LatentPilot: Scene-Aware Vision-and-Language Navigation by Dreaming Ahead with Latent Visual Reasoning

Haihong Hao[1], Lei Chen[1], Mingfei Han[2], Changlin Li[3], Dong An[4], Yuqiang Yang[5], Zhihui Li[1], and Xiaojun Chang[1]*

[1] University of Science and Technology of China, Hefei, China
haohaihong@mail.ustc.edu.cn, xjchang@ustc.edu.cn
[2] MBZUAI, Abu Dhabi, United Arab Emirates
[3] Stanford University, Stanford, CA, USA
[4] Amap, Alibaba Group, Hangzhou, China
[5] Shanghai AI Laboratory, Shanghai, China

Abstract. Existing vision-and-language navigation (VLN) models primarily reason over past and current visual observations, while largely ignoring the future visual dynamics induced by actions. As a result, they often lack an effective understanding of the causal relationship between actions and how the visual world changes, limiting robust decision-making. Humans, in contrast, can "imagine" the near future by leveraging action–dynamics causality, which improves both environmental understanding and navigation choices. Inspired by this capability, we propose LatentPilot, a new paradigm that exploits future observations during training as a valuable data source to learn action-conditioned visual dynamics, while requiring no access to future frames at inference. Concretely, we propose a flywheel-style training mechanism that iteratively collects on-policy trajectories and retrains the model to better match the agent's behavior distribution, with an expert takeover triggered when the agent deviates excessively. LatentPilot further learns visual latent tokens without explicit supervision; these latent tokens attend globally in a continuous latent space and are carried across steps, serving as both the current output and the next input, which enabling the agent to "dream ahead" and reason about how actions will affect subsequent observations. Experiments on R2R-CE, RxR-CE, and R2R-PE benchmarks achieve new SOTA results, and real-robot tests across diverse environments demonstrate LatentPilot's superior understanding of environment–action dynamics in scene.

Keywords: VLN· VLM · Latent Visual Reasoning

1 Introduction

Vision-and-Language Navigation (VLN) requires an embodied agent to move through a 3D environment by following natural-language instructions and reach-

* Corresponding author.

ing a target location [3,13,27,28]. Solving VLN demands more than aligning language with the current visual observation. The agent must accumulate spatial memory, interpret scene semantics, and correct itself under uncertainty during sequential decision making. Crucially, VLN is not a static "perceive then act" problem. It is an interactive process in which every action changes what will be observed next. Therefore, robust navigation is tightly linked to the ability to understand the environment and to anticipate how it will evolve under the agent's actions, namely an ability to imagine near-future observations.

Most existing VLN approaches formulate navigation as conditional policy learning [3,27]. Given an instruction and past or current observations, the model predicts the next action, typically trained via imitation learning on expert trajectories [41] that fit "what to do now." As shown in Fig. 1, although training trajectories inherently contain rich future frames that reflect the consequences of actions, mainstream training pipelines usually treat them only as later observations in a sequence. They are seldom used as explicit supervision to shape representations that are predictive and anticipatory at the current step. A complementary line of work equips agents with explicit lookahead via world models or imagination modules that predict or synthesize future observations and then plan, rerank, or prompt decisions accordingly (e.g., Pathdreamer [24], future-view generation objectives [30], diffusion-based imagination for VLN [18, 19], and scene-imagination prompting pipelines [36,68], as well as world-model-based VLN-CE frameworks [59]). While effective, these approaches typically externalize imagination as separate predictors, planners, or sampling procedures, introducing additional computation and exposing the agent to compounding prediction errors and policy–model mismatch. In contrast, we argue that it is beneficial to internalize imagination: distilling action–observation causality into the navigator itself so that anticipatory reasoning becomes an amortized, end-to-end capability of the same backbone that makes decisions.

If decisions depend only on past and current observations, policies can become myopic, behaving in a "step and see" manner, and may commit to erroneous branches that are difficult to recover from in complex layouts. In contrast, if a model can anticipate how a candidate action will shape near-future observations, it becomes more likely to avoid inefficient exploration, reduce collisions and backtracking, and improve overall stability.Although future observations are not available at decision time, they are fully recorded in offline training trajectories. This raises a central question: can we use these recorded future observations during training to learn how actions change future visual inputs, while keeping inference strictly causal?

When moving in the real world, people rarely make decisions purely on what is visible at the moment. Instead, they perform quick mental simulation before acting, for example, whether turning left will reveal a corridor or a doorway, or whether moving forward will lead to an open area or a dead end. This ability is not innate. It is acquired through experience and reflection on the true consequences of past actions, gradually forming an internal forward model that supports pre-action simulation. By comparison, many VLN methods are effective

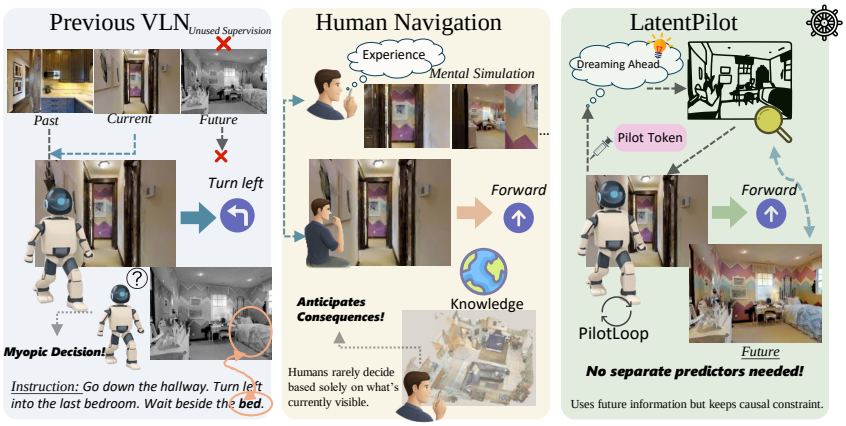

Fig. 1: Prior VLN pipelines optimize next-action prediction from past/current observations, leaving future observations in trajectories largely unused as supervision. LatentPilot uses future observations only as training-time privileged supervision to internalize action-conditioned visual dynamics while inference remains strictly causal and requires neither future frames nor an external world-model rollout.

at extracting cues from past and current observations, yet they rarely model the causal relation between actions, environment change explicitly. Fig. 1 suggests a more natural direction. During training, trajectories already include what the agent actually sees after taking actions. If these future observations are used as supervision, the "hindsight" consequences can be converted into "foresight" representations, allowing the model to learn more anticipatory decision making without requiring access to any future images at test time.

Building on this insight, we propose **Latent** Visual Reasoning–based Dreaming ahead **P**olicy for Scene-Aware Vision-and-Language Navigation, **LatentPilot**. It is an end-to-end VLM navigator that maintains a step-propagated visual latent, termed the Pilot Token, to carry compact near-future imagination across decisions. Unlike explicit imagination/world-model pipelines that rely on separate predictors or rollouts at inference [18, 19, 59, 68], LatentPilot amortizes lookahead into the same decision backbone via the step-propagated Pilot Token. The key observation is that while future observations are unavailable before an action is executed at test time, they are naturally recorded in pre-collected navigation trajectories. We therefore treat these future views as training-only privileged supervision, encouraging the model to produce Pilot Tokens that are predictive of the visual evidence it will encounter after acting. Importantly, LatentPilot remains strictly causal at inference: it never accesses future frames, and instead relies on the current observation, the instruction, and the propagated Pilot Token for decision making. To reduce the mismatch between offline training and interactive deployment, we further adopt a flywheel-style learning loop called PilotLoop that repeatedly collects rollouts and updates the model [41].

To summarize, our contributions are as follows:

- LatentPilot framework: We introduce an end-to-end LMM-based navigator with a propagated visual latent (Pilot Token) that internalizes future-aware reasoning within the decision backbone, without requiring any external imagination module, world-model rollout, or planning procedure at inference.
- We introduce a training-only privileged supervision objective that leverages future observations in trajectories to learn action-conditioned visual dynamics in latent space, effectively converting trajectory hindsight into test-time foresight while keeping inference strictly causal. It does not introduce future information leakage at test time.
- We demonstrate consistent gains in both simulator benchmarks like R2R-CE, RxR-CE, and R2R-PE, and real-robot settings, indicating strong generalization and practical potential.

2 Related Work

Vision-and-Language Navigation (VLN) studies embodied agents that navigate in 3D environments by following natural-language instructions and reaching a target location. Early benchmarks such as Room-to-Room (R2R) are built on the discretized viewpoints and navigation graphs of Matterport3D [3,5]. RxR extends this setting with multilingual instructions and denser annotations, further increasing coverage and difficulty [28]. VLN-CE moves evaluation to continuous environments, making the task more deployment-oriented [27]. Large-scale indoor 3D datasets such as HM3D substantially increase scene diversity for training and evaluation [37]. ScaleVLN synthesizes large-scale instruction–trajectory pairs via data generation [53, 57]. VLN-PE systematically characterizes performance degradation caused by physical and visual disparities and provides a more realistic evaluation platform across different robot embodiments [50]. ETPNav performs online topological mapping and separates long-horizon planning from low-level control to improve executability in continuous environments [2]. BEVBert advances multimodal pretraining from a map-centric perspective, strengthening spatially aware representations for navigation [1]. NaVid formulates continuous navigation as video-conditioned next-step planning and outputs actions from streaming visual observations [64], while Uni-NaVid further unifies multiple embodied navigation tasks under a video-based VLA formulation [63].

Some works build world models that synthesize unobserved or future views to support planning or action evaluation, such as Pathdreamer [25] and NavMorph [59]. Other approaches treat future views as learning targets and introduce auxiliary objectives that predict or align with next-step observation semantics, such as future-view image semantics generation [21, 30]. While these methods demonstrate the value of future-aware signals, anticipatory reasoning is often implemented as separate predictors, generative models, or inference-time rollouts and fusion procedures, which may introduce additional computation and policy–model mismatch during closed-loop execution. In contrast, our work aims to integrate this capability into a single end-to-end navigator.

Latent reasoning aims to allocate multi-step computation in continuous latent space rather than emitting long explicit chain-of-thought texts, thereby

reducing token overhead while retaining iterative internal computation. In language models, Think Before introduces learnable pause tokens to trigger extra forward computation without producing visible tokens [12], and Quiet-STaR learns implicit rationales that generalize across text generation settings [60]. Coconut further promotes continuous thought by feeding hidden states back as reasoning states [14]. CoDi distills chain-of-thought into continuous space via self-distillation [43], SIM-CoT highlights potential instability when scaling implicit reasoning tokens [56], and Think Silently, Think Fast reduces reasoning cost through dynamic latent compression [45]. In multimodal large models, latent reasoning has expanded from purely language latent states to vision or joint vision–language latent spaces. LVR performs autoregressive reasoning directly in visual embedding space [29]. Reasoning in the Dark interleaves vision–text latent reasoning steps to reduce reliance on explicit intermediate annotations [6]. CoCoVa leverages continuous cross-modal "thought chains" for iterative multimodal refinement [34], while Latent Implicit Visual Reasoning emphasizes learning visual reasoning tokens without explicit intermediate supervision, allowing the model to discover task-adaptive visual abstractions end-to-end [31]. Collectively, these studies suggest that continuous latents can serve as flexible internal carriers of computation and abstraction, offering a useful perspective for embodied tasks that require memory, prediction, and planning across sequential decisions.

3 Method

3.1 Task Definition

We consider instruction-following Vision-and-Language Navigation in continuous 3D environments as an instruction-conditioned partially observable Markov decision process (POMDP) [23]. Formally, let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \Omega)$, where $s_t \in \mathcal{S}$ denotes the (latent) environment state at time t. Given a natural-language instruction $\mathbf{x} = (w_1, \ldots, w_L)$, the agent receives an egocentric monocular observation $\mathbf{o}_t \in \mathcal{O} \subset \mathbb{R}^{H \times W \times C}$ (with $C = 3$ for RGB and optionally $C = 4$ for RGB-D), and outputs a low-level action $a_t \in \mathcal{A}$. The environment evolves according to

$$s_{t+1} \sim \mathcal{T}(\cdot \mid s_t, a_t), \qquad \mathbf{o}_{t+1} \sim \Omega(\cdot \mid s_{t+1}), \tag{1}$$

where $\mathcal{T}$ and Ω are the state transition and observation models, respectively. This interaction repeats over time, producing a trajectory

$$\tau = \big(\mathbf{x},\, \mathbf{o}_{1:T},\, a_{1:T}\big), \qquad \mathbf{o}_{1:t} \triangleq (\mathbf{o}_1, \ldots, \mathbf{o}_t), \quad a_{1:t} \triangleq (a_1, \ldots, a_t), \tag{2}$$

and terminates when $a_t = \texttt{STOP}$ or $t = T_{\max}$. We adopt the standard discrete action set:

$$\mathcal{A} = \{\texttt{FWD}, \texttt{LEFT}, \texttt{RIGHT}, \texttt{STOP}\}, \tag{3}$$

where each action corresponds to a small motion primitive in continuous space. Our goal is to learn a policy $\pi_\theta(a_t \mid \mathbf{x}, \mathbf{o}_{1:t})$ such that its rollouts reach the goal region $\mathcal{G} \subset \mathcal{S}$ and terminate by predicting $\texttt{STOP}$ at the target.

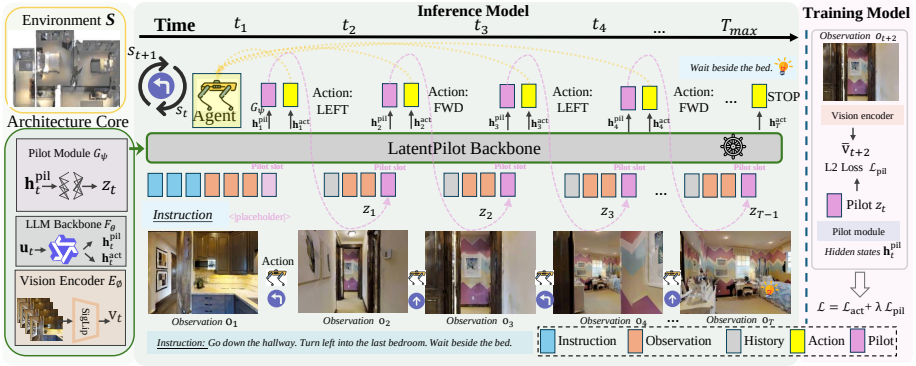

Fig. 2: Overview of LatentPilot. The vision encoder extracts visual tokens from the current observation, the LLM backbone predicts the action, and a lightweight Pilot module updates a propagated Pilot Token that is cached and reused as the Pilot slot input at the next step, enabling strictly causal dreaming-ahead.

3.2 LatentPilot

As illustrated in Fig. 2, LatentPilot is an end-to-end navigator composed of three components: a vision encoder E_ϕ, a lightweight Pilot module G_ψ, and a LLM decision backbone F_θ. The core idea is to maintain a step-propagated continuous latent state $\mathbf{z}_t \in \mathbb{R}^d$, termed the Pilot Token. Pilot Token serves as a compact, persistent internal reasoning state throughout navigation, enabling the policy to carry forward anticipatory cues without generating verbose textual rationales.

Pilot initial. Given the egocentric observation $\mathbf{o}_t \in \mathbb{R}^{H\times W\times C}$ at step t, we adopt a SigLIP [62]-initialized vision encoder to obtain a sequence of visual tokens

$$\mathbf{v}_t = E_\phi(\mathbf{o}_t) \in \mathbb{R}^{N_v \times d}, \tag{4}$$

where N_v is the number of visual tokens and d matches the hidden size of the backbone. We initialize the decision backbone F_θ from the 7B LLaVA-Video model [67]. At each step, the backbone consumes a concatenated multimodal sequence $\mathbf{u}_t$ consisting of instruction tokens, current visual tokens, and the Pilot Token $\mathbf{z}_{t-1}$carried from the previous step:

$$\mathbf{u}_t \triangleq \Big[\ \mathrm{Tok}(\mathbf{x})\ ;\ \mathbf{v}_t\ ;\ \texttt{PILOT}(\mathbf{z}_{t-1})\ \Big]. \tag{5}$$

Here $\texttt{PILOT}(\cdot)$ denotes placing $\mathbf{z}_{t-1}$ at a dedicated token positions, called Pilot slot. Pilot slot is implemented by a special placeholder token `<|placeholder|>` whose embedding provides $\mathbf{z}_0$ at the first step. Unlike discrete language tokens, $\mathbf{z}_t$ lies in a continuous latent space and is carried across steps, making it a natural vehicle for long-horizon, trajectory-level internal computation.

Pilot propagation. The LLM backbone computes hidden states with causal attention,

$$\mathbf{H}_t = F_\theta(\mathbf{u}_t) \in \mathbb{R}^{N\times d}. \tag{6}$$

Let $\mathbf{h}_t^{\text{act}}$ and $\mathbf{h}_t^{\text{pil}}$ denote the action and Pilot-position hidden states from $\mathbf{H}_t$, respectively. We obtain the low-level action distribution via a linear action head,

$$\pi_\theta(a_t \mid \mathbf{x}, \mathbf{o}_t, \mathbf{z}_{t-1}) = \text{Softmax}\big(\mathbf{W}_a\, \mathbf{h}_t^{\text{act}}\big), \quad a_t \in \mathcal{A}. \tag{7}$$

The Pilot Token is then updated by a lightweight projection-only Pilot module G_ψ:

$$\mathbf{z}_t = G_\psi\Big(\mathbf{h}_t^{\text{pil}}\Big) \in \mathbb{R}^d, \tag{8}$$

where G_ψ is implemented as a simple linear layer, with only a few parameters that projects the Pilot-position hidden state back to the latent space. Here "action head" is simply the VLM's native linear projection, only a naming distinction from the Pilot Token prediction branch. Importantly, G_ψ does not introduce any additional observations or external predictive components (e.g., world-model rollouts); it merely performs a compact latent projection to support cross-step propagation. Overall, one navigation step can be summarized as

$$(a_t, \mathbf{z}_t) = \mathcal{F}_{\theta,\psi}\big(\mathbf{x}, \mathbf{o}_t, \mathbf{z}_{t-1}\big), \tag{9}$$

where $\mathcal{F}_{\theta,\psi}$ denotes the overall step transition induced by the model: given the instruction, the current observation, and the previous Pilot Token, it outputs the next action and the updated Pilot Token. The action a_t is executed in the environment to yield the next observation $\mathbf{o}_{t+1}$, while $\mathbf{z}_t$ is fed into the next step through the Pilot slot. This recurrent latent propagation links stepwise decisions into a coherent, trajectory-level internal reasoning process within a single backbone. So far we have described the architecture and information flow of LatentPilot. In the next subsection, we introduce how training-time supervision derived from future observations shapes $\mathbf{z}_t$ into an anticipatory latent that captures action-conditioned visual dynamics, while keeping test-time inputs unchanged. $\bar{\mathbf{v}}_{t+1}$ and $\mathbf{z}_t$ is supervised toward $\bar{\mathbf{v}}_{t+2}$.

3.3 Training with Future-Privileged Supervision

We train LatentPilot in a flywheel-style closed-loop procedure (Fig. 3), where data collection and model updates iterate in multiple rounds. Each round cycles through four stages: (i) *rollout collection* under the current policy, (ii) *deviation-aware correction* where an expert takes over when the agent drifts too far from a reference trajectory, (iii) *privileged target generation* from the collected rollouts, and (iv) *fine-tuning* with a joint objective. The key supervision signal comes from future observations recorded in these trajectories, used as training-only privileged information in the spirit of LUPI [46].

Flywheel data collection with expert takeover. Let $\mathcal{D} = \{\tau^{(i)}\}_{i=1}^{N}$ denote the trajectory buffer collected in the current flywheel round, where

$$\tau = \big(\mathbf{x},\, \mathbf{o}_{1:T},\, a_{1:T}^{\text{col}}\big). \tag{10}$$

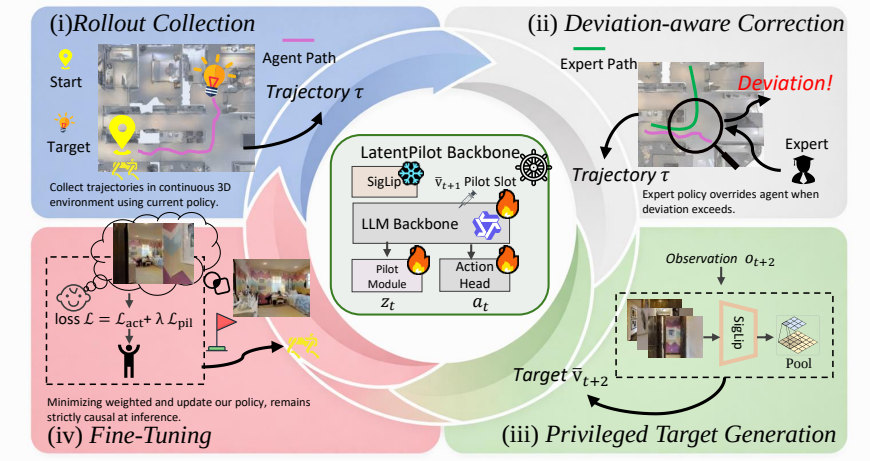

Fig. 3: PilotLoop: flywheel-style closed-loop training. Each round iterates rollout collection, expert takeover for deviation-aware correction, future-privileged target construction, and fine-tuning with joint action imitation and Pilot supervision.

Here a_t^{col} is the collected action at step t. By default it is produced by the current model, while an expert policy may override it when the deviation from the reference becomes large [41]. We then unroll LatentPilot along τ using the same recurrent interface as Eq. (9) and optimize a standard action cross-entropy loss, here actions are decoded by the backbone's native LLM output projection.

Two-step future privilege. A central design choice is to explicitly distinguish two future steps: $t{+}1$ as input, $t{+}2$ as target. At training step t, we treat the next observation $\mathbf{o}_{t+1}$ as a privileged input to the Pilot slot, and the next-next observation $\mathbf{o}_{t+2}$ as a privileged target for supervising the predicted Pilot Token. Concretely, we first compress an observation into a single latent vector by mean pooling the vision tokens:

$$\bar{\mathbf{v}}_{t+1} \triangleq \text{Pool}(E_\phi(\mathbf{o}_{t+1})) \in \mathbb{R}^d. \tag{11}$$

During training, we fill the Pilot slot with the one-step future latent $\bar{\mathbf{v}}_{t+1}$ as training-only privileged input. So multimodal input sequence $\mathbf{u}_t$ consisting of instruction tokens, current visual tokens, and one-step future:

$$\mathbf{u}_t^{\text{tr}} \triangleq \Big[\ \text{Tok}(\mathbf{x})\ ;\ \mathbf{v}_t\ ;\ \texttt{PILOT}(\bar{\mathbf{v}}_{t+1})\ \Big], \tag{12}$$

where $\mathbf{v}_t = E_\phi(\mathbf{o}_t)$ is the visual token sequence . A forward pass yields hidden states $\mathbf{H}_t = F_\theta(\mathbf{u}_t^{\text{tr}})$ (Eq. (6)). Let $\mathbf{h}_t^{\text{act}}$ and $\mathbf{h}_t^{\text{pil}}$ denote the action and Pilot-position hidden states from $\mathbf{H}_t$. We supervise the action via:

$$p_\theta(\cdot) = \text{Softmax}\big(\mathbf{W}_{\text{LM}}\mathbf{h}_t^{\text{act}}\big)\,, \qquad \mathcal{L}_{\text{act}} \triangleq -\sum_{t=1}^{T} \log p_\theta\big(a_t^{\text{col}} \mid \mathbf{x}, \mathbf{o}_t, \bar{\mathbf{v}}_{t+1}\big)\,. \tag{13}$$

To shape the Pilot Token into an anticipatory latent, we regress the predicted Pilot Token to match the two-step future latent. Concretely, the Pilot module projects the Pilot-position hidden state into the continuous latent space and the privileged target is defined from the two-step future observation:

$$\mathbf{z}_t \triangleq G_\psi\left(\mathbf{h}_t^{\text{pil}}\right) \in \mathbb{R}^d, \qquad \mathcal{L}_{\text{pil}} \triangleq \sum_{t=1}^{T-2} \|\mathbf{z}_t - \bar{\mathbf{v}}_{t+2}\|_2^2 . \tag{14}$$

Overall objective and train–test interface. We fine-tune the model in each flywheel round by minimizing

$$\min_{\theta,\phi,\psi} \mathbb{E}_{\tau\sim\mathcal{D}} \left[\mathcal{L}_{\text{act}}(\tau) + \lambda\, \mathcal{L}_{\text{pil}}(\tau)\right], \tag{15}$$

where λ balances imitation learning and future-privileged supervision and we set 0.1 in practice. Finally, note the train–test interface is simple and strictly causal at evaluation: during training, the Pilot slot is teacher-forced with We train LatentPilot on a mixed corpus of instruction-following trajectories from Matterport3D (MP3D) scenes, including R2R [3], RxR [28], and the EnvDrop-augmented R2R set [44], and further include ScaleVLN trajectories synthesized on HM3D scenes [37, 53]. We first bootstrap the policy with imitation learning using Habitat's shortest-path follower as the expert policy [27, 42], and then perform flywheel-style closed-loop fine-tuning.

Inference with stored pilot. During inference, the model performs a purely recurrent rollout a `PilotCache`, without accessing any future frames or computing additional Pilot representations. Instead of generating a separate Pilot signal, the model directly reuses the previously predicted Pilot Token through a read–write cache mechanism. Specifically, after step t–1 the model outputs $\mathbf{z}_{t-1}$, which is stored in the cache and used as the Pilot-slot input at step t (i.e., $\texttt{PILOT}(\mathbf{z}_{t-1})$ in $\mathbf{u}_t$). The new output $\mathbf{z}_t$ then overwrites the cache for the next step. This read–write update repeats until `STOP`, enabling cross-step Pilot propagation using only the current observation and the cached latent. Importantly, no separate Pilot computation is required at test time as the cached latent itself serves as the Pilot representation. As a result, cross-step Pilot propagation is achieved using only the current observation and a single cached latent.

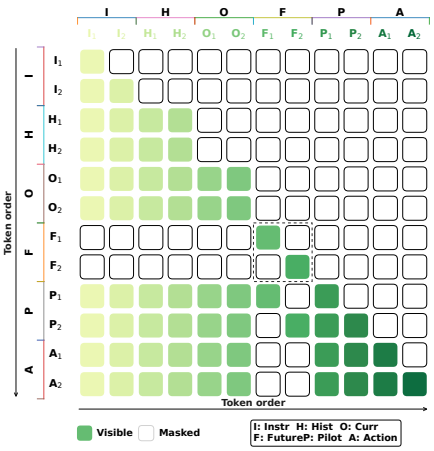

Fig. 4: Visibility matrix of LatentPilot.

Table 1: Comparison with state-of-the-art methods on VLN-CE R2R and RxR **Val-Unseen** splits. Pano, Odo and D respectively represent panoramic view, odometry and depth, SRGB denotes monocular RGB.

Method	Observation				R2R Val-Unseen				RxR Val-Unseen			
	Pano.	Odo.	D.	SRGB	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
CMA [15]	✓	✓	✓		6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0
VLN-BERT [15]	✓	✓	✓		5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7
Sim2Sim [26]	✓	✓	✓		6.07	52.0	43.0	36.0	–	–	–	–
Ego^2-Map [16]	✓	✓	✓		5.54	56.0	47.0	41.0	–	–	–	–
DreamWalker [48]	✓	✓	✓		5.53	59.0	49.0	44.0	–	–	–	–
GridMM [51]	✓	✓	✓		5.11	61.0	49.0	41.0	–	–	–	–
ETPNav [2]	✓	✓	✓		4.71	65.0	57.0	49.0	5.64	54.7	44.8	61.9
ScaleVLN [53]	✓	✓	✓		4.80	–	55.0	51.0	–	–	–	–
InstructNav [33]	✓	✓	✓	✓	6.89	–	31.0	24.0	–	–	–	–
R2R-CMTP [8]	✓	✓	✓		7.90	38.0	26.4	22.7	–	–	–	–
LAW [38]		✓	✓	✓	6.83	44.0	35.0	31.0	10.90	8.0	8.0	38.0
CM2 [11]		✓	✓	✓	7.02	41.5	34.3	27.6	–	–	–	–
WS-MGMap [9]		✓	✓	✓	6.28	47.6	38.9	34.3	–	–	–	–
Sim2Real [52]		✓	✓	✓	5.95	55.8	44.9	30.4	8.79	25.5	18.1	–
Seq2Seq [27]			✓	✓	7.77	37.0	25.0	22.0	12.10	13.9	11.9	30.8
CMA [27]			✓	✓	7.37	40.0	32.0	30.0	–	–	–	–
NaVid [64]				✓	5.47	49.1	37.4	35.9	–	–	–	–
AO-Planner [7]	✓		✓		5.55	59.0	47.0	33.0	7.06	43.3	30.5	50.1
COSMO [66]	✓				–	56.0	47.0	40.0	–	–	–	–
MapNav [65]				✓	4.93	53.0	39.7	37.2	–	–	–	–
NaVid-4D [32]			✓	✓	5.99	55.7	43.8	37.1	–	–	–	–
NavMorph [59]			✓	✓	5.75	56.9	47.9	33.2	8.85	30.8	22.8	44.2
Uni-NaVid [63]				✓	5.58	53.3	47.0	42.7	6.24	48.7	40.9	–
NaVILA [10]				✓	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8
StreamVLN [55]				✓	4.98	64.2	56.9	51.9	6.22	52.9	46.0	61.9
JanusVLN [61]				✓	4.78	65.2	60.5	56.8	6.06	56.2	47.5	62.1
LatentPilot (Ours)				✓	4.41	66.3	62.0	58.0	5.19	58.2	49.9	67.5

4 Experiments

We conduct experiments to answer the following questions: (1) Can we use these recorded future observations during training to learn how actions change future visual inputs, while keeping inference strictly causal? (2)What are the benefits of integrating future information and "imagination" directly into the decision-making LLM, compared to relying on external predictive models or auxiliary modules? We evaluate our method across representative robot embodiments, including **humanoid robots**, **wheeled platforms**, and **quadruped robots**, to demonstrate the broad applicability of LatentPilot. VLN-PE enables controlled experiments with simulated humanoid robots, and our real-world deployments further showcase LatentPilot on wheeled and quadruped robots.

Table 2: Evaluation Metrics on VLN-PE benchmark with physical locomotion controller. +: model is first trained on Habitat and fine-tuned on VLN-PE. †: model is trained with data augmentation.

Method	R2R Validation Seen						R2R Validation Unseen					
	NE↓	FR↓	StR↓	OS↑	SR↑	SPL↑	NE↓	FR↓	StR↓	OS↑	SR↑	SPL↑
Train-free Map-based Exploration and Navigation												
VLMaps [17]	–	–	–	–	–	–	6.98	23.00	0.00	20.00	20.00	12.70
Train on VLN-PE												
Seq2Seq [27]	7.73	22.19	3.04	30.55	19.60	15.67	7.91	19.67	3.71	27.62	15.89	12.58
Seq2Seq+ [27]	7.54	26.11	5.59	31.93	19.58	15.13	7.64	21.82	5.12	30.47	18.13	14.06
CMA [27]	7.59	23.71	3.19	34.94	21.58	16.10	7.98	22.64	3.27	33.11	19.15	14.05
CMA+ [27]	7.14	23.56	3.50	36.17	25.84	21.75	7.26	21.75	3.27	31.40	22.12	18.65
RDP [50]	6.76	27.51	1.82	38.60	25.08	17.07	6.72	24.57	3.11	36.90	25.24	17.73
Zero-shot Transfer Evaluation from VLN-CE												
Seq2Seq [27] †	7.62	20.21	3.04	19.30	15.20	12.79	7.18	18.04	3.04	22.42	16.48	14.11
CMA [27]†	7.37	20.06	3.95	18.54	16.11	14.64	7.09	17.07	3.79	20.86	16.93	15.24
NaVid [64]	6.20	11.25	0.46	24.32	21.58	17.45	5.94	8.61	0.45	27.32	22.42	18.58
DualVLN [54]	4.13	17.78	1.82	62.31	58.97	47.78	4.66	12.32	2.23	55.90	51.60	42.49
Ours	**4.10**	**11.09**	**1.08**	**63.87**	**59.01**	**49.65**	**4.33**	**10.65**	**0.97**	**60.31**	**56.42**	**47.74**

4.1 Environment and Metrics.

We evaluate LatentPilot in both the standard continuous VLN setting and a physically realistic navigation setting. For continuous navigation, we follow the VLN-CE [27] and report results on the Val-Unseen splits of **R2R-CE** [3] and **RxR-CE** [28] and executed in the Habitat simulator [42]. Following prior work, we use the standard VLN metrics. Navigation Error **(NE)** measures the final geodesic distance (in meters) from the agent's stopping location to the goal. Success Rate **(SR)** is the fraction of episodes where the agent stops within 3 meters of the goal. Oracle Success Rate **(OSR)** considers the closest point along the trajectory as the stopping point. Success weighted by Path Length **(SPL)** [3] accounts for both success and path efficiency. Normalized Dynamic Time Warping **(nDTW)** [20] measures trajectory fidelity to the reference path.

We additionally report results on **VLN-PE** [50] follow DualVLN [54]. VLN-PE evaluation is executed on the physically realistic simulation stack built on NVIDIA Isaac Lab [35], which models robot dynamics and locomotion execution imperfections. We report four primary VLN metrics: NE, SR, OS and SPL. To explicitly measure physical robustness, we further report Fall Rate **(FR)** measures the frequency of robot falls and Stuck Rate **(StR)** occurrences where the agent is unable to move. For humanoid evaluation in VLN-PE, we adopt the Unitree H1 embodiment, a full-size humanoid platform, about 1.8 m tall and equipped with depth camera [40] in simulator. We perform on-policy rollout collection using 16×RTX 4090 GPUs and training using 8×NVIDIA A100 GPUs.

Table 3: Ablation study of supervision modality for the Pilot Token. NaN removes any extra Pilot supervision and optimizes only the action cross-entropy, serving as a baseline that tests whether the propagated latent becomes predictive by itself.

Modality	R2R Validation Unseen				R2R Validation Seen			
	SR↑	SPL↑	NE↓	OS↑	SR↑	SPL↑	NE↓	OS↑
NaN	51.7	47.1	5.3	57.0	53.1	46.9	4.6	60.7
3D	50.3	45.6	5.4	55.8	51.5	45.1	4.9	59.9
Text	53.2	48.7	5.2	59.6	54.9	48.6	4.5	63.9
Vision(ours)	62.0	58.0	4.4	66.3	65.1	59.6	4.3	71.6

4.2 Main result

Results on VLN-CE benchmark. Table 1 summarizes results on the VLN-CE benchmarks. We compare against classic recurrent baselines (Seq2Seq/CMA) as well as stronger pipelines that rely on richer observations such as panoramic views, odometry, and depth (e.g., ETPNav and ScaleVLN) [2, 27, 53]. Despite using only monocular RGB, LatentPilot achieves the strongest overall performance, surpassing prior single-RGB VLM navigators such as NaVid [64] and remaining competitive with several sensor-heavy methods. LatentPilot compares favorably with existing approaches across standard VLN benchmarks while requiring a simpler observation setup, supporting the motivation that internalizing action-conditioned foresight into the decision backbone can reduce myopic "step-and-see" behavior.

Results on VLN-PE benchmark. Table 2 reports results on VLN-PE [50], which evaluates continuous navigation under physically realistic humanoid locomotion with explicit robustness metrics (falls and deadlocks). Across all baselines our method achieves the best overall performance, especially on the challenging Val-Unseen split. LatentPilot not only more accurate at reaching goals but also more stable and safer under realistic execution noise.

4.3 Ablation Study

Supervision modality for the Pilot Token. We keep the LatentPilot architecture and inference interface unchanged and vary only the supervision signal used to shape the Pilot Token (Table 3). NaN removes any extra Pilot supervision and optimizes only the action cross-entropy, serving as a baseline that tests whether the propagated latent becomes predictive by itself. 3D uses geometry-aware features extracted by a 3D foundation model (VGGT [49]) as the privileged target, and Text uses textual descriptions of future views generated by a strong VLM (Qwen2.5-VL [4]) and then tokenized as supervision. We find that supervision choice matters substantially: proxy targets such as 3D embeddings or text provide limited improvements and can even degrade performance.

Internalized imagination vs. plug-in world models. To further answer Question (2), we compare LatentPilot with a representative class of *external* attach a video world model [47, 58] to predict future views and then use these

Table 4: Internalized future supervision vs. plug-in video world models on R2R. T/Act denotes latency per action; Mem denotes peak GPU memory.

World Model	T/Act (ms)↓	Mem (GB)↓	R2R Val-Seen				R2R Val-Unseen			
			SR↑	SPL↑	NE↓	OS↑	SR↑	SPL↑	NE↓	OS↑
Wan2.1 1.3B	2040	41.5	56.6	51.7	5.1	64.5	53.9	48.6	5.3	62.4
CogVideoX1.5 5B	5460	32.4	59.3	53.9	4.9	66.8	–	–	–	–
V-JEPA2 plug-in	143	23.6	60.8	54.9	4.5	68.2	58.1	52.5	4.9	65.6
Ours	**130**	**22.8**	**65.1**	**59.6**	**4.3**	**71.6**	**62.0**	**58.0**	**4.4**	**66.3**

predictions to guide action selection (Table 4). Although such plug-in designs can provide explicit foresight, they also introduce an additional generative backbone at inference time, substantially increasing per-step latency and memory footprint. In contrast, LatentPilot *internalizes* the action-observation dynamics into the decision backbone via training-only future supervision, so inference remains a single forward pass with a lightweight latent propagation. As a result, our approach achieves markedly lower time-per-action and peak memory, while also delivering stronger navigation performance. Integrated imagination are not only improved success but also faster and more resource-efficient deployment.

Effect of flywheel iterations (PilotLoop). Figure 5 plots performance on R2R Val-Unseen after each PilotLoop round. We observe a clear, steady improvement in both success (SR) and path efficiency (SPL) as the flywheel progresses, with the largest gains appearing in the early rounds and gradually saturating later. The expert-takeover mechanism stabilizes the loop by preventing severe drift when the policy deviates too far, echoing the dataset-aggregation intuition in DAgger-style training [41].

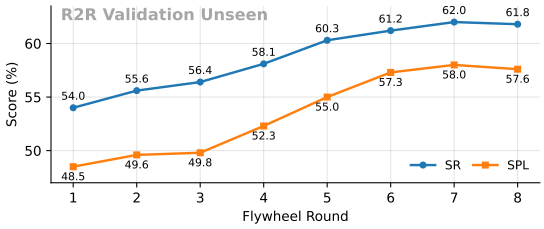

Fig. 5: Performance over PilotLoop.

Latent collapse analysis. A common failure mode of latent-token methods is latent collapse, where the latent vectors degenerate into near-constant, uninformative representations. To diagnose this, we collect the propagated Pilot Tokens $\mathbf{z}_t$ from rollouts and project them to 2D using principal component analysis (PCA) [22]. We then color each point by the executed action (`LEFT`, `FWD`, `RIGHT`); `STOP` is excluded due to its low frequency. As shown in Fig. 6, Pilot Tokens

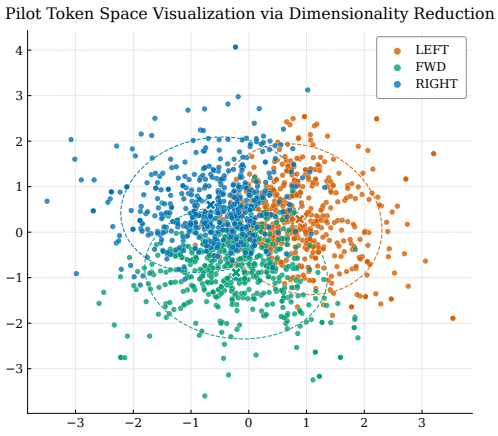

Fig. 6: Latent collapse Analysis via PCA.

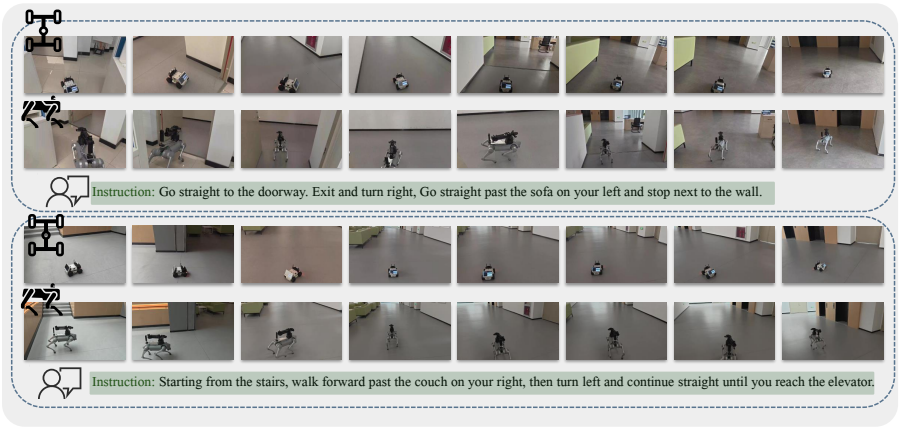

Fig. 7: Real-world Cross-Embodiment Experiments: Wheeled and Quadruped Robots. LatentPilot successfully follows long, multi-step instructions on both Robots.

form action-correlated and structured clusters rather than collapsing to a single mode, indicating that $\mathbf{z}_t$ preserves meaningful variation and remains strongly grounded in control-relevant information. This supports that action grounding together with predictive supervision can maintain a non-degenerate latent space, enabling the Pilot Token to serve as a compact internal state instead of an empty placeholder.

4.4 Real-world Experiments

To further assess practical deployability beyond simulation, we deploy LatentPilot on two mainstream robot embodiments in indoor corridors: a wheeled AgileX LiMO Pro and a quadruped Unitree Go2. LiMO Pro is equipped with LiDAR and a Intel Realsense D435 depth camera [39], while Go2 is a bionic quadruped featuring a wide-angle camera. Both robots run ROS 2 for I/O and control. During execution, the robot streams egocentric observations to a remote workstation with RTX 4090 GPU over a local network; the server performs inference and sends back the predicted navigation action, which is then executed by the robot-specific low-level controller. As shown in Fig. 7, LatentPilot successfully follows long, multi-step instructions on both wheeled and legged platforms, suggesting that the learned Pilot Token provides a deployment-friendly form of internalized "dreaming ahead" without requiring any plug-in world-model rollout.

5 Conclusion

In this work, we presented LatentPilot, an end-to-end VLM-based navigator that internalizes lookahead for vision-and-language navigation. Instead of relying on external imagination modules or world-model rollouts at inference, LatentPilot leverages a simple yet effective principle: although future observations are

unavailable when an action is chosen, they are naturally recorded in offline trajectories and can serve as training-only privileged supervision. By distilling these action-conditioned future visual consequences into a compact Pilot Token, our model learns anticipatory representations that support more stable sequential decision making while keeping inference strictly causal.

6 Acknowledgements

This work was partially supported by New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123100) and by The National Natural Science Foundation of China (NSFC) under no. 62573399 and U25A20530.

References

1. An, D., Qi, Y., Li, Y., Huang, Y., Wang, L., Tan, T., Shao, J.: Bevbert: Multimodal map pre-training for language-guided navigation. arXiv preprint arXiv:2212.04385 (2022)
2. An, D., Wang, H., Wang, W., Wang, Z., Huang, Y., He, K., Wang, L.: Etpnav: Evolving topological planning for vision-language navigation in continuous environments. arXiv preprint arXiv:2304.03047 (2023)
3. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. CoRR **abs/2502.13923** (2025). `https://doi.org/10.48550/ARXIV.2502.13923`, `https://doi.org/10.48550/arXiv.2502.13923`
5. Chang, A.X., Dai, A., Funkhouser, T., Halber, M., Nießner, M., Savva, M., Song, S., Zeng, A., Zhang, Y., Xiao, J.: Matterport3d: Learning from rgb-d data in indoor environments. In: Proceedings of the International Conference on 3D Vision (3DV). pp. 667–676 (2017)
6. Chen, C., Ma, Z., Li, Y., Hu, Y., Wei, Y., Li, W., Nie, L.: Reasoning in the dark: Interleaved vision-text reasoning in latent space. arXiv preprint arXiv:2510.12603 (2025)
7. Chen, J., Lin, B., Liu, X., Ma, L., Liang, X., Wong, K.Y.K.: Affordances-oriented planning using foundation models for continuous vision-language navigation. In: AAAI Conference on Artificial Intelligence (AAAI) (2025)
8. Chen, K., Chen, J.K., Chuang, J., Vazquez, M., Savarese, S.: Topological planning with transformers for vision-and-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

9. Chen, P., Ji, D., Lin, K., Zeng, R., Li, T.H., Tan, M., Gan, C.: Weakly-supervised multi-granularity map learning for vision-and-language navigation. arXiv preprint arXiv:2210.07506 (2022)
10. Cheng, A.C., Ji, Y., Yang, Z., Gongye, Z., Zou, X., Kautz, J., Bıyık, E., Yin, H., Liu, S., Wang, X.: Navila: Legged robot vision-language-action model for navigation. In: Robotics: Science and Systems (RSS) (2025)
11. Georgakis, G., Schmeckpeper, K., Wanchoo, K., Dan, S., Miltsakaki, E., Roth, D., Daniilidis, K.: Cross-modal map learning for vision and language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
12. Goyal, S., Ji, Z., Rawat, A.S., Menon, A.K., Kumar, S., Nagarajan, V.: Think before you speak: Training language models with pause tokens. arXiv preprint arXiv:2310.02226 (2023)
13. Hao, H., Han, M., Li, C., Li, Z., Chang, X.: Conav: Collaborative cross-modal reasoning for embodied navigation. ArXiv **abs/2505.16663** (2025), https://api.semanticscholar.org/CorpusID:278789505
14. Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., Tian, Y.: Training large language models to reason in a continuous latent space. arXiv preprint arXiv:2412.06769 (2024)
15. Hong, Y., Wang, Z., Wu, Q., Gould, S.: 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) (2022)
16. Hong, Y., Zhou, Y., Zhang, R., Dernoncourt, F., Bui, T., Gould, S., Tan, H.: Learning navigational visual representations with semantic map supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
17. Huang, C., Mees, O., Zeng, A., Burgard, W.: Visual language maps for robot navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2023)
18. Huang, Y., Jiang, X., Gao, X., Wu, M., Tu, Z.: Vistav2: World imagination for indoor vision-and-language navigation (2025). https://doi.org/10.48550/arXiv.2512.00041, https://arxiv.org/abs/2512.00041
19. Huang, Y., Wu, M., Li, R., Tu, Z.: Vista: Generative visual imagination for vision-and-language navigation (2025). https://doi.org/10.48550/arXiv.2505.07868, https://arxiv.org/abs/2505.07868
20. Ilharco, G., Jain, V., Ku, A., Ie, E., Baldridge, J.: General evaluation for instruction conditioned navigation using dynamic time warping. arXiv preprint arXiv:1907.05446 (2019)
21. Jin, M., Liao, M., Han, M., Li, Z., Chang, X.: Beyond dense futures: World models as structured planners for robotic manipulation. ArXiv **abs/2603.12553** (2026), https://api.semanticscholar.org/CorpusID:286517739
22. Jolliffe, I.T.: Principal Component Analysis. Springer, 2nd edn. (2002)
23. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence **101**(1–2), 99–134 (1998)
24. Koh, J.Y., Lee, H., Yang, Y., Baldridge, J., Anderson, P.: Pathdreamer: A world model for indoor navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14738–14748 (2021)
25. Koh, J.Y., Lee, H., Yang, Y., Baldridge, J., Anderson, P.: Pathdreamer: A world model for indoor navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 14738–14748 (October 2021),

https://openaccess.thecvf.com/content/ICCV2021/html/Koh_Pathdreamer_A_World_Model_for_Indoor_Navigation_ICCV_2021_paper.html
26. Krantz, J., Lee, S.: Sim-2-sim transfer for vision-and-language navigation in continuous environments. In: European Conference on Computer Vision (ECCV) (2022)
27. Krantz, J., Wijmans, E., Majundar, A., Batra, D., Lee, S.: Beyond the nav-graph: Vision-and-language navigation in continuous environments. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
28. Ku, A., Anderson, P., Patel, R., Ie, E., Baldridge, J.: 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) (2020)
29. Li, B., Sun, X., Liu, J., Wang, Z., Wu, J., Yu, X., Chen, H., Barsoum, E., Chen, M., Liu, Z.: Latent visual reasoning. arXiv preprint arXiv:2509.24251 (2025)
30. Li, J., Bansal, M.: Improving vision-and-language navigation by generating future-view image semantics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
31. Li, K., Shang, C., Karlinsky, L., Feris, R., Darrell, T., Herzig, R.: Latent implicit visual reasoning. arXiv preprint arXiv:2512.21218 (2025)
32. Liu, H., Wan, W., Yu, X., Li, M., Zhang, J., Zhao, B., Chen, Z., Wang, Z., Zhang, Z., Wang, H.: Navid-4d: Unleashing spatial intelligence in egocentric rgb-d videos for vision-and-language navigation. In: IEEE International Conference on Robotics and Automation (ICRA) (2025)
33. Long, Y., Cai, W., Wang, H., Zhan, G., Dong, H.: Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. arXiv preprint arXiv:2406.04882 (2024), accepted by CoRL 2024
34. Ma, J., Zhou, X., Song, Y., Yan, H.: Cocova: Chain of continuous vision-language thought for latent space reasoning. arXiv e-prints pp. arXiv–2511 (2025)
35. Mittal, M., Roth, P., Tigue, J., Richard, A., Zhang, O., Du, P., Serrano-Muñoz, A., Yao, X., Zurbrügg, R., Rudin, N., et al.: Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. arXiv preprint arXiv:2511.04831 (2025)
36. Perincherry, A., Krantz, J., Lee, S.: Do visual imaginations improve vision-and-language navigation agents? (2025)
37. Ramakrishnan, S., et al.: Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. arXiv preprint arXiv:2109.08238 (2021)
38. Raychaudhuri, S., Wani, S., Patel, S., Jain, U., Chang, A.X.: Language-aligned waypoint (LAW) supervision for vision-and-language navigation in continuous environments. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2021)
39. Robotics, A.: Limo pro: Ros2 mobile robot platform (product page). Web page, https://global.agilex.ai/products/limo-pro
40. Robotics, U.: Unitree h1 humanoid robot. https://www.unitree.com/h1/
41. Ross, S., Gordon, G., Bagnell, J.A.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS) (2011)
42. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D.: Habitat: A platform for embodied ai research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
43. Shen, Z., Yan, H., Zhang, L., Hu, Z., Du, Y., He, Y.: Codi: Compressing chain-of-thought into continuous space via self-distillation. arXiv preprint arXiv:2502.21074 (2025)

44. Tan, H., Yu, L., Bansal, M.: Learning to navigate unseen environments: Back translation with environmental dropout. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 3356–3366. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1268, https://aclanthology.org/N19-1268
45. Tan, W., Li, J., Ju, J., Luo, Z., Luan, J., Song, R.: Think silently, think fast: Dynamic latent compression of llm reasoning chains. arXiv preprint arXiv:2505.16552 (2025)
46. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. Neural Networks **22**(5–6), 544–557 (2009)
47. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
48. Wang, H., Liang, W., Van Gool, L., Wang, W.: DREAMWALKER: Mental planning for continuous vision-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
49. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotný, D.: VGGT: visual geometry grounded transformer. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11–15, 2025. pp. 5294–5306. Computer Vision Foundation / IEEE (2025). https://doi.org/10.1109/CVPR52734.2025.00499, https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html
50. Wang, L., Xia, X., Zhao, H., Wang, H., Wang, T., Chen, Y., Liu, C., Chen, Q., Pang, J.: Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9455–9465 (Oct 2025)
51. Wang, Z., Li, X., Yang, J., Liu, Y., Jiang, S.: GridMM: Grid memory map for vision-and-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
52. Wang, Z., Li, X., Yang, J., Liu, Y., Jiang, S.: Sim-to-real transfer via 3d feature fields for vision-and-language navigation. arXiv preprint arXiv:2406.09798 (2024)
53. Wang, Z., Li, J., Hong, Y., Wang, Y., Wu, Q., Bansal, M., Gould, S., Tan, H., Qiao, Y.: Scaling data generation in vision-and-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
54. Wei, M., Wan, C., Peng, J., Yu, X., Yang, Y., Feng, D., Cai, W., Zhu, C., Wang, T., Pang, J., Liu, X.: Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. arXiv preprint arXiv:2512.08186 (2025)
55. Wei, M., Wan, C., Yu, X., Wang, T., Yang, Y., Mao, X., Zhu, C., Cai, W., Wang, H., Chen, Y., Liu, X., Pang, J.: Streamvln: Streaming vision-and-language navigation via slowfast context modeling. arXiv preprint arXiv:2507.05240 (2025)
56. Wei, X., Liu, X., Zang, Y., Dong, X., Cao, Y., Wang, J., Qiu, X., Lin, D.: Sim-cot: Supervised implicit chain-of-thought. arXiv preprint arXiv:2509.20317 (2025)
57. Xia, F., Zamir, A.R., He, Z., Sax, A., Malik, J., Savarese, S.: Gibson env: Real-world perception for embodied agents. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
58. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)

59. Yao, X., Gao, J., Xu, C.: 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) (2025)
60. Zelikman, E., Harik, G., Shao, Y., Jayasiri, V., Haber, N., Goodman, N.D.: Quiet-star: Language models can teach themselves to think before speaking. arXiv preprint arXiv:2403.09629 (2024)
61. Zeng, S., Qi, D., Chang, X., Xiong, F., Xie, S., Wu, X., Liang, S., Xu, M., Wei, X.: Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. arXiv preprint arXiv:2509.22548 (2025)
62. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
63. Zhang, J., Wang, K., Wang, S., Li, M., Liu, H., Wei, S., Wang, Z., Zhang, Z., Wang, H.: Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. In: Robotics: Science and Systems (RSS) (2025)
64. Zhang, J., Wang, K., Xu, R., Zhou, G., Hong, Y., Fang, X., Wu, Q., Zhang, Z., Wang, H.: Navid: Video-based vlm plans the next step for vision-and-language navigation. In: Robotics: Science and Systems (RSS) (2024)
65. Zhang, L., Hao, X., Xu, Q., Zhang, Q., Zhang, X., Wang, P., Zhang, J., Wang, Z., Zhang, S., Xu, R.: 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 (ACL) (2025)
66. Zhang, S., Qiao, Y., Wang, Q., Yan, Z., Wu, Q., Wei, Z., Liu, J.: COSMO: Combination of selective memorization for low-cost vision-and-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
67. Zhang, Y., Wu, J., Li, W., Li, B., Ma, Z., Liu, Z., Li, C.: Llava-video: Video instruction tuning with synthetic data. arXiv preprint arXiv:2410.02713 (2024)
68. Zhao, X., Cai, W., Tang, L., Wang, T.: Imaginenav: Prompting vision-language models as embodied navigator through scene imagination. In: International Conference on Learning Representations (ICLR) (2025), https://openreview.net/forum?id=vQFw9ryKyK