ALIGNED-VLN: MODELING NAVIGATION PROGRESS OVER ORDERED INSTRUCTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many recent Vision-and-Language Navigation (VLN) methods based on vision-language models directly predict actions from natural-language instructions and visual observations. While action supervision specifies what action to execute, it provides limited guidance about which instruction content should guide decisions at different navigation stages. In long-horizon navigation, a locally feasible action may be premature under the instruction order, making action selection dependent on execution progress. We refer to this challenge as *instruction-progress mismatch*. We propose **Aligned-VLN**, an action-chunk VLM policy that models navigation progress relative to the ordered instruction. Our key idea is to learn an *instruction-conditioned latent alignment* over the original instruction tokens. The alignment yields an instruction-relative progress coordinate and an attended instruction representation, which are fused with a learned navigation-state representation into a progress context for action decoding. This context conditions the decoder through layer-wise progress attention and final readout injection. The latent alignment is learned from weak trajectory-derived supervision without token-level execution annotations, with monotonic supervision encouraging progress estimates to follow trajectory order. At each decision step, Aligned-VLN predicts an action chunk with one VLM forward pass, without generated progress descriptions, predefined sub-instructions, or an external progress model. On R2R-CE Val-Unseen, Aligned-VLN achieves 63.5% SR and 56.3% SPL. On RxR-CE, it achieves 59.6% SR and 51.5% SPL. Separate R2R-only controlled experiments show improvements over an action-only policy using the same backbone and action interface.

1 INTRODUCTION

Recent advances in vision-language models (VLMs) have greatly advanced multimodal representation learning and embodied reasoning, laying a solid foundation for a wide range of vision-language grounded tasks. As a representative embodied AI task, Vision-and-Language Navigation (VLN) requires embodied agents to navigate real-world environments following natural language instructions based on egocentric visual observations (Anderson et al., 2018b; Krantz et al., 2020; Ku et al., 2020). Powered by the strong cross-modal modeling capability of VLMs, modern end-to-end VLN policies directly predict atomic or chunked actions through imitation learning and robust cross-modal alignment (Zhang et al., 2024; Zheng et al., 2025; Wei et al., 2026; Zhu et al., 2026). Despite improved overall performance, existing action supervision does not explicitly identify which part of the ordered instruction should guide a decision at a given navigation stage. This paradigm induces a critical **instruction-progress mismatch** problem: an action may be locally feasible under current visual observations but inconsistent with the execution stage specified by the instruction sequence. Without explicit alignment between navigation state and instruction order, such mismatch can lead to premature or delayed actions and accumulate spatial errors along the trajectory.

Existing efforts mitigate progress mismatch by introducing explicit representations of execution progress. Early methods use scalar completion signals, such as normalized goal distance, to summarize global route progress (Ma et al., 2019a;b). While compact and effective, such scalars do not indicate which part of the ordered instruction is currently most relevant. More recent approaches represent progress through semantic descriptions, structured sub-goals, or step-level execution states (Wang et al., 2026b; Wu et al., 2026; Chen et al., 2025b; Guo et al., 2026; Huang et al., 2026). These

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

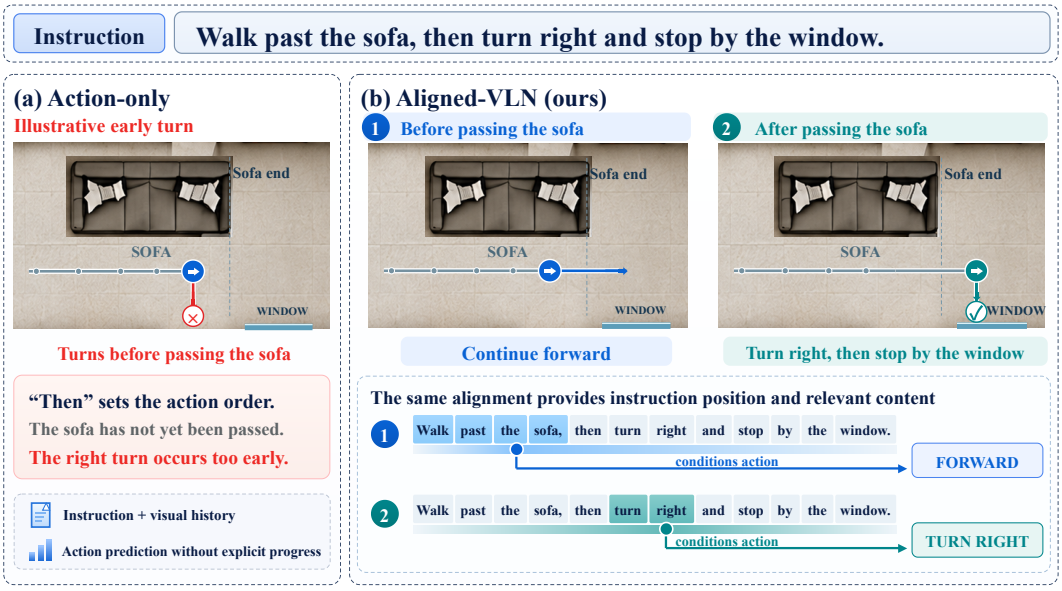

Figure 1: Motivation of **Aligned-VLN**. The same navigation behavior can be premature at an early stage but appropriate at a later stage. Aligned-VLN models navigation progress via instruction-conditioned latent alignment, where the same alignment identifies an instruction-relative position and the instruction content used to condition action prediction. All routes and alignment diagrams are schematic.

explicit progress interfaces provide richer execution-stage information, but introduce separate intermediate representations between the navigation state and the original instruction. This motivates a complementary question: *can navigation progress be represented directly as a latent alignment over the ordered instruction itself?*

To address this challenge, we propose **Aligned-VLN**, an instruction-aligned navigation framework that models sequential progress via **instruction-conditioned latent alignment**. Rather than generating a separate progress description or relying on an external progress model, Aligned-VLN computes a state-conditioned distribution over the original instruction tokens. The same weights yield an expected instruction-token position and an attended instruction representation, linking progress estimation to the language features used for action prediction. Together with the route state, these readouts form a progress context for action decoding. We train the alignment and an auxiliary local progress-increment predictor using coarse trajectory-derived targets, without requiring token-level execution annotations. At each decision step, the shared VLM backbone supplies features for alignment and short action-chunk prediction in a single forward pass. Experiments on R2R-CE and RxR-CE, together with controlled R2R-only ablations, demonstrate the effectiveness of the framework and examine the contribution of its progress-related components. Our contributions are summarized as follows:

- We introduce Aligned-VLN, a unified framework for VLN that models sequential navigation progress through instruction-conditioned latent alignment, deriving continuous progress coordinates and instruction-attended representations from a shared token-level distribution.
- We design a progress-enhanced action decoder equipped with multi-source attention, which tightly integrates latent alignment contexts into chunk-based action prediction for stage-aware navigation decision-making.
- We develop trajectory-derived weak supervision for latent alignment and local progress-increment prediction. Evaluations on R2R-CE and RxR-CE show competitive navigation performance, while controlled R2R-only ablations assess the contribution of the progress-related components.

2 Related Work

Vision-and-Language Navigation. Traditional vision-and-language navigation systems depend on recurrent cross-modal architectures, explicit spatial mappings, or standalone waypoint prediction modules to precisely align visual observations with linguistic navigation instructions (Hong et al., 2021; Georgakis et al., 2022; Krantz et al., 2021). Driven by powerful large-scale vision-language pre-training, modern VLN methods adopt streamlined end-to-end policy learning to directly predict atomic or chunked navigation actions from raw egocentric inputs. Recent representative approaches improve overall navigation performance from diverse technical perspectives: NaVid models complex temporal video dynamics, Efficient-VLN optimizes training stability and streaming inference efficiency, StreamVLN aggregates long-range visual contexts, and NaVIDA introduces hierarchical action chunks to refine decision granularity (Zhang et al., 2024; Zheng et al., 2025; Wei et al., 2026; Zhu et al., 2026). These approaches advance perception, temporal modeling, and action generation. Our work instead focuses on relating the current navigation state to positions in the ordered instruction. Meanwhile, VLN benchmarks have gradually evolved toward more realistic and complex embodied settings. R2R builds the basic instruction-following paradigm on discrete navigation graphs (Anderson et al., 2018b), VLN-CE extends tasks to continuous photorealistic environments for practical robotic navigation (Krantz et al., 2020), and RxR enhances task complexity via multilingual instructions and dense trajectory annotations (Ku et al., 2020). Complementing these perception- and action-oriented studies, we model instruction-relative alignment within the VLM policy at each decision step.

Progress Modeling in VLN. Progress-aware modeling is essential for resolving inherent instruction-stage mismatches and stabilizing long-horizon navigation decisions. Early methods adopt simple scalar metrics to estimate global route completion status. Self-Monitoring unifies vision-language grounding and scalar progress estimation, and the Regretful Agent leverages progress signals for effective backtracking correction (Ma et al., 2019a;b). However, such scalar supervision lacks fine-grained linguistic semantics and fails to capture diverse stage-specific instruction constraints. Recent works enrich progress modeling with explicit semantic reasoning and structured state decomposition. Progress-Think generates descriptive textual progress descriptions (Wang et al., 2026b), Dual-Anchoring tracks completed and residual sub-goals (Wu et al., 2026), CA-Nav and AwareVLN introduce constraint and self-aware reasoning mechanisms (Chen et al., 2025b; Guo et al., 2026), and Route2Step explicitly decomposes instructions into step-wise execution states (Huang et al., 2026). These approaches represent progress through different intermediate interfaces. Aligned-VLN instead derives a continuous progress coordinate and an attended instruction representation from one state-conditioned token distribution. This design couples the two readouts without generating progress text or manually segmenting the instruction, while retaining one VLM forward pass per decision step.

3 Method

3.1 Problem Formulation

At decision step t, a vision-and-language navigation agent receives an instruction $I = (w_1, \ldots, w_N)$, sampled visual history $V_{<t}$, and the current view V_t. It predicts a short action chunk

$$\mathbf{a}_t = f_\theta(I, V_{<t}, V_t), \qquad \mathbf{a}_t = (a_{t,1}, \ldots, a_{t,K}), \quad a_{t,k} \in \mathcal{A}. \tag{1}$$

Here K denotes the action-chunk length, and $k \in \{1, \ldots, K\}$ indexes an action slot. Following NaVIDA (Zhu et al., 2026), we adopt the same ten-symbol action vocabulary $\mathcal{A}$.

We infer a state-conditioned alignment distribution $\boldsymbol{\pi}_t$ over ordered instruction tokens. Its shared weights produce an instruction-relative coordinate and an attended instruction state, coupling progress estimation with the language features used for action prediction. Their fused context conditions the decoder, as shown in Figure 2.

3.2 Shared Multimodal Encoding

We use a trainable Qwen2.5-VL backbone (Bai et al., 2025) to encode the instruction, visual history, and current view in one forward pass. The input also contains M learnable PROGRESS-Q tokens

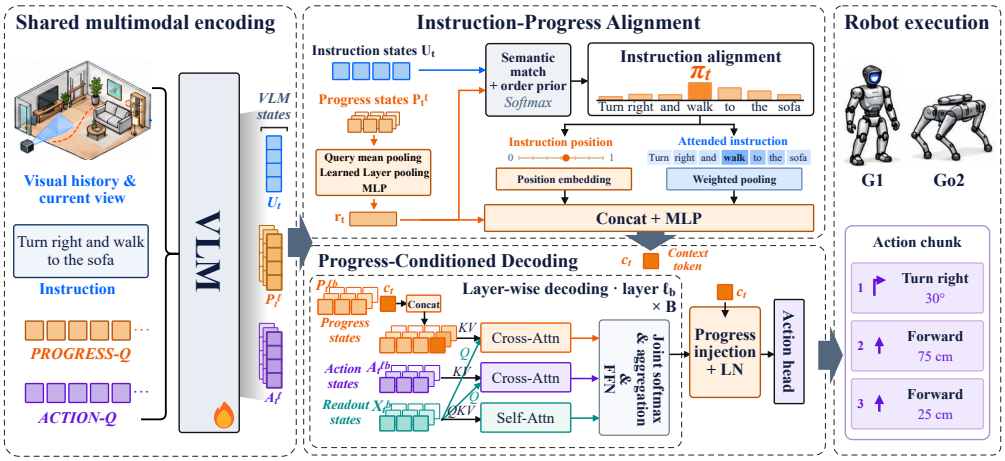

Figure 2: **Aligned-VLN overview.** A trainable shared multimodal backbone provides instruction, progress-query, and action-query states. Instruction-progress alignment uses one distribution to obtain a progress coordinate and attended instruction state. Their fused context conditions layer-wise action decoding.

followed by Q ACTION-Q tokens. From selected late backbone layers $\ell \in \mathcal{S}$, we retain progress-query states $\mathbf{P}_t^\ell \in \mathbb{R}^{M\times d}$ and action-query states $\mathbf{A}_t^\ell \in \mathbb{R}^{Q\times d}$. We denote the final-layer states of the N_t valid instruction tokens by $\mathbf{U}_t = (\mathbf{u}_{t,1}, \ldots, \mathbf{u}_{t,N_t})$. These three representations share the same visual and linguistic encoding, but play different roles below: $\mathbf{U}_t$ is aligned, $\mathbf{P}_t^\ell$ summarizes route state, and $\mathbf{A}_t^\ell$ supports action decoding.

3.3 INSTRUCTION-PROGRESS ALIGNMENT

To represent route progress in instruction space, we align the current navigation state with the original ordered instruction tokens. Route-dependent content matching and a positional score jointly define $\boldsymbol{\pi}_t$, which supplies the shared weights for both readouts below.

Route state. We first summarize the progress queries across tokens and selected layers:

$$\mathbf{r}_t = \mathrm{MLP}_r\left(\sum_{\ell\in\mathcal{S}} \alpha_\ell \frac{1}{M}\sum_{m=1}^{M} \mathbf{P}_{t,m}^\ell\right). \tag{2}$$

Here α_ℓ are learned softmax-normalized layer weights. The resulting route state $\mathbf{r}_t$ is the query that relates the current navigation state to individual instruction tokens.

Content- and order-aware alignment. For valid token i, let $x_{t,i} = (i-1)/\max(N_t-1, 1)$ be its normalized position in the instruction. The route state predicts the center μ_t and positive width ρ_t of a positional bias. We combine this bias with content matching:

$$\pi_{t,i} = \mathrm{softmax}_i\left(\frac{(W_q\mathbf{r}_t)^\top W_k\,\mathrm{LN}(\mathbf{u}_{t,i})}{\sqrt{d}} - \gamma\frac{(x_{t,i}-\mu_t)^2}{2\rho_t^2}\right). \tag{3}$$

Padded tokens are masked before softmax. The first term matches route state to token content. The second makes instruction order available to the same distribution.

Two readouts from one alignment. The alignment becomes useful to the decoder through two complementary expectations:

$$\hat{p}_t = \sum_{i=1}^{N_t} \pi_{t,i}x_{t,i}, \qquad \bar{\mathbf{u}}_t = \sum_{i=1}^{N_t} \pi_{t,i}\mathbf{u}_{t,i}. \tag{4}$$

The scalar $\hat{p}_t$ is an expected *instruction-token position*, not measured travel distance. The vector $\bar{\mathbf{u}}_t$ is the attended instruction state computed with exactly the same weights. Thus the state-conditioned alignment that sets the progress coordinate also sets the instruction features passed to action prediction. We concatenate these readouts with the route state:

$$\mathbf{c}_t = \mathrm{MLP}_c([\mathbf{r}_t, \bar{\mathbf{u}}_t, \phi(\hat{p}_t)]), \tag{5}$$

where ϕ embeds the scalar coordinate. Thus $\mathbf{c}_t$ carries route state, instruction-relative position, and the attended instruction state into the action decoder.

3.4 Progress-Conditioned Action Decoding

The decoder starts from R learned readouts $\mathbf{X}_t^0$ and updates them through B blocks. Block b uses action-query states $\mathbf{A}_t^{\ell_b}$ and the progress source $\mathbf{G}_t^b = [\mathbf{P}_t^{\ell_b}, \mathbf{c}_t]$, which joins layer-specific progress states with the aligned context. The context also follows a final residual path to the action head.

Each block attends to current readouts (X), action queries (A), and the progress source (G). Queries $\mathbf{Q}_t^b$ come from $\mathbf{X}_t^b$. Keys and values $\mathbf{K}_{j,t}^b, \mathbf{V}_{j,t}^b$ come from source $j \in \{X, A, G\}$. Suppressing the head index, all three branches share:

$$\begin{aligned} \mathbf{S}_{j,t}^b &= \frac{\mathbf{Q}_t^b (\mathbf{K}_{j,t}^b)^\top}{\sqrt{d_h}}, \qquad j \in \{X, A, G\}, \\ \mathbf{\Omega}_t^b &= \mathrm{softmax}_{\mathrm{tokens}}\big([\mathbf{S}_{X,t}^b, \mathbf{S}_{A,t}^b, g_b \mathbf{S}_{G,t}^b]\big), \qquad g_b = \tanh(\eta_b), \\ \mathbf{Y}_t^b &= \mathbf{\Omega}_t^b [\mathbf{V}_{X,t}^b, \mathbf{V}_{A,t}^b, \mathbf{V}_{G,t}^b]. \end{aligned} \tag{6}$$

The scores are concatenated along source tokens *before one softmax*, with values in the same order. The learned g_b scales progress-source logits. Output projection, residual update, and a feed-forward network produce $\mathbf{X}_t^{b+1}$.

In each block, $\mathbf{c}_t$ participates in joint attention with action and readout sources. A separately scaled residual also carries it directly to the action head:

$$\widehat{\mathbf{X}}_t = \mathrm{LN}\big(\mathbf{X}_t^B + \tanh(\eta_{\mathrm{out}}) W_g \mathbf{c}_t\big), \qquad \mathbf{Z}_t = \mathrm{ActionHead}\Big(\mathrm{vec}(\widehat{\mathbf{X}}_t)\Big). \tag{7}$$

Here the projected context is broadcast to all final readouts, without another attention normalization. The linear action head produces $\mathbf{Z}_t \in \mathbb{R}^{K \times |\mathcal{A}|}$, one logit row per slot. Section 4.1 specifies the closed-loop execution schedule. During training, an auxiliary head predicts the progress increment from the mean final readout and $\mathbf{c}_t$. It does not select actions.

3.5 Trajectory-Derived Progress Learning

Expert demonstrations provide ordered states and action chunks, but no state-to-instruction-token annotations. We use the fraction of completed forward primitive actions as a coarse target $p_t^\star$ for the instruction-relative position at step t. The target $\Delta p_t^\star$ is the change in this fraction over the supervised expert chunk. In-place turns do not contribute to either target.

Action imitation is the primary training signal. $\mathcal{L}_{\mathrm{act}}$ averages cross-entropy between the K action-logit rows and their expert symbols. Because the action logits depend on the progress context $\mathbf{c}_t$, gradients from this loss can also reach the alignment $\boldsymbol{\pi}_t$ through the decoder. The trajectory-derived target supplies an additional, coarse reference for its position: $\mathcal{L}_{\mathrm{prog}}$ applies smooth-L_1 regression between the expected coordinate $\hat{p}_t$ and $p_t^\star$. To guide the token distribution beyond this scalar readout, we form a soft target over valid instruction positions:

$$q_{t,i} \propto \exp\left(-\frac{(x_{t,i} - p_t^\star)^2}{2\sigma_{\mathrm{align}}^2}\right), \qquad \mathcal{L}_{\mathrm{align}} = -\sum_{i=1}^{N_t} q_{t,i} \log \pi_{t,i}. \tag{8}$$

We normalize $q_{t,i}$ over valid tokens. Thus the two losses derived from $p_t^\star$ constrain different aspects of the same alignment: $\mathcal{L}_{\mathrm{prog}}$ targets its expected coordinate, while $\mathcal{L}_{\mathrm{align}}$ guides the distribution whose weights also determine the attended instruction state.

Table 1: Comparison with recent VLN methods on R2R-CE and RxR-CE validation-unseen. Numbers are from the cited papers and may differ in modalities, training data, or evaluation implementation. Model size denotes the reported VLM backbone. **Bold** and underlined denote the best and second-best reported results.

Method	Model Size	R2R-CE Val-Unseen				RxR-CE Val-Unseen			
		NE ↓	OSR ↑	SR ↑	SPL ↑	NE ↓	SR ↑	SPL ↑	nDTW ↑
InstructNav (Long et al., 2025) [CoRL24]	–	6.89	–	31.0	24.0	–	–	–	–
AO-Planner (Chen et al., 2025a) [AAAI25]	–	5.55	59.0	47.0	33.0	7.06	43.3	30.5	50.1
LaViRA (Ding et al., 2026) [ICRA2026]	–	6.54	48.7	38.3	28.3	–	–	–	–
CA-Nav (Chen et al., 2025b) [TPAMI2025]	–	7.58	48.0	25.3	10.8	–	–	–	–
CMA (Krantz et al., 2020; Hong et al., 2022) [CVPR22]	–	6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0
VLN○BERT (Hong et al., 2021; 2022) [CVPR22]	–	5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7
Ego^2-Map (Hong et al., 2023) [ICCV23]	–	5.54	56.0	47.0	41.0	–	–	–	–
g3D-LF (Wang & Lee, 2025) [CVPR25]	–	5.70	59.5	47.2	34.6	–	–	–	–
Sim2Real (Wang et al., 2025b) [CoRL24]	–	5.95	55.8	44.9	30.4	8.79	25.5	18.1	–
NavMorph (Yao et al., 2025) [ICCV25]	–	5.75	56.9	47.9	33.2	8.85	30.8	22.8	44.2
MapNav (Zhang et al., 2025b) [ACL25]	7B	4.93	53.0	39.7	37.2	7.62	32.6	27.7	43.5
NaVid (Zhang et al., 2024) [RSS24]	7B	5.47	49.1	37.4	35.9	8.41	23.8	21.2	–
NaVid-4D (Liu et al., 2025) [ICRA25]	7B	5.99	55.7	43.8	37.1	–	–	–	–
NaVILA (Cheng et al., 2025) [RSS25]	8B	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8
Uni-NaVid (Zhang et al., 2025a) [RSS25]	7B	5.58	53.3	47.0	42.7	6.24	48.7	40.9	–
Aux-Think (Wang et al., 2025a) [NeurIPS25]	8B	6.08	60.0	54.8	46.9	6.24	52.2	40.2	–
StreamVLN (Wei et al., 2026) [ICRA26]	7B	4.98	64.2	56.9	51.9	6.22	52.9	46.0	61.9
Efficient-VLN (Zheng et al., 2025) [arXiv25 v2]	8B	**3.29**	**78.2**	**73.2**	**65.0**	**3.18**	**75.6**	60.0	–
MonoDream (Wang et al., 2026a) [AAAI26]	2B	5.45	61.5	55.8	49.1	6.38	49.4	40.9	–
CorrectNav (Yu et al., 2026) [AAAI26]	7B	4.24	67.5	65.1	62.3	4.09	69.3	**63.3**	**75.2**
DualVLN (Wei et al., 2025) [arXiv25]	7B	4.05	70.7	64.3	58.5	4.58	61.4	51.8	70.0
NavFoM (Zhang et al., 2026) [ICLR26]	7B	5.01	64.9	56.2	51.2	5.51	57.4	49.4	60.2
JanusVLN (Zeng et al., 2026) [ICLR26]	8B	4.78	65.2	60.5	56.8	6.06	56.2	47.5	62.1
NaVIDA (Zhu et al., 2026) [arXiv26]	3B	4.32	69.5	61.4	54.7	5.23	57.4	49.6	67.0
Aligned-VLN (Ours)	3B	4.12	72.2	63.5	56.3	5.03	59.6	51.5	67.7

The remaining objectives supervise temporal change and order. $\mathcal{L}_{\Delta}$ applies smooth-L_1 loss to the auxiliary increment prediction and $\Delta p_t^{\star}$. For an earlier–later state pair (e, l) from the same demonstration, $\mathcal{L}_{\text{mono}}$ penalizes $\max(0, \hat{p}_e - \hat{p}_l - \epsilon)$, allowing decreases within the tolerance ϵ. We combine all five terms as

$$\mathcal{L} = \lambda_{\text{act}}\mathcal{L}_{\text{act}} + \lambda_{\text{prog}}\mathcal{L}_{\text{prog}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\Delta}\mathcal{L}_{\Delta} + \lambda_{\text{mono}}\mathcal{L}_{\text{mono}}. \tag{9}$$

4 Experiments

4.1 Experimental Setup

Datasets, Metrics and Implementation. We conduct extensive evaluations on two challenging and widely adopted benchmarks, namely R2R-CE and RxR-CE. Following existing state-of-the-art VLN practices, our main comparison experiments are trained on a large-scale aggregated trajectory corpus, which integrates rich navigation data from R2R, RxR, and ScaleVLN datasets (Anderson et al., 2018b; Krantz et al., 2020; Ku et al., 2020; Wang et al., 2023). To ensure rigorous and fair ablation analysis, all controlled ablation experiments are trained purely on original R2R trajectories with strictly unified data partitioning, optimization budgets, and evaluation settings, and we consistently evaluate ablation models on the standard R2R-CE validation-unseen split, which contains 1,839 navigation instructions, 613 distinct physical routes, and 11 completely unseen novel scenes for generalized performance verification.

We adopt five mainstream VLN evaluation metrics, including NE, OSR, SR, SPL, and nDTW, in strict accordance with standard evaluation protocols (Anderson et al., 2018a; Ilharco et al., 2019). Specifically, NE denotes the final navigation error distance toward the target goal, OSR checks whether the agent successfully enters the predefined goal success region, SR represents the overall task success rate, SPL calculates the success-weighted path length efficiency across all test episodes,

Table 2: Controlled ablations on R2R-CE validation-unseen. All variants use R2R-only training and identical optimization/evaluation settings. All but the Qwen3-VL-2B row use Qwen2.5-VL-3B. That row replaces the backbone with Qwen3-VL-2B while keeping the remaining controlled setting unchanged. Content-based matching denotes the semantic scoring term in Eq. 3, not the language-value readout.

Variant	Progress queries	Scalar	Positional bias	Content-based matching	Mono	NE ↓	OSR ↑	SR ↑	SPL ↑
ActionOnly	–	–	–	–	–	7.99	41.22	31.81	28.97
PositionOnly	✓	✓	✓	–	✓	7.33	44.64	36.65	33.44
NoMono	✓	✓	✓	✓	–	7.32	44.26	36.65	33.46
Qwen3-VL-2B Full	✓	✓	✓	✓	✓	7.47	44.90	35.00	30.70
ScalarOnly	✓	✓	–	–	✓	7.12	45.46	38.77	34.40
Full	✓	✓	✓	✓	✓	**6.91**	**47.20**	**39.37**	**36.07**

and nDTW measures the trajectory similarity and behavioral fidelity between agent predictions and expert reference routes.

For our default model configuration, we use Qwen2.5-VL-3B as the vision-language backbone and adopt the same ten-symbol action vocabulary as NaVIDA (Zhu et al., 2026). At each decision step, the policy predicts a three-symbol action chunk. We execute the first predicted symbol by expanding it into simulator primitives, then obtain a new observation before replanning. Our full Aligned-VLN model uses 32 progress query tokens, 32 action query tokens, and 8 readout query tokens. We extract hidden states from eight backbone layers and use up to eight historical visual observations.

4.2 Comparison with Recent VLN Methods

Table 1 compares Aligned-VLN with modular and VLM-based methods on the R2R-CE and RxR-CE validation-unseen splits. Because published systems differ in modalities, training data, and evaluation details, this table provides contextual rather than controlled comparisons.

Using a Qwen2.5-VL-3B backbone and the aggregated training corpus described above, Aligned-VLN reaches 63.5% SR and 56.3% SPL on R2R-CE, and 59.6% SR, 51.5% SPL, and 67.7 nDTW on RxR-CE. Its 3B backbone is smaller than the 7B–8B backbones used by several VLM-based methods in Table 1, while achieving competitive results subject to the protocol differences noted above. At each decision step, instruction-progress alignment and action-chunk prediction share a single VLM forward pass, without a separately invoked progress model or generated progress description.

4.3 Controlled Ablations

We conduct R2R-only controlled ablations to examine the roles of progress conditioning, content-based instruction matching, and temporal-order supervision. The variants follow the training and evaluation setting in Section 4.1; the Qwen3-VL-2B Full variant uses a different backbone and is not used for component attribution. We compare four baselines with Full. ActionOnly removes progress queries, progress supervision, and progress-dependent conditioning. PositionOnly removes content-based matching from the alignment logits while retaining the positional bias and instruction-attended representation. NoMono disables temporal monotonic regularization. ScalarOnly predicts a scalar progress value from the route state and uses only its embedding as the decoder's progress source, omitting token-level alignment. Full includes all proposed components.

Effect of progress modeling. In the R2R-only ablation, Full exceeds ActionOnly by 7.56, 7.10, and 5.98 percentage points in SR, SPL, and OSR, respectively, and reduces NE by 1.08 meters. We quantify uncertainty using paired route-level bootstrap resampling with 10,000 resamples. The corresponding 95% confidence intervals for the SR, SPL, and OSR differences are [4.79, 10.33], [4.54, 9.66], and [3.15, 8.75], respectively. All three intervals exclude zero under route-level resampling, supporting an improvement for these evaluated checkpoints.

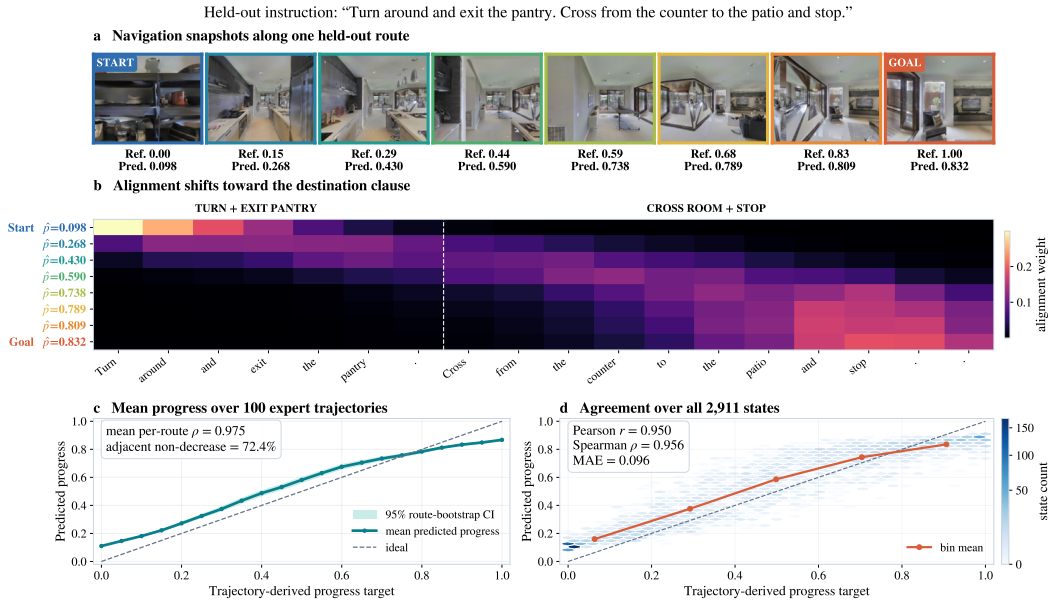

Figure 3: **Visualization of learned progress and token alignment.** (a) Eight states selected near uniformly spaced reference-progress targets from a held-out expert route. (b) Token alignment shifts from the opening clause toward the destination clause. (c) Mean progress over 100 held-out expert trajectories with 95% route-bootstrap CI. (d) Agreement across all 2,911 states.

Role of instruction content and temporal order. SCALARONLY improves SR from 31.81 to 38.77 over ACTIONONLY, indicating a benefit from scalar progress conditioning in this R2R-only setting. FULL further increases SR by 0.60 and SPL by 1.67 percentage points over SCALARONLY, suggesting an additional contribution from token-level alignment under the same setting. Relative to POSITIONONLY, FULL improves SR by 2.72 and SPL by 2.63 points, consistent with a benefit from content-based matching. Removing monotonic regularization lowers SR, SPL, and OSR by 2.72, 2.61, and 2.94 points, respectively, consistent with a contribution from temporal-order supervision. Table 2 also reports a Qwen3-VL-2B FULL variant. Because its backbone differs, we do not use that row to attribute gains to an individual component.

4.4 ANALYSIS OF LEARNED ALIGNMENT AND PROGRESS

We replay a fixed collection of held-out episodes from the R2R-CE Val-Unseen split following expert trajectories, and compare the model's predicted progress against the normalized fraction of completed forward-motion primitives. In-place rotational actions are excluded and do not accumulate toward the reference progress metric. Over all 2,911 evaluated decision states, we obtain Pearson and Spearman correlation coefficients of 0.950 and 0.956, respectively. The strong correlation demonstrates that our learned progress coordinate faithfully tracks the sequential execution order along unseen expert trajectories. It is worth noting that this replay analysis only quantifies consistency with trajectory-based progress targets, and cannot independently verify semantic alignment with the natural language instruction completion status.

Figure 3(b) further supplies qualitative evidence for our alignment mechanism. For a representative unseen expert route, the model's token-level alignment gradually transfers from the introductory instruction segment toward the destination clause as navigation proceeds. The mean absolute error (MAE) of predicted progress relative to trajectory-derived ground-truth targets reaches 0.096. Nevertheless, temporal monotonicity is not perfectly preserved: 72.4% of consecutive progress predictions satisfy the non-decreasing property. This statistic counts all detected progress drops, rather than only those exceeding the tolerance threshold used during model training. We emphasize that these expert-trajectory replay experiments are limited and cannot evaluate the model's error-recovery capability once the agent deviates from the reference route.

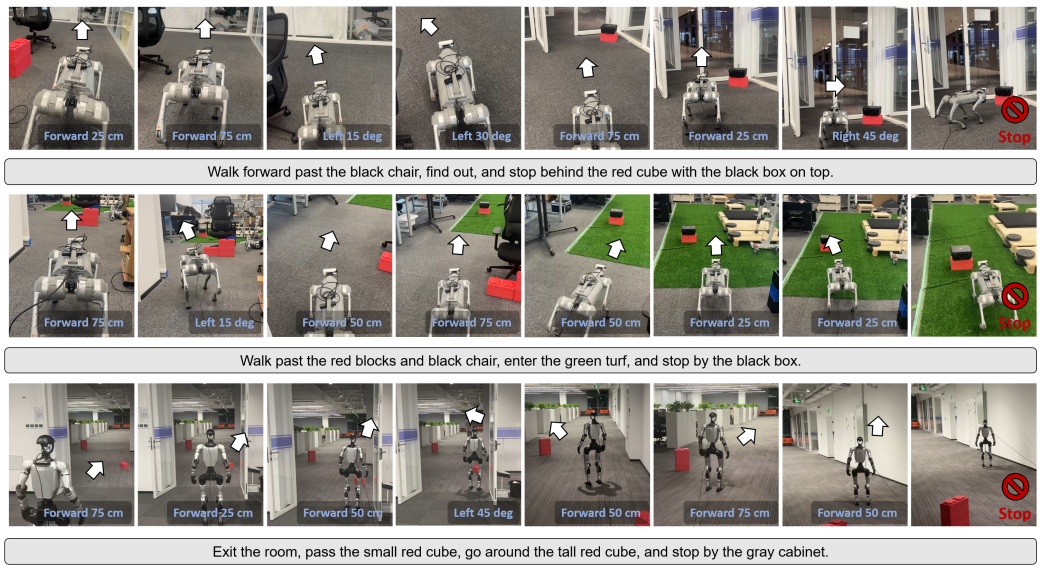

Figure 4: **Real-world deployment of Aligned-VLN.** Representative closed-loop indoor trajectories under multi-stage instructions. Arrows show executed motion and overlays show predicted action symbols. The deployment uses the same ten-symbol action vocabulary as in simulation, without an external progress model or inference-time search.

4.5 Real-World Deployment

We qualitatively deploy Aligned-VLN with the ten-symbol action vocabulary adopted from NaVIDA (Zhu et al., 2026) on two physical robot platforms, the Unitree G1 humanoid and Unitree Go2 quadruped, across three indoor navigation tasks. The deployment uses neither an external progress model nor inference-time search. Figure 4 shows representative closed-loop trajectories under multi-stage instructions. In these examples, the policy follows ordered landmarks and issues the STOP action near the designated destination. We present these cross-platform demonstrations as preliminary qualitative evidence of feasibility, not as a quantitative benchmark comparison. Larger-scale real-world evaluations across more diverse scenes and challenging navigation scenarios remain future work.

5 Discussion

Aligned-VLN couples progress estimation and instruction grounding through a shared token-level distribution. In the controlled R2R-CE setting, FULL improves over ACTIONONLY and shows a smaller additional gain over SCALARONLY, consistent with contributions from progress conditioning and token-level alignment. Trajectory-derived progress targets may diverge from semantic instruction completion. Expert replays validate trajectory alignment but cannot guarantee instruction-order correctness in autonomous navigation. Bootstrap statistics and real-robot tests offer preliminary evidence, while thorough semantic grounding evaluation, multi-seed validation, and large-scale real-world deployment are reserved for future work.

AI Use Statement

We used generative AI tools to improve the clarity and readability of the manuscript. The authors take responsibility for the final content.

REFERENCES

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents, 2018a. URL https://arxiv.org/abs/1807.06757.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3674–3683, June 2018b. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Anderson_Vision-and-Language_Navigation_Interpreting_CVPR_2018_paper.html.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Jiaqi Chen, Bingqian Lin, Xinmin Liu, Lin Ma, Xiaodan Liang, and Kwan-Yee K. Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23568–23576, 2025a. doi: 10.1609/aaai.v39i22.34526. URL https://ojs.aaai.org/index.php/AAAI/article/view/34526.

Kehan Chen, Dong An, Yan Huang, Rongtao Xu, Yifei Su, Yonggen Ling, Ian Reid, and Liang Wang. Constraint-aware zero-shot vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11):10441–10456, 2025b. doi: 10.1109/TPAMI.2025.3594204. URL https://doi.org/10.1109/TPAMI.2025.3594204.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. In *Proceedings of Robotics: Science and Systems*, Los Angeles, California, June 2025. doi: 10.15607/RSS.2025.XXI.018. URL https://www.roboticsproceedings.org/rss21/p018.html.

Hongyu Ding, Ziming Xu, Yudong Fang, You Wu, Zixuan Chen, Jieqi Shi, Jing Huo, Yifan Zhang, and Yang Gao. LaViRA: Language-vision-robot actions translation for zero-shot vision language navigation in continuous environments. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://arxiv.org/abs/2510.19655.

Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15460–15470, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Georgakis_Cross-Modal_Map_Learning_for_Vision_and_Language_Navigation_CVPR_2022_paper.html.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4065–4075, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Guo_AwareVLN_Reasoning_with_Self-awareness_for_Vision-Language_Navigation_CVPR_2026_paper.html.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. VLN BERT: A recurrent vision-and-language BERT for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1643–1653, June 2021. URL

https://openaccess.thecvf.com/content/CVPR2021/html/Hong_VLN_BERT_A_Recurrent_Vision-and-Language_BERT_for_Navigation_CVPR_2021_paper.html.

Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15439–15449, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Hong_Bridging_the_Gap_Between_Learning_in_Discrete_and_Continuous_Environments_CVPR_2022_paper.html.

Yicong Hong, Yang Zhou, Ruiyi Zhang, Franck Dernoncourt, Trung Bui, Stephen Gould, and Hao Tan. Learning navigational visual representations with semantic map supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3055–3067, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Hong_Learning_Navigational_Visual_Representations_with_Semantic_Map_Supervision_ICCV_2023_paper.html.

Xiangyun Huang, Xiangchen Wang, Runfeng Lin, Yihao Xu, Kangyu Huang, Jiang Hengchen, Xiwang Dong, and Lin Jiarong. From routes to steps: Separating semantic progress from local execution in vision-and-language navigation, 2026. URL https://arxiv.org/abs/2608.03143.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop*, 2019. URL https://arxiv.org/abs/1907.05446.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision – ECCV 2020*, pp. 104–120. Springer, 2020. doi: 10.1007/978-3-030-58604-1_7. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/html/6018_ECCV_2020_paper.php.

Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15162–15171, October 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Krantz_Waypoint_Models_for_Instruction-Guided_Navigation_in_Continuous_Environments_ICCV_2021_paper.html.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4392–4412, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.356. URL https://aclanthology.org/2020.emnlp-main.356/.

Haoran Liu, Weikang Wan, Xiqian Yu, Minghan Li, Jiazhao Zhang, Bo Zhao, Zhibo Chen, Zhongyuan Wang, Zhizheng Zhang, and He Wang. NaVid-4D: Unleashing spatial intelligence in egocentric RGB-D videos for vision-and-language navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10607–10615, 2025. doi: 10.1109/ICRA55743.2025.11128467. URL https://doi.org/10.1109/ICRA55743.2025.11128467.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. InstructNav: Zero-shot system for generic instruction navigation in unexplored environment. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2049–2060. PMLR, 2025. URL https://proceedings.mlr.press/v270/long25b.html.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *International*

Conference on Learning Representations (ICLR), 2019a. URL https://openreview.net/forum?id=r1GAsjC5Fm.

Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6732–6740, June 2019b. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Ma_The_Regretful_Agent_Heuristic-Aided_Navigation_Through_Progress_Estimation_CVPR_2019_paper.html.

Shuo Wang, Yongcai Wang, Wanting Li, Xudong Cai, Yucheng Wang, Maiyue Chen, Kaihui Wang, Zhizhong Su, Deying Li, and Zhaoxin Fan. Aux-Think: Exploring reasoning strategies for data-efficient vision-language navigation. In *Advances in Neural Information Processing Systems 38*, pp. 34960–34984, 2025a. doi: 10.52202/085713-1040. URL https://doi.org/10.52202/085713-1040.

Shuo Wang, Yongcai Wang, Zhaoxin Fan, Yucheng Wang, Maiyue Chen, Kaihui Wang, Zhizhong Su, Wanting Li, Xudong Cai, Yeying Jin, and Deying Li. MonoDream: Monocular vision-language navigation with panoramic dreaming. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(12):10074–10082, 2026a. doi: 10.1609/aaai.v40i12.37974. URL https://doi.org/10.1609/aaai.v40i12.37974.

Shuo Wang, Yucheng Wang, Guoxin Lian, Yongcai Wang, Maiyue Chen, Kaihui Wang, Bo Zhang, Zhizhong Su, Yutian Zhou, Wanting Li, Deying Li, and Zhaoxin Fan. Progress-Think: Semantic progress reasoning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4076–4086, June 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wang_Progress-Think_Semantic_Progress_Reasoning_for_Vision-Language_Navigation_CVPR_2026_paper.html.

Zihan Wang and Gim Hee Lee. g3D-LF: Generalizable 3d-language feature fields for embodied tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14191–14202, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_g3D-LF_Generalizable_3D-Language_Feature_Fields_for_Embodied_Tasks_CVPR_2025_paper.html.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Sim-to-real transfer via 3d feature fields for vision-and-language navigation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2982–2995. PMLR, 2025b. URL https://proceedings.mlr.press/v270/wang25i.html.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12009–12020, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Wang_Scaling_Data_Generation_in_Vision-and-Language_Navigation_ICCV_2023_paper.html.

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, 2025. URL https://arxiv.org/abs/2512.08186.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. StreamVLN: Streaming vision-and-language navigation via SlowFast context modeling. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://arxiv.org/abs/2507.05240.

Kangyi Wu, Pengna Li, Kailin Lyu, Xi Lin, Lin Zhao, Qingrong He, Jinjun Wang, and Jianyi Liu. Dual-Anchoring: Addressing state drift in vision-language navigation. In *European Conference on Computer Vision (ECCV)*, 2026. URL https://arxiv.org/abs/2604.17473.

Xuan Yao, Junyu Gao, and Changsheng Xu. NavMorph: A self-evolving world model for vision-and-language navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5536–5546, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Yao_NavMorph_A_Self-Evolving_World_Model_for_Vision-and-Language_Navigation_in_Continuous_ICCV_2025_paper.html.

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(22), pp. 18737–18745, 2026. doi: 10.1609/aaai.v40i22.38942. URL https://ojs.aaai.org/index.php/AAAI/article/view/38942.

Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, and Xing Wei. JanusVLN: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=RnuB0Nlbd5.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.079. URL https://roboticsproceedings.org/rss20/p079.html.

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*, Los Angeles, California, June 2025a. 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 (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ceb45c48e29e7f49b0b47edb98e43691-Abstract-Conference.html.

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A novel memory representation via annotated semantic maps for VLM-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13032–13056. Association for Computational Linguistics, July 2025b. doi: 10.18653/v1/2025.acl-long.638. URL https://aclanthology.org/2025.acl-long.638/.

Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Efficient-VLN: A simple yet strong baseline for efficient vision-language navigation, 2025. URL https://arxiv.org/abs/2512.10310v2. Version 2, updated June 2026.

Weiye Zhu, Zekai Zhang, Xiangchen Wang, Hewei Pan, Teng Wang, Tiantian Geng, Rongtao Xu, and Feng Zheng. NaVIDA: Vision-language navigation with inverse dynamics augmentation, 2026. URL https://arxiv.org/abs/2601.18188.

APPENDIX OVERVIEW

These appendices complement the main text with implementation details, additional controlled outcome breakdowns, qualitative simulation examples, and real-world setup material. All quantitative results below refer to the controlled evaluation reported in the main paper.

A IMPLEMENTATION AND REPRODUCIBILITY DETAILS

A.1 MODEL AND EVALUATION CONFIGURATION

Table 3 summarizes the configuration used in the main experiments and controlled evaluation.

Table 3: Model and evaluation configuration.

Component	Setting
Main backbone	Qwen2.5-VL-3B
Backbone training	End-to-end trainable
Controlled backbone swap	Qwen3-VL-2B
Action space	10-symbol vocabulary
Action chunk length	3
Progress queries	32
Action queries	32
Readout queries	8
Backbone hidden states	Last 8 layers
Decoder blocks B	8
Decoder attention heads	8
Historical observations	At most 8
R2R-CE Val-Unseen	1,839 instructions / 613 routes / 11 scenes
Bootstrap samples	10,000

Simulation scenes. We evaluate on the R2R-CE and RxR-CE Val-Unseen splits in Habitat-based continuous indoor environments (Krantz et al., 2020; Ku et al., 2020). Each episode places the agent in a photorealistic reconstruction of a building, where routes can pass through corridors, doorways, and furnished rooms. The changing first-person views contain the objects and spatial landmarks described by the route instruction. Rather than moving between discrete viewpoints on a navigation graph, the agent moves through the indoor scene while observing it through a forward-facing RGB camera. The policy uses the current view, sampled visual history, and instruction; depth, GPS/compass, goal coordinates, and a top-down map are not policy inputs. Figure 5 shows two example scene sequences; their maps visualize the completed trajectories after evaluation.

Layer-to-block mapping and readout. The eight selected backbone layers supply progress-query and action-query states to eight decoder blocks in the same order; instruction-token states are taken from the final backbone layer. At each block, the current readouts attend jointly to themselves, that layer's action-query states, and a progress source formed by concatenating that layer's progress-query states with the shared alignment context $\mathbf{c}_t$. The decoder uses eight attention heads per block. After the final progress-context residual and layer normalization, the eight readouts are flattened and passed through a linear action head to produce three rows of ten action logits. The auxiliary progress-increment head instead receives the mean of the final readouts together with $\mathbf{c}_t$.

Visual-history selection. The current observation is provided separately from the visual history. When more than eight earlier observations are available, we sample eight approximately uniformly over the history, retaining its earliest and most recent views.

A.2 OPTIMIZATION DETAILS

Table 4 summarizes the optimization settings and compute used for the full-model training run.

Table 4: Optimization and training configuration for the full model.

Item	Setting
Learning rate	1×10^{-5}
LR scheduler	Cosine
Training epochs	1
Training hardware	8 NVIDIA H100 GPUs
Per-device batch size	4
Gradient accumulation steps	2
Effective batch size	64 ($4 \times 2 \times 8$)
Training wall time	Approximately 80 hours
Precision	BF16
Gradient checkpointing	Enabled
Maximum gradient norm	0.5
Attention implementation	FlashAttention-2
DeepSpeed configuration	ZeRO-2
Data loader workers	8
λ_{prog}	0.1
λ_{Δ}	0.5
λ_{act}	1.0
λ_{align}	0.1
λ_{mono}	0.05
Progress Smooth-L1 β	0.05
Increment Smooth-L1 β	0.05
Alignment target width σ_{align}	0.18
Positional width bounds $(\rho_{\min}, \rho_{\max})$	$(0.04, 0.40)$
Positional scale γ	Learned, initialized to 1; unconstrained
Monotonic tolerance	0.02
Maximum progress increment $\Delta_{\max}$	1.0

A.3 Action Vocabulary

We adopt the same ten-symbol action vocabulary as NaVIDA (Zhu et al., 2026):

$$\mathcal{A} = \{\textsc{Stop}, \mathrm{F}_{25}, \mathrm{F}_{50}, \mathrm{F}_{75}, \mathrm{L}_{15}, \mathrm{L}_{30}, \mathrm{L}_{45}, \mathrm{R}_{15}, \mathrm{R}_{30}, \mathrm{R}_{45}\}. \tag{10}$$

The subscripts denote forward distance in centimeters or turn angle in degrees. A 50 or 75 cm forward symbol expands into two or three 25 cm primitive actions; likewise, a 30- or 45-degree turn expands into two or three 15-degree primitives of the same direction. Each policy call predicts three action symbols. Training pads trailing slots with STOP; the action cross-entropy averages over all three slots, including STOP padding. At evaluation time, the first predicted symbol is executed and the policy replans after the next observation.

A.4 Progress Targets and Training Losses

For a primitive-action demonstration $y_{1:T}$, let s_t and s_t^+ denote the state boundaries before and after the current supervised chunk. We use

$$f(s) = \frac{\sum_{j<s} \mathbb{I}[y_j = \textsc{Forward}]}{\max\Big(1, \sum_{j=1}^{T} \mathbb{I}[y_j = \textsc{Forward}]\Big)}, \quad p_t^\star = f(s_t), \quad \Delta p_t^\star = f(s_t^+) - f(s_t). \tag{11}$$

The count is over primitive forward actions, including those obtained by expanding a forward symbol. In-place turns do not increase this target.

For the positional score in Eq. 3, we predict $\mu_t = \sigma(h_\mu(\mathbf{r}_t))$ and $\rho_t = 0.04 + 0.36\,\sigma(h_\rho(\mathbf{r}_t))$. The scalar γ is learned and initialized to one. Invalid instruction positions are masked before normalization.

We supervise $\hat{p}_t$ and the auxiliary increment prediction with smooth-L_1 losses. The alignment target $q_{t,i}$ is obtained by softmax-normalizing $-(x_{t,i} - p_t^\star)^2/(2\sigma^2_{\text{align}})$ over valid tokens; its cross-entropy with $\pi_{t,i}$ is $\mathcal{L}_{\text{align}} = -\sum_i q_{t,i} \log \pi_{t,i}$. Action logits receive per-slot cross-entropy. For an ordered early–late state pair (e, l) from the same demonstration, the monotonicity penalty is $\max(0, \hat{p}_e - \hat{p}_l - \epsilon)$. Pair members contribute to this temporal loss but are excluded from the action cross-entropy. The complete objective in Eq. 9 uses the coefficients and tolerance listed in Table 4.

A.5 Ablation Definitions

PositionOnly zeroes the content-based matching logits but retains the attended instruction representation. ScalarOnly predicts a scalar from the route state and uses only its embedding as the progress source; it omits instruction alignment and its loss. Full retains both the progress-query states and the fused context as decoder sources. NoMono sets the temporal regularization weight to zero.

A.6 Evaluation Uncertainty

Controlled variants use the same controller and environment configuration. Results are paired by episode. We estimate 95% confidence intervals using 10,000 bootstrap samples over 613 (scene, trajectory) clusters, keeping language annotations of the same physical route grouped. These intervals quantify evaluation-route uncertainty rather than training variation.

For Full versus ActionOnly, the reported differences are 7.56 SR, 7.10 SPL, and 5.98 OSR points, with route-cluster 95% confidence intervals of [4.79, 10.33], [4.54, 9.66], and [3.15, 8.75], respectively.

B Additional Experimental Results

B.1 Instruction Length Breakdown

We additionally report the SR improvement of the complete model over ActionOnly after grouping evaluation instructions by length. This breakdown is intended as a descriptive complement to the main controlled results.

Table 5: SR improvement over ActionOnly across instruction-length groups on the controlled R2R-CE evaluation.

Instruction length	SR improvement (points)
≤ 20 words	+8.54
21–40 words	+6.45
> 40 words	+10.29

B.2 Goal-reaching Outcome Breakdown

The aggregate outcome counts provide a complementary view of the controlled comparison. We distinguish successful termination, episodes that never enter the goal region, and episodes that enter it but terminate unsuccessfully.

Table 6: Goal-reaching outcome decomposition on the 1,839 controlled R2R-CE Val-Unseen episodes. Percentages are over all evaluation instructions. The categories correspond to $\text{SR} = 1$, $\text{OSR} = 0$, and $\text{OSR} = 1, \text{SR} = 0$, respectively.

Outcome	ActionOnly	Aligned-VLN
Successful termination	585 (31.81%)	724 (39.37%)
Goal never reached	1081 (58.78%)	971 (52.80%)
Goal reached, unsuccessful termination	173 (9.41%)	144 (7.83%)

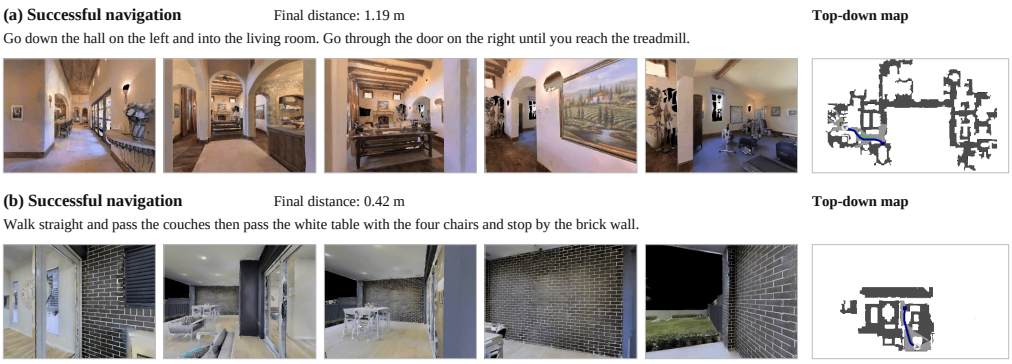

Figure 5: Two successful closed-loop navigation episodes on R2R-CE Val-Unseen using the Qwen2.5-VL-3B FULL checkpoint. Rows (a) and (b) show episodes 1725 and 902, with final goal distances of 1.19 m and 0.42 m, respectively. The five RGB views in each row are ordered chronologically. The final top-down maps are generated by the evaluator; green shows the shortest path and blue the executed route.

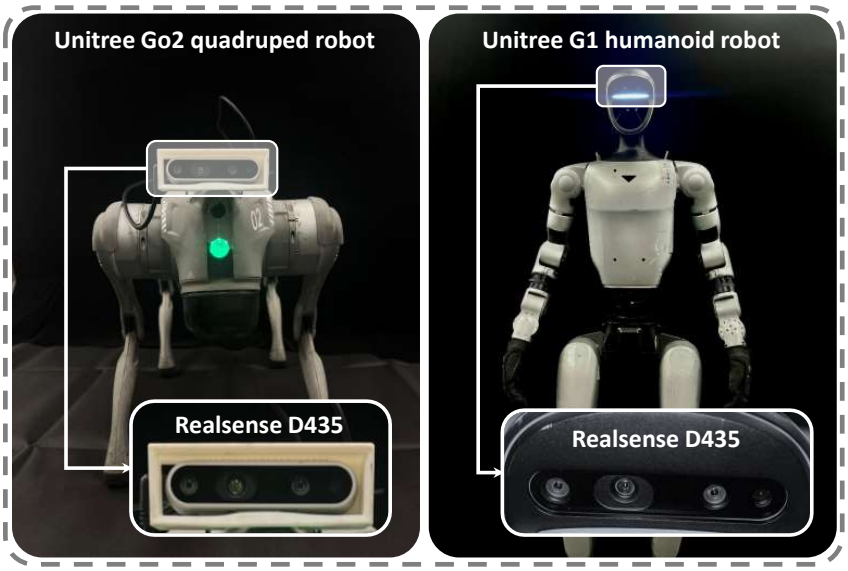

Figure 6: Physical robot platforms and sensing setup.

B.3 QUALITATIVE SIMULATION EXAMPLES

Figure 5 shows two successful closed-loop episodes on R2R-CE Val-Unseen. Each row pairs the original instruction with five egocentric RGB views sampled in execution order and a final top-down map. The first route passes from a hallway into a furnished living area; the second passes couches and a dining table before approaching a brick wall. The views illustrate the room transitions and visual landmarks encountered in these simulated indoor scenes.

C REAL-WORLD DEPLOYMENT DETAILS

This section documents the physical platforms and sensor setup used for the real-world demonstrations in Figure 4. The physical deployment uses Unitree Go2 and Unitree G1 robot platforms with RealSense D435 sensing. Figure 6 shows the robot platforms and sensing setup, while Figure 4 presents the closed-loop trajectories.