (STT) and distracted (DT) tracking (Liu et al., 2025; Zhang et al., 2026b; Chu et al., 2026; AMAP CV Lab, 2026); best per observation setting. (b) InternVLA-N1 point-goal (Liu et al., 2026a; Xu et al., 2025). (c) Indoor ABotN-PointBench, collision-free success (AMAP CV Lab, 2026). (d) VLN-CE Val-Unseen (Liu et al., 2026a; Wei et al., 2026; Cai et al., 2025; AMAP CV Lab, 2026; Zhang et al., 2026a;b); best per observation setting; *: waypoint-predictor-based; Qwen-RobotNav at 4B, the scale of our backbone. (e) VLNVerse, 3 m threshold, Full Test (Zhang et al., 2026b) and Val-Unseen (Lin et al., 2025; Li et al., 2026); best per split.

(a) EVT-Bench

	STT			DT		
Method	TR↑	SR↑	CR↓	TR↑	SR↑	CR↓
Monocular						
Uni-NaVid	39.5	25.7	41.90	27.4	11.3	43.50
TrackVLA	78.6	85.1	**1.65**	63.2	57.6	5.80
NavFoM	80.5	85.0	–	68.2	61.4	–
TrackVLA++	81.0	86.0	2.10	68.8	66.5	**4.71**
ABot-N0	87.6	86.9	8.54	75.4	66.7	11.60
Qwen-RobotNav-4B	90.0	77.4	6.40	–	–	–
Qwen-RobotNav-8B	89.7	78.6	5.70	–	–	–
RepNav	**92.7**	**87.1**	4.63	**85.5**	**68.3**	11.89
Multi-view						
NavFoM	80.7	88.4	–	67.9	62.0	–
TrackVLA++	82.7	**90.9**	**1.50**	73.7	74.0	**3.51**
ABot-N1	89.8	90.1	4.27	84.4	67.4	17.90
RepNav	**92.9**	90.0	2.85	**86.8**	**75.6**	8.11

(b) InternVLA-N1 point-goal

	Home		Commercial		Mean	
Method	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
InternVLA-N1	60.0	55.6	71.4	68.2	65.7	61.9
NavDP	60.3	54.7	74.1	70.5	67.2	62.6
SIDP	63.2	56.5	81.2	73.4	72.2	65.0
MM-Nav (mono.)	79.9	77.6	86.7	84.7	83.3	81.2
MM-Nav (multi.)	86.3	81.1	89.9	85.5	88.1	83.3
SPAN-Nav	90.9	**85.7**	91.0	88.1	91.0	**86.9**
RepNav	**95.5**	83.4	**94.5**	**89.9**	**95.0**	86.7

(c) ABotN-PointBench

	SR↑			SPL↑		
Method	All	Low	High	All	Low	High
GNM	26.7	30.8	22.5	26.6	30.6	22.5
ViNT	27.9	32.5	23.3	27.9	32.4	23.3
NoMaD	20.0	20.8	19.2	19.6	20.5	18.6
CityWalker	21.7	23.3	20.0	21.6	23.3	19.9
SocialNav	42.5	46.7	38.3	42.5	46.7	38.3
ABot-N0	89.6	91.7	87.5	85.6	87.5	83.7
ABot-N1	95.4	95.0	**95.8**	93.7	93.3	**94.2**
RepNav	**97.5**	**100**	95.0	**95.8**	**97.6**	93.9

(d) VLN-CE Val-Unseen

	R2R				RxR			
Method	NE↓	OSR↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
Monocular								
NaVid	5.72	49.2	41.9	36.5	5.72	45.7	38.2	–
Uni-NaVid	5.58	53.3	47.0	42.7	6.24	48.7	40.9	–
NaVILA	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8
StreamVLN	4.98	64.2	56.9	51.9	6.22	52.9	46.0	61.9
InternVLA-N1	4.83	63.3	58.2	54.0	5.91	53.5	46.1	65.3
Qwen-VLA	5.10	69.0	57.3	51.2	5.80	59.6	47.8	–
DualVLN	4.05	70.7	64.3	58.5	4.58	61.4	51.8	**70.0**
CorrectNav	4.24	67.5	65.1	**62.3**	**4.09**	**69.3**	**63.3**	–
NavForesee	**3.94**	**78.4**	**66.2**	59.7	4.20	66.3	53.2	–
Multi-view								
CMA*	6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0
VLN-BERT*	5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7
AO-Planner	5.55	59.0	47.0	33.0	7.06	43.3	30.5	50.1
Reborn*	5.40	57.0	50.0	46.0	5.98	48.6	42.0	63.3
ETPNav*	4.71	65.0	57.0	49.0	5.64	54.7	44.8	61.9
HNR*	4.42	67.0	61.0	51.0	5.50	56.3	46.7	63.5
NavFoM	4.61	72.1	61.7	55.3	4.74	64.4	56.2	65.8
SPAN-Nav	4.07	75.3	66.3	59.3	4.20	69.7	60.1	67.9
ABot-N0	3.78	70.8	66.4	63.9	3.83	69.3	60.0	–
Qwen-RobotNav-4B	3.80	**77.2**	69.5	63.6	**3.80**	**75.2**	**65.0**	**71.9**
RepNav	**3.75**	73.8	**71.1**	**66.6**	3.85	73.0	64.2	**71.9**

(e) VLNVerse

	Fine-grained				Coarse-grained			
Method	NE↓	OSR↑	SR↑	SPL↑	NE↓	OSR↑	SR↑	SPL↑
Full Test								
InternVLA-N1	5.68	38.69	28.95	25.00	5.71	23.32	17.51	16.54
Uni-NaVid	4.11	67.02	45.74	26.91	5.05	42.70	29.68	13.47
NavFoM	4.13	69.68	51.59	32.40	4.93	45.93	38.02	23.15
Qwen-RobotNav-4B	3.05	70.36	62.61	56.22	4.40	51.03	41.25	37.37
Qwen-RobotNav-8B	3.00	72.81	63.75	57.93	4.08	55.97	46.59	41.54
RepNav	**1.60**	**89.00**	**83.80**	**80.90**	**3.30**	**69.10**	**59.00**	**57.40**
Val-Unseen								
CMA	5.16	62.79	31.15	27.92	–	–	35.52	32.48
Seq2Seq	4.36	49.58	35.03	33.37	–	–	33.09	31.00
HNR	–	–	32.95	29.56	–	–	40.92	37.27
RDP	3.75	71.93	48.60	42.72	–	–	40.85	35.13
GAMA	–	–	38.69	34.38	–	–	48.30	41.50
InternVLA-N1	4.09	64.13	45.56	34.98	–	–	–	–
Uni-NaVid	3.97	62.50	45.00	39.42	–	–	–	–
TANGO	3.72	71.07	52.89	40.18	–	–	–	–
RepNav	**1.42**	**92.73**	**89.21**	**86.50**	–	–	**64.67**	**62.86**

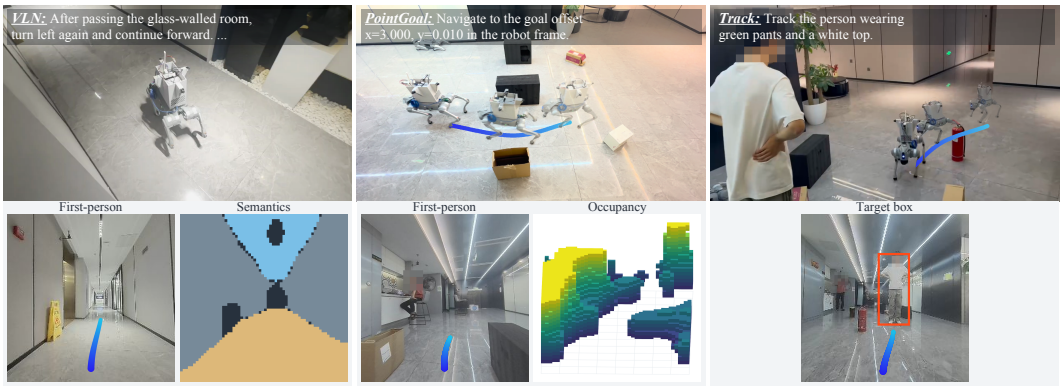


Figure 3: **Real-world navigation and representation readouts.** RepresentNav performs point-goal navigation, person following, and vision-and-language navigation in real environments. Insets show occupancy, target-box, and semantic readouts.

5 Ablation Study and Analysis

This section examines where these gains come from, on the task of Section 1: a robot crossing a cluttered hall, here the InternVLA-N1 PointGoal protocol (Cai et al., 2025) on 4,000 trajectories. Success is near its ceiling there, so the differences show mainly in collisions.

Table 2: **What the policy sees and where it dreams.** Point-goal ablations; no row in (a) dreams. General description: semantic (DINOv3), geometric (DA3), and dynamic (OpenD4RT) vision foundation models. W/o supervision: the latents are rolled out with no loss on them.

General description							
Sem.	Geo.	Dyn.	Interface	Dream target	SR (%)↑	SPL↑	CR (%)↓
(a) What the policy sees (no dream)							
–	–	–	native ViT	none	79.65	73.43	40.80
–	✓	–	concat.	none	83.25	76.08	35.40
✓	✓	–	concat.	none	86.70	79.57	30.90
✓	✓	✓	concat.	none	87.30	80.20	28.70
✓	✓	–	Representation Residuals	none	89.80	82.20	24.30
✓	✓	✓	Representation Residuals	none	93.45	84.28	18.05
(b) What the dream predicts							
✓	✓	✓	Representation Residuals	w/o supervision	93.35	84.34	15.90
✓	✓	✓	Representation Residuals	native, pooled	92.75	82.69	22.00
RepNav (ours)							
✓	✓	✓	Representation Residuals	self-constructed	**95.00**	**86.70**	**13.45**

5.1 The Self-Constructed Space Exploits the Internal Structure of the Priors

The interface shapes what a policy gets from foundation models. Table 2(a) changes only the space the policy sees through, and no row dreams. Three vision foundation models concatenated with the backbone's features lift success from 79.65% to 87.30% and cut collisions from 40.80% to 28.70%; the same models through a space the policy constructs reach 93.45% and 18.05%, nearly twice the gain. The dynamic model adds 0.60 points through the fixed interface but 3.65 through the constructed space, whose attention suggests why (Figure 4; Appendix Figure 8): it takes 67% of the dynamic model's share from L30 and only 30% from the final layer, the one a fixed interface receives, and within one view it leans on the geometric encoder for walls, the semantic one for furniture, and the dynamic one for floor and ceiling. For navigation, it pays to let the policy construct the space it sees through. The same question arises for the dream: whose space should it predict in?

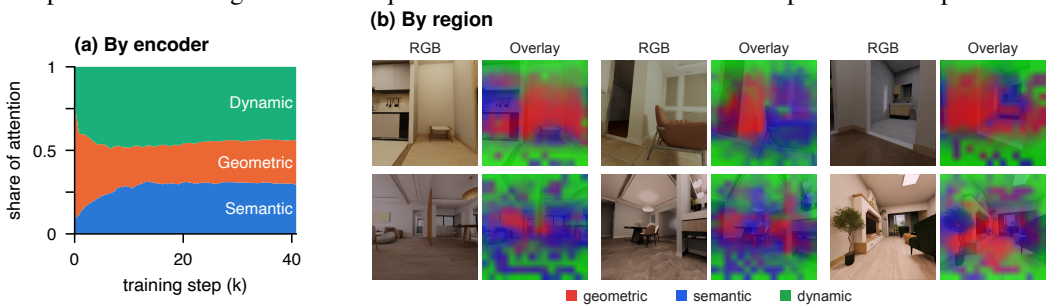


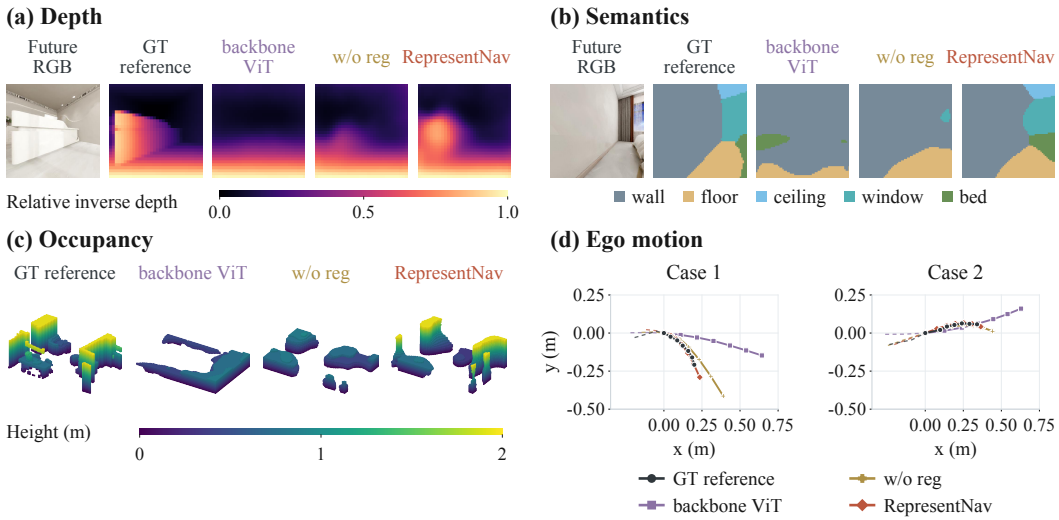
Figure 4: **What the constructed space draws on.** (a) Attention shares of Representation Residuals by encoder during training; the running state, initialized from the geometric encoder, counts as geometric. (b) Where Representation Residuals in the final model prefer each encoder, in six selected VLNVerse views; colors show contrast-enhanced relative preferences, not absolute weights. Details and shares by depth in Appendix B.2.

5.2 The Dream Pays Off in the Self-Constructed Space

The dream does best in the space the policy sees through. Table 2(b) starts from the last row of (a) and changes only what the policy dreams. Without a dream it collides in 18.05% of episodes, and rolling out the same latents without supervision lowers this to 15.90%. A target in a space defined by someone else, the backbone ViT's pooled features, raises it to 22.00%, while one in the self-constructed space lowers it to 13.45%, a quarter fewer than without a dream. Its targets come

from the same trainable Representation Residuals, and VISReg keeps them from collapsing; without it, collisions rise to 16.40% (Appendix B.3). This raises the natural question: what does a dream in this space hold?

5.3 One Dreamed Latent Holds a Compact World State



Representation	Depth MAE↓	Occupancy IoU (%)↑	Semantic mIoU (%)↑	Pixel acc. (%)↑	Ego motion ADE (m)↓
Backbone ViT on real future	0.16257	25.47	16.05	57.32	0.17932
DA3 on real future	0.14472	26.14	15.25	59.99	0.14095
DINOv3 on real future	0.15206	25.85	16.84	58.72	0.14209
openD4RT on real future	0.10253	29.79	20.34	66.67	0.04713
Representation Residuals [CLS] on real future	**0.07520**	**35.95**	**24.82**	**73.45**	**0.02962**
w/o reg. (predicted)	0.10958	30.99	18.83	66.36	0.04583
RepNav (predicted)	0.09131	34.81	23.38	71.35	0.04448

Figure 5: **What one dreamed latent holds.** Post-hoc readouts from the $+0.4$ s latent ((a) depth, (b) semantics, (c) 3D occupancy) and all four latents ((d) ego motion), on 1,000 test observations. Encoder features and the [CLS] tokens of Representation Residuals from real future images are reference inputs, not predicted latents; full results are in Table 7. Examples were selected to show the model's strengths; more in Appendix Figure 9.

One dreamed latent holds a decodable world state. From the latent for $+0.4$ s, decoders recover free space, surface depth, and object layout more accurately than from the pooled features of any of the four pretrained encoders applied to the real future frame, which the latent never sees (Figure 5). Occupancy stands out, 34.81% IoU against at most 29.79%, although no loss supervises it and no encoder outputs it. Across the four latents, decoders also recover the robot's displacement up to $+1.2$ s and, on tracking, where the target is heading (Appendix B.4.1). One latent holds the state and the rollout its change: the dream emerges as a compact latent world model.

6 Conclusion and Limitations

We asked what visual representation a navigation policy should act on, and answered with a self-constructed representation space: a purposive representation built on top of the general-purpose description of vision foundation models. RepresentNav constructs this space with Representation Residuals and dreams in it with a latent chain-of-thought; it sets a new state of the art across vision-and-language navigation, tracking, and point-goal navigation, carries over to a physical robot, and holds in a single dreamed latent a compact world state.

Nothing in this construction is specific to navigation: the priors are general-purpose, and what the space makes explicit is decided by whatever objective the policy is trained on. Yet we have validated it only on navigation; whether it carries over to manipulation, where latent reasoning has already proven useful (Chen et al., 2026; Liu et al., 2026b), dexterous hands, and humanoid whole-body control remains untested.

REFERENCES

AMAP CV Lab. ABot-N1: Toward a general visual language navigation foundation model, 2026. URL https://arxiv.org/abs/2607.10383.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. URL https://arxiv.org/abs/2511.08544.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Wenzhe Cai, Delin Feng, Yu Liu, Jiangmiao Pang, Jiaqi Peng, Chenyang Wan, Hanqing Wang, Liuyi Wang, Tai Wang, Meng Wei, Yuqiang Yang, Xiqian Yu, and Chenming Zhu. InternVLA-N1: An open dual-system vision-language navigation foundation model with learned latent plans. Technical report, Shanghai AI Laboratory, 2025. URL https://internrobotics.github.io/internvla-n1.github.io/.

Amirhosein Chahe and Lifeng Zhou. Policy-guided world model planning for language-conditioned visual navigation. *arXiv preprint arXiv:2603.25981*, 2026. URL https://arxiv.org/abs/2603.25981.

Hao Chen, Jiaming Liu, Zhonghao Yan, Nuowei Han, Renrui Zhang, Chenyang Gu, Jialin Gao, Ziyu Guo, Siyuan Qian, Yinxi Wang, Peng Jia, Shanghang Zhang, and Pheng-Ann Heng. LaST-R1: Reinforcing robotic manipulation via adaptive physical latent reasoning. *arXiv preprint arXiv:2604.28192*, 2026.

Zedong Chu, Shichao Xie, Xiaolong Wu, Yanfen Shen, Minghua Luo, Zhengbo Wang, Fei Liu, Xiaoxu Leng, Junjun Hu, Mingyang Yin, Jia Lu, Yingnan Guo, Kai Yang, Jiawei Han, Xu Chen, Yanqing Zhu, Yuxiang Zhao, Xin Liu, Yirong Yang, Ye He, Jiahang Wang, Yang Cai, Tianlin Zhang, Li Gao, Liu Liu, Mingchao Sun, Fan Jiang, Chiyu Wang, Zhicheng Liu, Hongyu Pan, Honglin Han, Zhining Gu, Kuan Yang, Jianfang Zhang, Di Jing, Zihao Guan, Wei Guo, Guoqing Liu, Di Yang, Xiangpo Yang, Menglin Yang, Hongguang Xing, Weiguo Li, and Mu Xu. ABot-N0: Technical report on the VLA foundation model for versatile embodied navigation, 2026. URL https://arxiv.org/abs/2602.11598.

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. URL https://arxiv.org/abs/2506.09839.

James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6904–6913, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Goyal_Making_the_v_CVPR_2017_paper.html.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024. URL https://arxiv.org/abs/2412.06769.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2901–2910, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Johnson_CLEVR_A_Diagnostic_CVPR_2017_paper.html.

Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, Yutian Chen, Junjie Yan, Ming Wei, Y. Zhang, Fanqing Meng, Chao Hong, Xiaotong Xie, Shaowei Liu, Enzhe Lu, Yunpeng Tai, Yanru Chen, Xin Men, Haiqing Guo, Y. Charles, Haoyu Lu, Lin Sui, Jinguo Zhu, Zaida Zhou, Weiran He, Weixiao Huang, Xinran Xu, Yuzhi Wang, Guokun Lai, Yulun Du, Yuxin Wu, Zhilin Yang, and Xinyu Zhou. Attention residuals, 2026. URL https://arxiv.org/abs/2603.15031.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, 2020.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, 2020.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539.

Anqi Li, Yuxin Chen, Zhaobo Li, Zhuo Cao, Junli Ren, Masayoshi Tomizuka, and Dhruv Shah. TANGO: Humanoid navigation in cluttered environments with a whole-body vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2026. URL https://arxiv.org/abs/2609.09158.

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. *arXiv preprint arXiv:2403.07376*, 2024. URL https://arxiv.org/abs/2403.07376.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yirunib8l8.

Sihao Lin, Zerui Li, Xunyi Zhao, Gengze Zhou, Liuyi Wang, Rong Wei, Rui Tang, Juncheng Li, Hanqing Wang, Jiangmiao Pang, Anton van den Hengel, Jiajun Liu, and Qi Wu. VLNVerse: A benchmark for vision-language navigation with versatile, embodied, realistic simulation and evaluation, 2025. URL https://arxiv.org/abs/2512.19021.

Jiahang Liu, Yunpeng Qi, Jiazhao Zhang, Minghan Li, Shaoan Wang, Kui Wu, Hanjing Ye, Hong Zhang, Zhibo Chen, Fangwei Zhong, Zhizheng Zhang, and He Wang. TrackVLA++: Unleashing reasoning and memory capabilities in VLA models for embodied visual tracking, 2025. URL https://arxiv.org/abs/2510.07134.

Jiahang Liu, Tianyu Xu, Jiawei Chen, Lu Yue, Jiazhao Zhang, Zhiyong Wang, Minghan Li, Qisheng Zhao, Anqi Li, Qi Su, Zhizheng Zhang, and He Wang. Span-nav: Generalized spatial awareness for versatile vision-language navigation. *arXiv preprint arXiv:2603.09163*, 2026a. URL https://arxiv.org/abs/2603.09163.

Jiaming Liu, Yinxi Wang, Chenyang Gu, Siyuan Qian, Xiangju Mi, Hao Chen, Jiawei Chen, Qingpo Wuwu, Xiaoqi Li, Nuowei Han, Yiming Zhang, Xuheng Zhang, Yang Yue, Yeqing Yang, Lei Wang, Peng Jia, Hao Tang, and Shanghang Zhang. LaST-HD: Learning latent physical reasoning from scalable human data for robot manipulation. *arXiv preprint arXiv:2606.23685*, 2026b.

Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, Zhengping Che, Jian Tang, Pheng-Ann Heng, and Shanghang Zhang. LaST_0: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. *arXiv preprint arXiv:2601.05248*, 2026c. URL https://arxiv.org/abs/2601.05248.

Yuechen Luo, Fang Li, Shaoqing Xu, Yang Ji, Zehan Zhang, Bing Wang, Yuannan Shen, Jianwei Cui, Long Chen, Guang Chen, Hangjun Ye, Zhi-Xin Yang, and Fuxi Wen. LaST-VLA: Thinking in latent spatio-temporal space for vision-language-action in autonomous driving, 2026. URL https://arxiv.org/abs/2603.01928.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorld-Model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems*, volume 36, 2023.

David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1982.

Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. DeepStack: Deeply stacking visual tokens is surprisingly simple and effective for LMMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Bingchen Miao, Rong Wei, Zhiqi Ge, Xiaoquan Sun, Shiqi Gao, Jingzhe Zhu, Renhan Wang, Siliang Tang, Jun Xiao, Rui Tang, and Juncheng Li. Towards physically executable 3D gaussian for embodied navigation, 2025. URL https://arxiv.org/abs/2510.21307.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, 2022.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Advances in Neural Information Processing Systems*, volume 34, 2021.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12490–12500, 2024.

RHOS Team. OpenD4RT: An unofficial PyTorch implementation of D4RT. `https://github.com/Lijiaxin0111/Open-d4rt`, 2026.

Jinghuan Shang, Karl Schmeckpeper, Brandon B. May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. In *Conference on Robot Learning*, 2024. URL `https://arxiv.org/abs/2407.20179`.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. TrackVLA: Embodied visual tracking in the wild. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4139–4164. PMLR, 2025. URL `https://proceedings.mlr.press/v305/wang25f.html`.

Yixiao Wang, Mingxiao Huo, Zhixuan Liang, Yushi Du, Lingfeng Sun, Haotian Lin, Jinghuan Shang, Chensheng Peng, Mohit Bansal, Mingyu Ding, and Masayoshi Tomizuka. Ver: Vision expert transformer for robot learning via foundation distillation and dynamic routing. In *International Conference on Learning Representations*, 2026. URL `https://arxiv.org/abs/2510.05213`.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. In *International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=GK4rznYwhn`.

Haiyu Wu, Randall Balestriero, and Morgan Levine. VISReg: Variance-invariance-sketching regularization for JEPA training. *arXiv preprint arXiv:2606.02572*, 2026.

Tianyu Xu, Jiawei Chen, Jiazhao Zhang, Wenyao Zhang, Zekun Qi, Minghan Li, Jiahang Liu, Lu Yue, Zhizheng Zhang, and He Wang. MM-Nav: Multi-view VLA model for robust visual navigation via multi-expert learning, 2025. URL `https://arxiv.org/abs/2510.03142v2`. Version 2, revised June 2026.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one D4RT at a time. *arXiv preprint arXiv:2512.08924*, 2025a.

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. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.013. URL https://www.roboticsproceedings.org/rss21/p013.html.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, Yuxin Fan, Wenjun Li, Zhibo Chen, Fei Gao, Qi Wu, Zhizheng Zhang, and He Wang. Embodied navigation foundation model. In *International Conference on Learning Representations*, 2026a. URL https://arxiv.org/abs/2509.12129.

Jiazhao Zhang, Gengze Zhou, Hale Yin, Yiyang Huang, Zixing Lei, Qihang Peng, Haoqi Yuan, Jie Zhang, Xudong Guo, Xiaoyue Chen, An Yang, Fei Huang, Zhibo Yang, Junyang Lin, Dayiheng Liu, Jingren Zhou, Zhuoyuan Yu, Jingyang Fan, Zhixuan Liang, Pei Lin, Ye Wang, Haoyang Li, Anzhe Chen, Kun Yan, Xiao Xu, Jiahao Li, Lulu Hu, Minying Zhang, Shurui Li, Wenhu Xiao, Shuai Bai, Xuancheng Ren, Chenxu Lv, Chenfei Wu, and Xiong-Hui Chen. Qwen-RobotNav technical report: A scalable navigation model designed for an agentic navigation system, 2026b. URL https://arxiv.org/abs/2606.18112v3.

Han Zhao, Jingbo Wang, Wenxuan Song, Shuai Chen, Yang Liu, Yan Wang, Haoang Li, and Donglin Wang. Frappe: Infusing world modeling into generalist policies via multiple future representation alignment. *arXiv preprint arXiv:2602.17259*, 2026. URL https://arxiv.org/abs/2602.17259.

Weipeng Zhong, Peizhou Cao, Yichen Jin, Li Luo, Wenzhe Cai, Jingli Lin, Hanqing Wang, Zhaoyang Lyu, Tai Wang, Bo Dai, Xudong Xu, and Jiangmiao Pang. InternScenes: A large-scale simulatable indoor scene dataset with realistic layouts, 2025. URL https://arxiv.org/abs/2509.10813.

Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. MoVA: Adapting mixture of vision experts to multimodal context. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jing Zuo, Lingzhou Mu, Fan Jiang, Chengcheng Ma, Mu Xu, and Yonggang Qi. Fantasyvln: Unified multimodal chain-of-thought reasoning for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29674–29683, 2026. URL https://arxiv.org/abs/2601.13976.

A Method Details

A.1 Architecture

Encoders and adapters. For each navigation observation at time t' and view v, DINOv3 and DA3 process the single frame $I^v_{t'}$, while OpenD4RT processes the same-view four-frame clip $I^v_{t'-3:t'}$. Features from several layers of each encoder are aligned to a 16×16 grid and a common channel dimension by independent adapters, followed by source, layer, and position embeddings.

Token readouts. Current and historical observations share Representation Residuals but use different spatial readouts. For each historical view, spatial features are average-pooled over a 2×4 partition and projected into eight coarse tokens. For each current view, nonoverlapping 2×2 neighborhoods are concatenated along channels and projected into an 8×8 grid of 64 fine tokens. The normalized [CLS] of each view is a 1024-dimensional global representation; an independent MLP maps it to the language-model dimension for input alongside the spatial tokens. Each current view therefore contributes [CLS] plus 64 fine tokens, and each historical view [CLS] plus eight coarse tokens.

Targets and projectors. Each [CLS] has 1024 dimensions; the front–right–back–left concatenation yields 4096 dimensions, which the target projector G maps to 256, with missing views filled with zeros. The prediction head Proj maps each 2,560-dimensional hidden state to the same 256 dimensions.

Regularizer computation. For each future time, target representations are gathered across the global batch and VISReg is computed separately, then averaged across times.

A.2 Training

Stage I: learning navigation-relevant future representations. The language backbone starts from pretrained Qwen3-VL-4B weights; Representation Residuals, the [CLS] embedding, the planner, and the target and prediction projectors G and Proj are newly initialized. The model uses the full visual input, performs four-step latent CoT, and constructs future targets online. Following the attention curriculum of LaST-VLA (Luo et al., 2026), we isolate the visual and language prefixes in the navigation sequence's causal mask. Latents retain access to both, while the action position attends only to language and generated latents. We set $(\lambda_{\mathrm{nav}}, \lambda_{\mathrm{wm}}, \lambda_{\mathrm{reg}}, \lambda_{\mathrm{qa}}) = (0.01, 0.1, 0.9, 1)$, emphasizing future representation learning while retaining trajectory supervision; the QA weight of 1 in both stages offsets QA samples making up only a tenth of the training mixture.

Stage II: planning from observations and predictions. We inherit the complete Stage-I model and optimization state, retain four-step latent CoT, and restore standard causal attention so that the action position can jointly read visual observations, instructions, and predicted latents. Loss weights become $(\lambda_{\mathrm{nav}}, \lambda_{\mathrm{wm}}, \lambda_{\mathrm{reg}}, \lambda_{\mathrm{qa}}) = (1, 0.001, 0.009, 1)$, emphasizing trajectory learning while retaining future prediction and target-space regularization.

Trainable modules and QA. Both stages update the language backbone, Representation Residuals and the learnable [CLS] embedding, the planner, and both projectors. The three source visual encoders and the Qwen3-VL visual encoder remain frozen. QA samples use the backbone's visual input and features from DA3 and DINOv3 to generate answers autoregressively, without navigation latent rollout or future-target construction.

Data split. The sample manifest is split once, with a fixed seed and stratified by source and task, into the two halves used by the two stages. The halves are disjoint in samples but not in episodes: the split is by sample, so one episode can contribute samples to both.

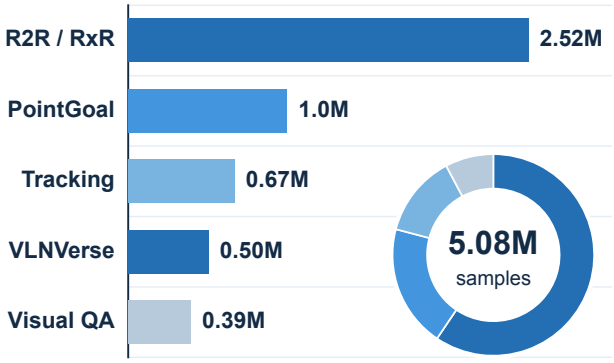


Figure 6: **Training data composition** (Section 3.4).

A.3 The Two-Stage Curriculum

Table 3 compares the curriculum of Section 3.3 with a single stage that opens attention from the start.

The policy acts through its dream. The two-stage policy relies more on the latent chain-of-thought despite similar occupancy readouts: 34.81% IoU versus 34.57% for the single-stage model

Table 3: **Effect of the two-stage curriculum.** Same model and data; the single-stage variant opens attention from the start instead of first restricting the action position to the latent chain-of-thought.

Restricted stage	Open stage	SR (%) ↑	SPL ↑	CR (%) ↓
–	✓	92.40	83.68	15.15
✓	✓	**95.00**	**86.70**	**13.45**

(Tables 7 and 4). We intervene on the latent inputs with full causal attention in both models, keeping images, history, instructions, and the action-readout input unchanged. Cross-scene swaps shift the two-stage model's predicted trajectory by 4.95 cm on average, versus 1.38 cm for the single-stage model; zeroing produces shifts of 8.24 cm and 2.68 cm (Table 4). Two-stage shifts are larger in 183 and 182 of the 200 observations, respectively, rather than being concentrated in a few cases. The latent chain-of-thought remains involved in action generation after full visual context is restored; disrupting its inputs also increases the two-stage model's ADE against the reference trajectory from 4.38 cm to 7.06 cm and 9.76 cm. Figure 7 shows examples and the same shifts for other dream targets and regularizers.

Table 4: **The two-stage policy responds more strongly to changes in its latent inputs.** Each model and condition uses 200 observations.

Training	Inference condition	Affected (%)	Shift (cm)	GT ADE (cm)
Single-stage	Normal	0.0	0.00	3.93
Single-stage	Cross-scene swap	0.0	1.38	4.32
Single-stage	Zero inputs	2.0	2.68	4.98
Two-stage	Normal	0.0	0.00	4.38
Two-stage	Cross-scene swap	27.0	4.95	7.06
Two-stage	Zero inputs	33.0	8.24	9.76

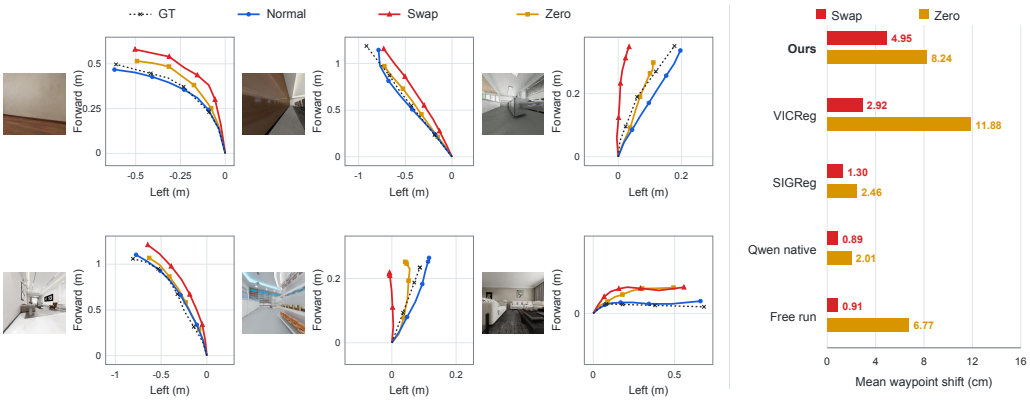


Figure 7: **Latent-input interventions and action sensitivity.** Left: GT and normal, cross-scene-swap, and zero-input predictions from our two-stage point-goal policy; axes are left and forward (meters), with equal XY scales. Right: mean XY change from each policy's normal prediction over eight waypoints on the same 200 observations. All five models use their own predicted latents; Qwen native dreams in the backbone ViT's pooled features and Free run rolls out the same latents with no loss on them (w/o supervision in Table 2(b)), while VICReg and SIGReg replace VISReg (Table 6). Only the four recurrent inputs are swapped within each checkpoint or zeroed; other inputs stay fixed, with no future images. These open-loop diagnostics on navigation-training observations measure sensitivity, not navigation performance.

Protocol. Observations come from 200 navigation-training trajectories in 155 scenes. Fixed single-stage and two-stage models keep images, history, instructions, and other inference settings unchanged; both use full causal attention. Interventions modify the four latent inputs, including the initial hidden state h_0; the action-readout input is unchanged. Cross-scene swaps use corresponding inputs from another scene under the same model. Self-input replay matches normal inference bit-for-bit for all 400 model–observation pairs. An action is affected when the first waypoint changes by at least 2 cm in XY or 2° in yaw; 0% does not mean unchanged outputs. Shift is displacement from normal outputs; ADE is error against the reference trajectory. Both average over eight

waypoints from $+0.2$ to $+1.2$ s. These are open-loop action responses, not closed-loop success or collision rates under intervention. The training schedules differ in both information pathways and loss weights.

B Supplementary Analyses

B.1 Protocols

Table 2. All source encoders are frozen; the language model, Representation Residuals where present, and the action head are trained. No row in (a) dreams; every row that dreams uses the free-running rollout of the full model, and each row differs from the full model only in the factors marked. In (b), the w/o-supervision row keeps the rollout but applies neither the prediction loss nor VISReg to the latents, and the native-pooled row predicts the backbone ViT's pooled features without regularization. SPL is on a 0–100 scale.

Readout decoders. Readouts use lightweight decoders, $\widehat{y}_r = D_r(h_1, \ldots, h_4)$, trained after the fact on frozen latents to recover geometry, semantics, target states, and motion. They serve only to analyze the representation, and each comparison trains the same decoder on every representation. Figures label the backbone ViT baseline "Qwen native" and the w/o-supervision model "Free run".

B.2 The Self-Constructed Space (Section 5.1)

Attention shares. Figure 8 plots the weights α of Eq. 1 logged during training, averaged over the three layers of Representation Residuals and over positions. The candidates are features from a subset of layers spaced along each encoder's depth, together with the running hidden state, which is initialized from the geometric encoder and therefore counted as geometric. Curves are smoothed with a moving average over 121 logged points, about 1,200 training steps. Table 5 varies the number of layers of Representation Residuals.

Preferences by region. Figure 4(b) uses the attention weights of Representation Residuals in the final model. Each encoder's weights are summed over its layers, divided by that encoder's spatial mean over the four current views, raised to the fourth power, and normalized across encoders; the running state is excluded. Colors therefore show contrast-enhanced relative preferences, not absolute weights or causal contributions.

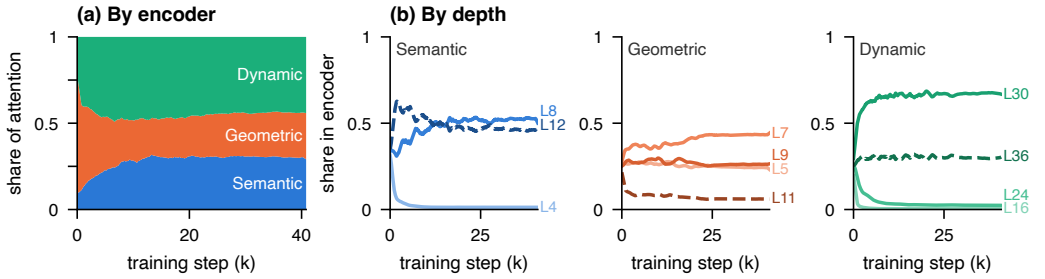


Figure 8: **Attention shares by encoder and by depth.** Attention shares of Representation Residuals during training over a subset of layers spaced along each encoder's depth, averaged over its three layers and all positions. (a) By encoder; the running state, initialized from the geometric encoder, counts as geometric. (b) By depth within each encoder; dashed lines mark final layers.

Table 5: **Capacity of Representation Residuals is not the source of the gain.** Point-goal navigation without the dream, all three encoders, varying the number of Representation Residuals layers. The default uses three layers (SR 93.45%, SPL 84.2%, CR 18.05%); all variants lie within 2.2 points of one another in SR and SPL, so the gain in Table 2 is not a matter of capacity.

Representation Residuals layers	1	2	4	8	16
SR (%)↑	92.75	91.15	92.25	91.70	93.30
SPL (%)↑	84.1	82.6	83.0	82.6	84.1
CR (%)↓	18.40	18.25	17.25	16.10	16.40

B.3 The Target of the Dream (Section 5.2)

The regularizer keeps the space from collapsing. Trained on prediction alone, a joint-embedding predictive architecture collapses, mapping every input to nearly the same representation, which is why it is paired with an anti-collapse regularizer (Maes et al., 2026; Balestriero & LeCun, 2025). Our target is no exception: the same Representation Residuals that navigation trains produce it, so it moves as they learn. VISReg constrains only the scale and shape of its distribution (Wu et al., 2026). Without it, success falls from 95.00% to 92.90% and collisions rise from 13.45% to 16.40%, worse than rolling out the latents with no supervision at all (93.35% and 15.90%, Table 2(b)); SIGReg or VICReg in its place recovers less (Table 6).

Table 6: **Choice of the target regularizer.** Same model and data; only the regularizer on the dream targets changes; None keeps the prediction loss and drops only the regularizer.

Regularizer	SR (%)↑	SPL↑	CR (%)↓
None	92.90	84.09	16.40
SIGReg	93.85	84.39	17.15
VICReg	93.25	83.98	16.40
VISReg (ours)	**95.00**	**86.70**	**13.45**

B.4 What the Dream Holds (Section 5.3)

Table 7 gives the full readout comparison behind Figure 5, and Figure 9 expands its examples.

Protocol. Depth targets are ground-truth depth maps, expressed as normalized relative inverse depth rather than metric depth; occupancy uses native voxel labels. Ego-motion ADE measures XY error at six future positions from $+0.2$ to $+1.2$ s in the current robot frame. The four pretrained-encoder rows and the Representation Residuals [CLS] row encode the real future observation, for openD4RT the four-frame clip ending at it; they are references rather than controls, since they see the frame the latent must predict while the latent sees only the policy's multi-view history. The rows that share the latent's inputs are w/o supervision, w/o regularization, SIGReg, VICReg, and single-stage. RepresentNav uses one fixed two-stage model and raw LLM hidden states. Semantic references are OneFormer-20 pseudo-labels. The test observations span 48 scenes; metrics accumulate over pixels in four 64×64 views. Each readout uses one vector per observation: globally pooled pretrained-encoder features, the pre-head [CLS] of Representation Residuals, or one raw LLM latent. LLM latents and backbone ViT features have 2,560 dimensions, and every input vector receives 16 additional static calibration values per camera. Configurations share the semantic decoder, training budget, and validation-selection protocol; results use validation-selected three-seed probability ensembles. Paired scene bootstrap does not establish a clear ordering between single-stage and two-stage semantic readouts.

Table 7: **Full readout comparison**, including the alternative regularizers and the single-stage model. Each metric uses 1,000 test observations.

Representation	Depth MAE↓	Occupancy IoU (%)↑	Semantic mIoU (%)↑	Pixel acc. (%)↑	Ego motion ADE (m)↓
Backbone ViT on real future	0.16257	25.47	16.05	57.32	0.17932
DA3 on real future	0.14472	26.14	15.25	59.99	0.14095
DINOv3 on real future	0.15206	25.85	16.84	58.72	0.14209
openD4RT on real future	0.10253	29.79	20.34	66.67	0.04713
Representation Residuals [CLS] on real future	**0.07520**	**35.95**	**24.82**	**73.45**	**0.02962**
w/o supervision	0.10680	31.37	18.50	67.39	0.05149
w/o regularization	0.10958	30.99	18.83	66.36	0.04583
SIGReg	0.10307	32.40	21.22	69.21	0.04539
VICReg	0.10233	31.82	22.27	69.00	0.04687
RepNav, single-stage	0.09449	34.57	23.64	71.53	0.04635
RepNav	0.09131	34.81	23.38	71.35	0.04448

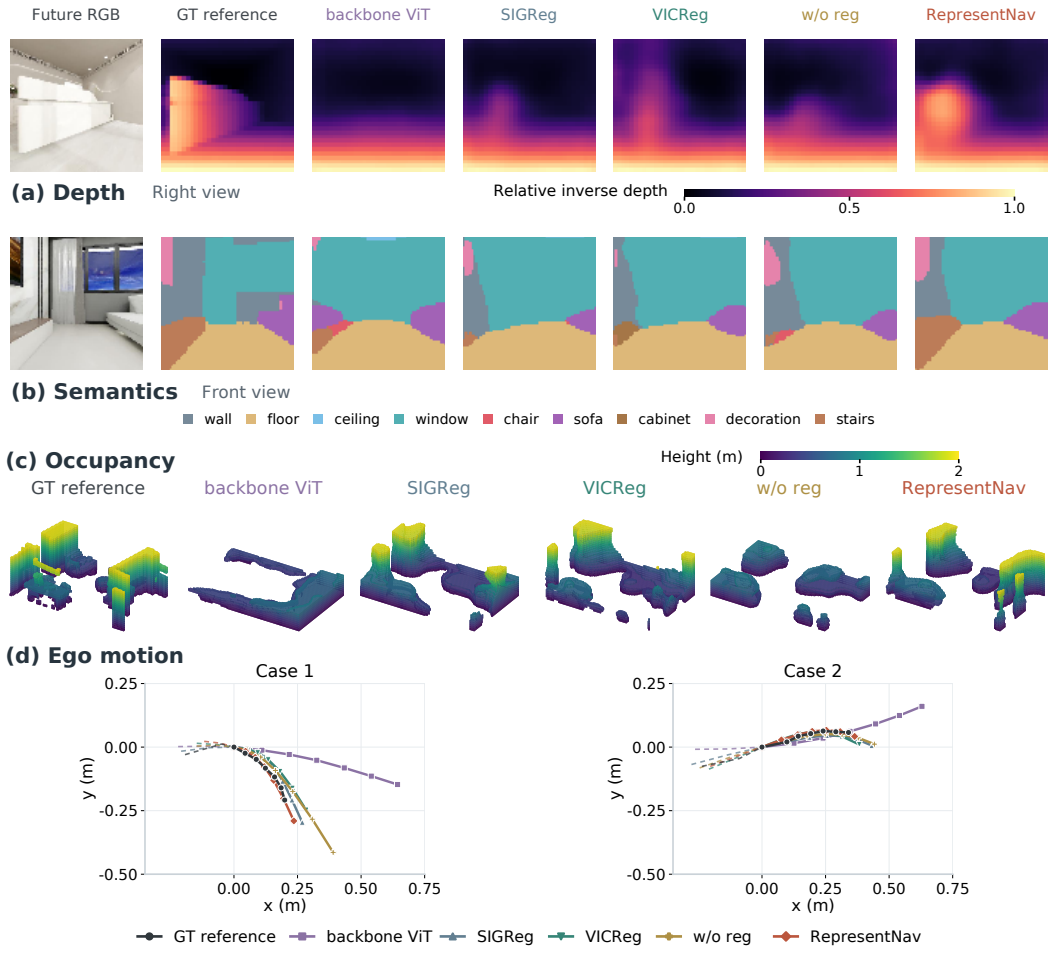


Figure 9: **Future scenes and ego motion can be read from the dream.** Expanded comparison for Figure 5, using the same observations and selected camera views. (a) The camera depth uses original 32×32 relative inverse-depth maps and GT references on a common color scale. (b) The camera semantics uses original 64×64 labels and GT references with a common class palette. Both target $+0.4$ s; future RGB provides visual context. (c) Occupancy uses native $120 \times 80 \times 20$ voxels with identical viewpoints, spatial bounds, and height colors. (d) Ego motion shows six future positions with shared ranges and equal axis scales; dashed and solid curves denote past and future motion. GT sources and full-test metrics are specified in Table 7. Groups use different observations, selected post hoc to illustrate RepresentNav's strengths.

B.4.1 Target Localization and Motion on Track

RepresentNav preserves target location and future motion more accurately on Track than the backbone ViT on the real future images and a two-stage SIGReg model (Table 8). Current-front box IoU increases from 33.69% to 59.33% to 72.81%, while future person-motion ADE falls from 1.09701 m to 0.78507 m to 0.44450 m. The improvement extends to ego motion, whose future ADE falls from 0.33404 m to 0.22126 m to 0.15455 m. Figures 10–12 visualize matched Track observations: RepresentNav recovers the target boxes and follows the reference motion more closely in these examples.

Table 8: **RepresentNav improves target localization and motion readouts on the expanded Track test set.** All three models are evaluated on the same 5,268 observations.

	Target localization (%)			Person motion (m)		Ego motion (m)	
Model	Front IoU↑	All IoU↑	AP50↑	ADE↓	FDE↓	ADE↓	FDE↓
Backbone ViT on real future	33.69	32.74	8.67	1.09701	1.27292	0.33404	0.54533
SIGReg (two-stage)	59.33	50.04	36.56	0.78507	0.90616	0.22126	0.37249
RepNav (ours)	**72.81**	**61.66**	**69.44**	**0.44450**	**0.55498**	**0.15455**	**0.25190**

Protocol. Frozen-model readout evaluation on Track benchmark samples, not closed-loop navigation. The expanded set contains 5,268 observations from eight decoder-test scenes, including additional episode/time anchors and native-camera variants; these are not 5,268 independent trajectories. Front IoU uses GT-visible targets in the current front view. All IoU uses GT-visible targets across four views and nine times from -0.4 to $+1.2$ s; AP50 also accounts for false positives and missed detections. ADE uses future offsets $+0.2$ to $+1.2$ s; FDE uses $+1.2$ s. Motion is measured in the fixed current-robot frame. The backbone ViT row encodes the real future images as a reference. RepresentNav uses one fixed two-stage checkpoint. Training configurations differ beyond the regularizer, so this is not an isolated regularizer ablation.

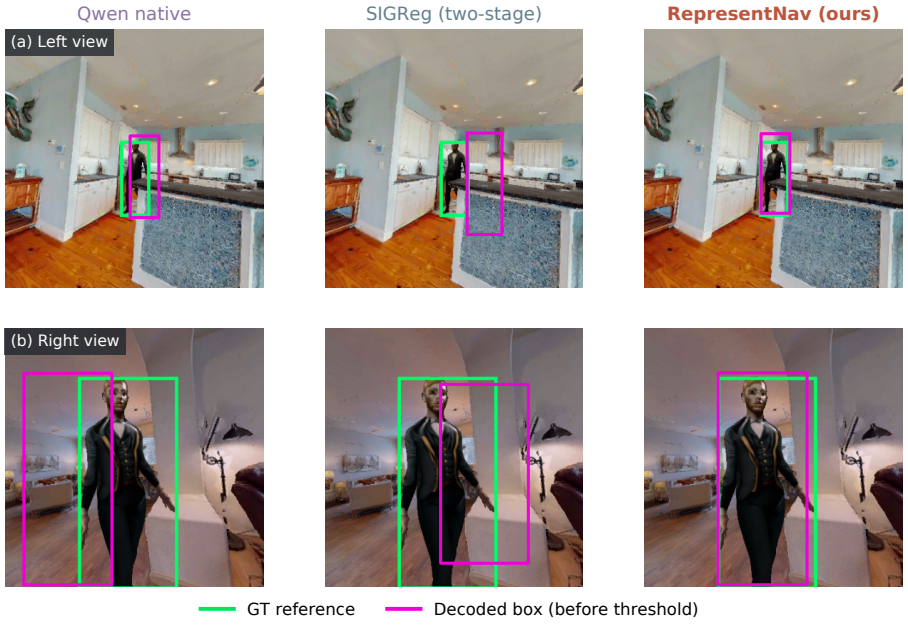


Figure 10: **RepresentNav localizes the target more accurately than the baselines.** Frozen-model bounding-box readouts on two expanded Track test observations. Each row shares the current-view RGB image and GT across models. Green: GT reference. Magenta: original decoded box coordinates before visibility-confidence filtering, shown for all models to compare localization on GT-visible targets, not thresholded detections. RepresentNav's boxes lie closer to GT; the baselines are displaced. Examples are selected post hoc; aggregate results are in Table 8.

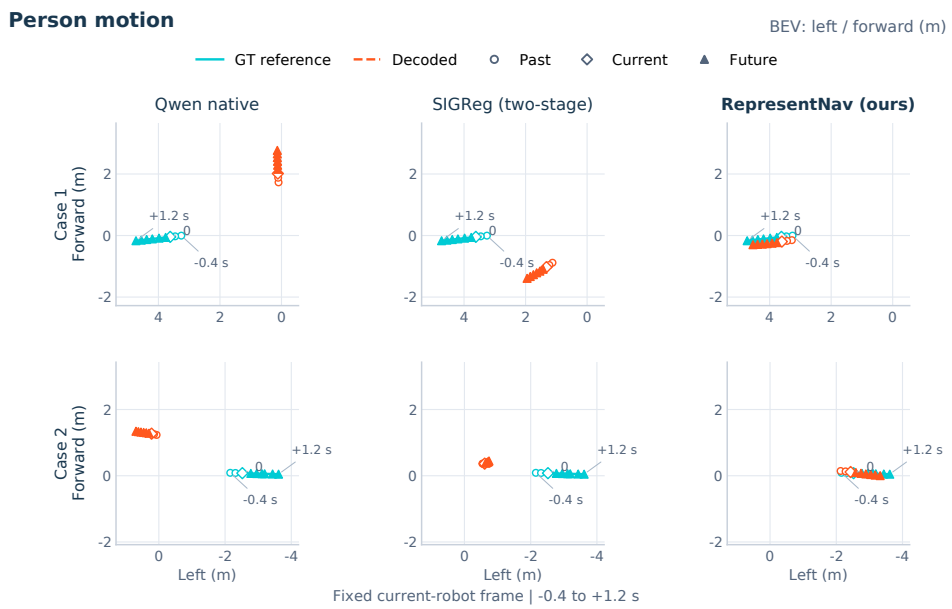


Figure 11: **RepresentNav recovers the target's position and subsequent motion.** Frozen-model readouts on two observations from the expanded Track test set. Each column compares one model with GT: cyan solid curves are GT references; orange dashed curves are readouts. Circles, diamonds, and triangles mark past, current, and future positions. Each row shares metric bounds across models, with left horizontal and forward vertical in the fixed current-robot frame. Examples are selected post hoc; aggregate results and model conventions are in Table 8.

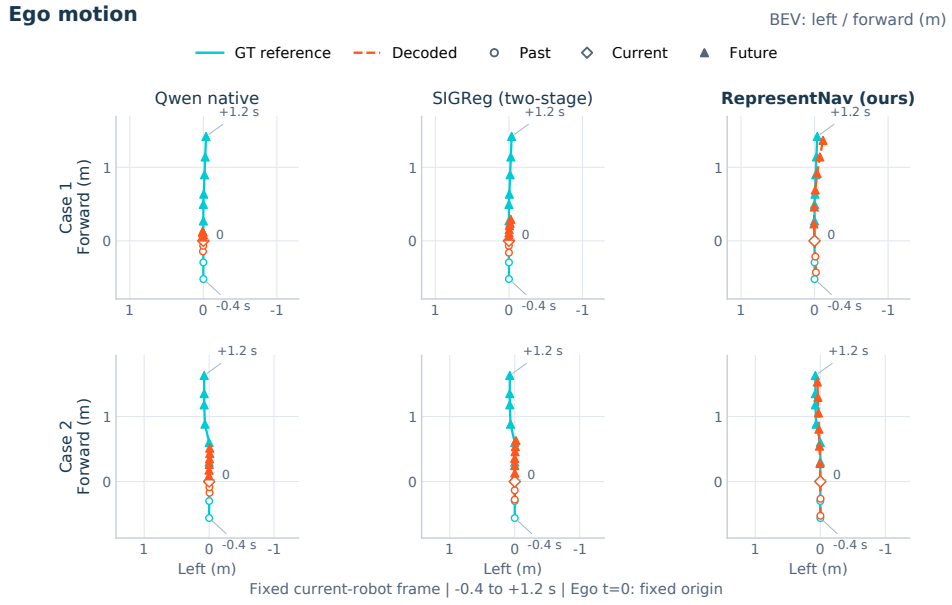


Figure 12: **RepresentNav recovers future ego displacement underestimated by the baselines.** Frozen-model readouts on two observations from the expanded Track test set. Colors, markers, and BEV axes follow Figure 11; each row shares metric bounds across models. Ego $t = 0$ is fixed at the origin. RepresentNav follows the GT displacement, while the backbone ViT and SIGReg predict shorter travel. Examples are selected post hoc. These curves are readouts, not executed trajectories; evaluation conventions are in Table 8.

C QUALITATIVE RESULTS

Figure 13 shows episodes from the benchmarks of Section 4, and Figure 15 real-world episodes across the three tasks.

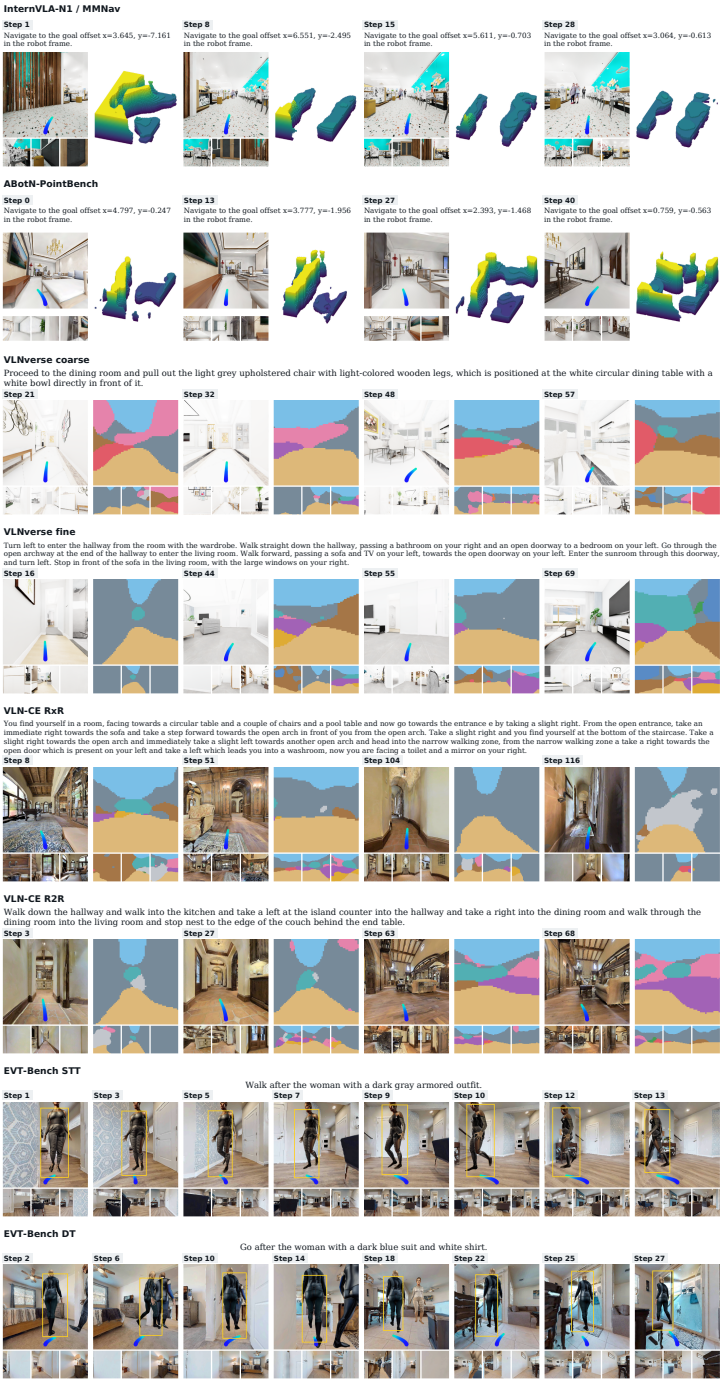


Figure 13: **Visualization of performance on benchmarks.**

Real-world performance on 40 test cases. We evaluate RepresentNav on two robotic platforms, a Go2 quadruped and a Bumi humanoid. Our quantitative evaluation comprises 20 VLN and 20 tracking test cases. VLN requires following natural-language instructions to reach specified destinations, while tracking requires following a designated moving target. Figure 16 presents third-person demonstrations and a quantitative comparison with Uni-NaVid. RepresentNav completes 14 of the 20 VLN cases (70%) and 18 of the 20 tracking cases (90%), compared with 7 (35%) and 8 (40%) for Uni-NaVid, respectively. These correspond to success-rate improvements of 35 percentage points on VLN and 50 percentage points on tracking.

Qualitative real-world demonstrations across tasks. Figure 15 provides first-person visualizations of point-goal navigation, VLN, and tracking. These examples illustrate how RepresentNav approaches specified goals, follows route instructions through indoor environments, and follows moving targets.

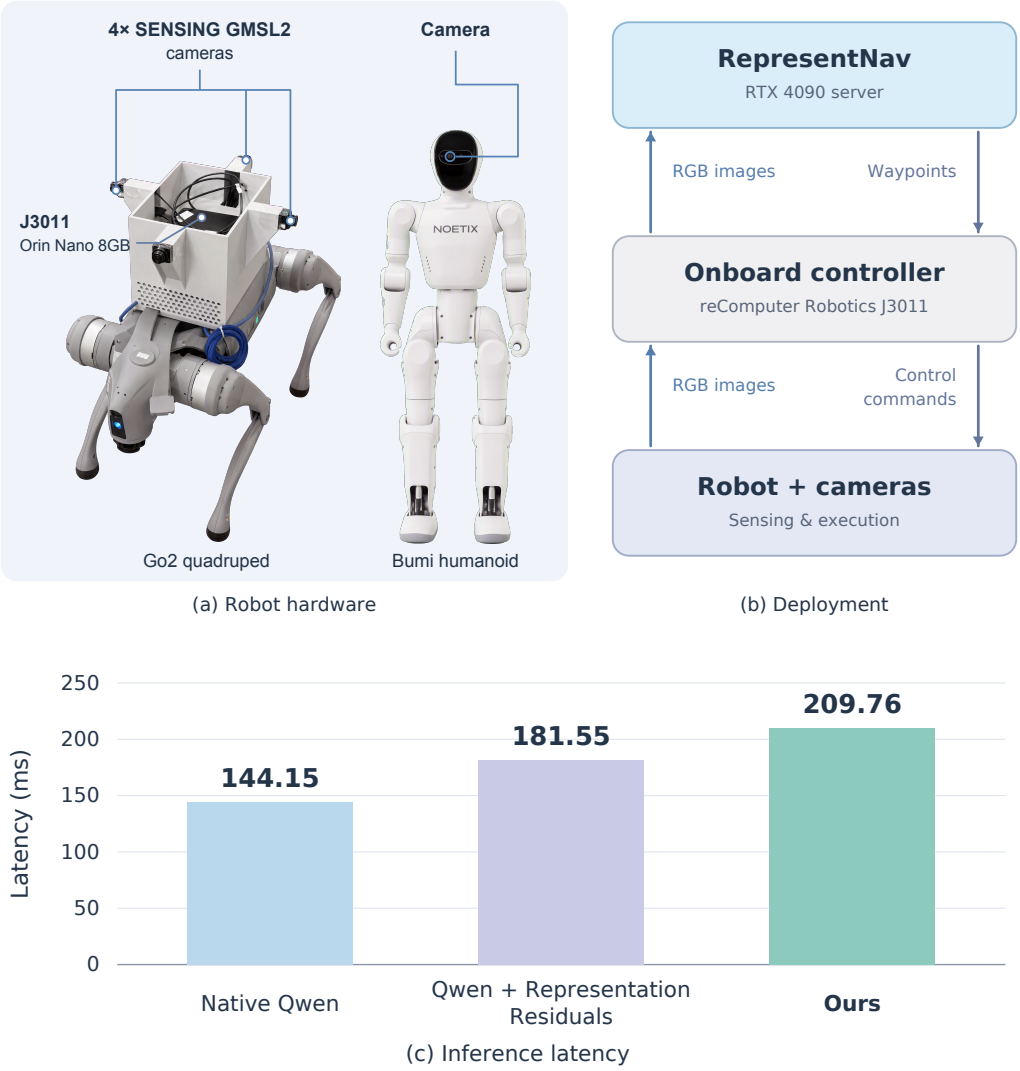


Figure 14: **Real-world robot platform and inference latency.** (a) The Go2 quadruped and Bumi humanoid, with camera locations marked. The Go2 carries four SENSING GMSL2 cameras and a reComputer Robotics J3011 with an NVIDIA Jetson Orin Nano 8GB module. (b) RGB observations are sent to an RTX 4090 server running RepresentNav; predicted waypoints are returned to the onboard controller for robot execution. (c) Inference latency of Native Qwen, Qwen with Representation Residuals, and our full model.

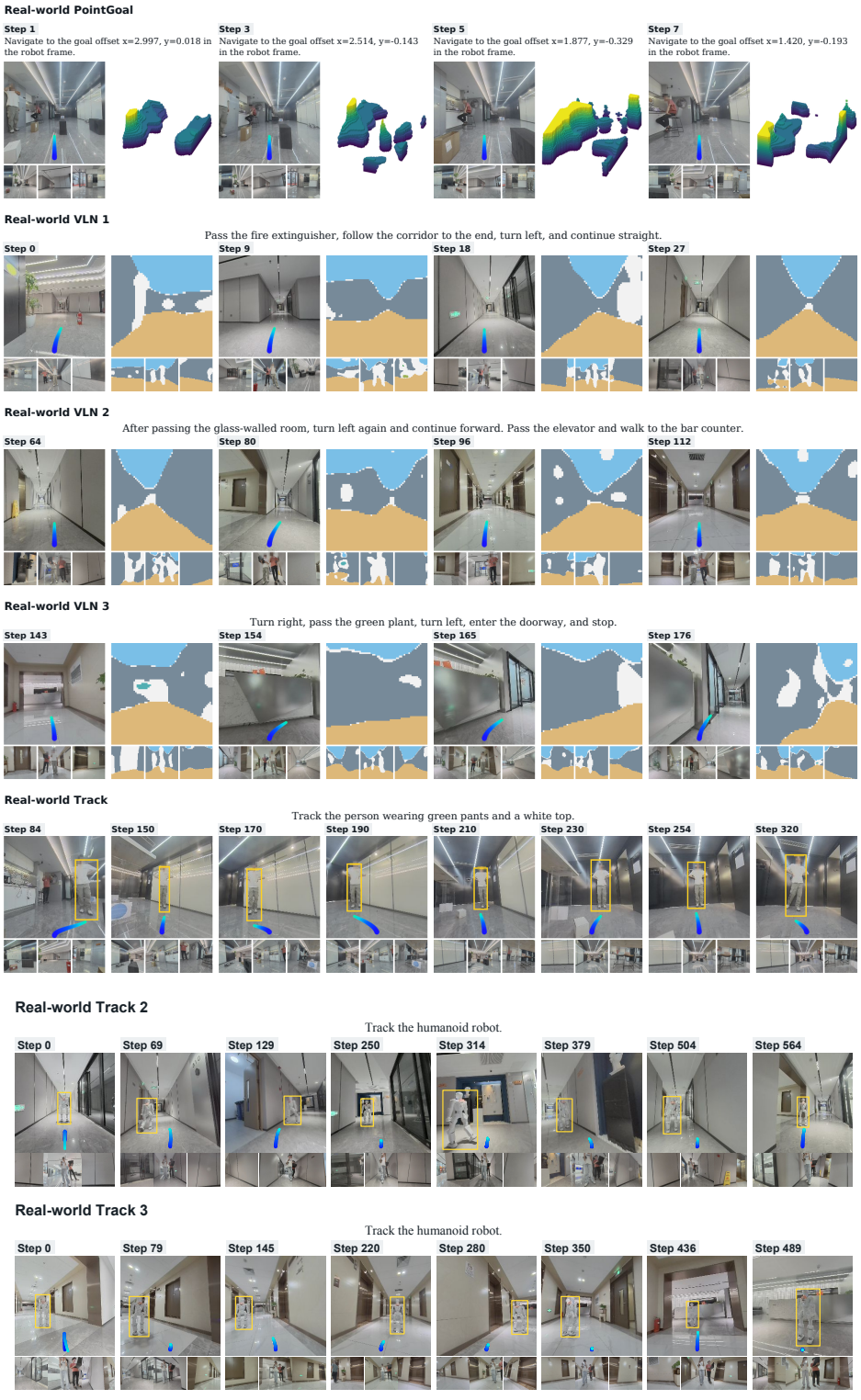


Figure 15: **Visualization of real-world experiments across tasks.**

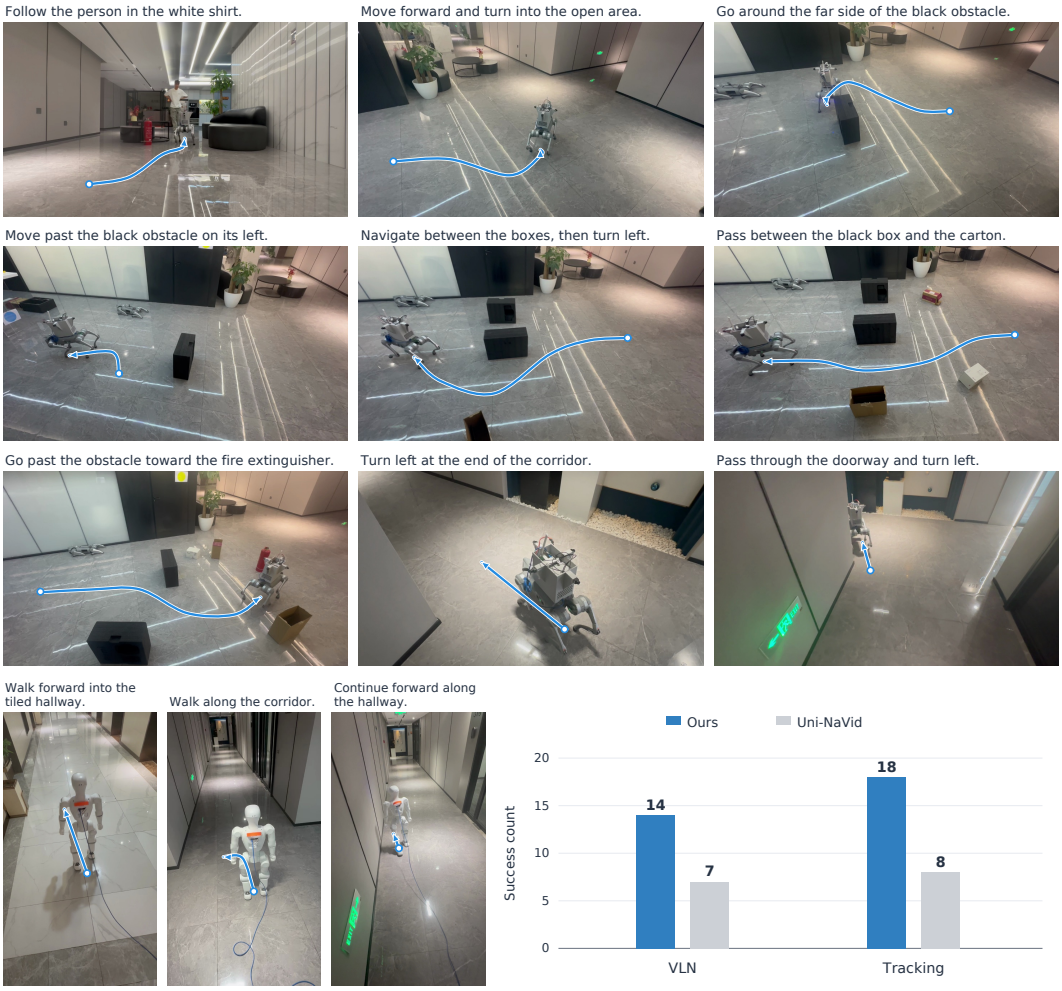


Figure 16: **Third-person real-world demonstrations.**

D AI USE STATEMENT

We use Claude to improve the writing of the manuscript. The authors take full responsibility for the final content, analyses, and conclusions.

E ETHICS STATEMENT

The public datasets and simulation assets used in this work are available under licenses permitting research use. Our real-world experiments do not disclose personal information: the faces of people who appear in our recordings are blurred.

F REPRODUCIBILITY STATEMENT

We describe Representation Residuals in Section 3.1, the latent chain-of-thought and its targets in Section 3.2, the two-stage curriculum in Section 3.3, and the training data in Section 3.4; Section 4 gives the benchmarks, metrics, and implementation details. Appendix A provides further architecture and training details, including the loss weights and data split of each stage, and Appendix B.1 gives the protocols of the ablations and representation analyses, to support reproducibility.