WAM-Nav: Asymmetric Latent World-Action Modeling for Unified Visual Navigation

Ning Yang[1,2], **Yan Huang**[2,3,4,*], **Kaiwen Peng**[2,3], **Ziheng He**[2,3], **Kai Wang**[2,3], **Cui Miao**[5], **Kailin Lyu**[2,3], **Guo Li**[7], **Xiaofeng Wang**[6,7], **Zheng Zhu**[7,*], **Jing Liu**[4], **Nianfeng Liu**[4]

[1]Nanjing University [2]Institute of Automation, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences [4]FiveAges
[5]National University of Defense Technology [6]Tsinghua University [7]GigaAI

Abstract: Visual navigation requires generating executable trajectories under complex geometric and physical constraints. Existing reactive policies that directly map observations to actions lack anticipatory reasoning, limiting their ability to incorporate future perceptual context. While visual imagination offers predictive foresight, conventional modular approaches separate scene prediction from policy learning, which can increase inference cost and sensitivity to prediction errors. To address these limitations, we propose WAM-Nav, a Latent World-Action Model for embodied visual navigation that jointly learns action generation and latent visual foresight, enabling more robust and foresighted navigation decisions while retaining practical inference efficiency. Specifically, WAM-Nav utilizes a shared Diffusion Transformer for asymmetric joint diffusion to concurrently generate long-horizon actions and short-horizon visual foresight, avoiding the need for multi-step autoregressive visual rollouts at inference. To further condition trajectory generation on recent motion, we introduce a dual-stream contextual conditioning mechanism that integrates episode-level ego-motion history with sequential visual observations. Combined with a unified goal alignment module that preserves balanced representations across goal types, WAM-Nav naturally supports Image-Goal, Point-Goal, and No-Goal exploration within a single policy. Extensive experiments on the challenging *ClutterScenes* and *InternScenes* benchmarks demonstrate strong generalization of WAM-Nav, particularly on Image-Goal and Point-Goal navigation, where it improves success rates by **15.7%** and **3.3%** relative to NavDP, respectively. Real-world deployment further validates effective zero-shot sim-to-real transfer, achieving an average **85%** task success rate across diverse indoor and outdoor environments. Project Page

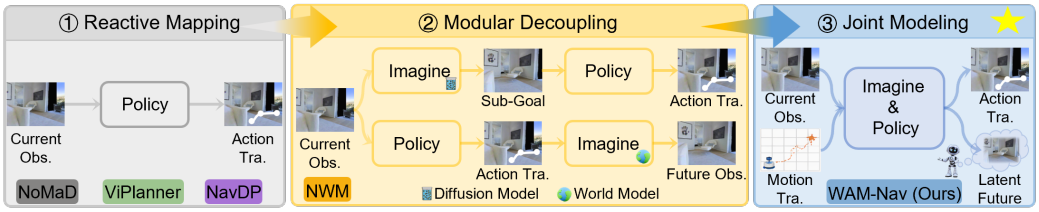
Keywords: Embodied Visual Navigation, World-Action Model, Diffusion Transformer

1 Introduction

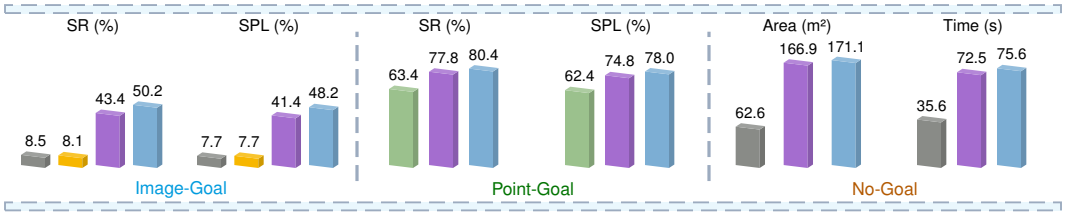
Embodied visual navigation [1, 2, 3, 4] aims to guide agents through complex, unseen physical environments by generating executable trajectories based on visual observations [5]. Over the past few years, the dominant navigation paradigm has gradually transitioned from traditional decoupled mapping-and-planning pipelines [6, 7, 8] to learning-based intelligent methods. As shown in Figure 1, recent end-to-end reactive approaches [9, 10, 11, 12, 13] directly map visual observations to executable actions. Moreover, to endow agents with predictive reasoning capabilities,

* Corresponding authors: Yan Huang (yhuang@nlpr.ia.ac.cn) and Zheng Zhu (zhengzhu@ieee.org).

10th Conference on Robot Learning (CoRL 2026), Austin, Texas, USA.



(a) Comparisons among different embodied navigation frameworks



(b) Comparative results of WAM-Nav and other methods on three navigation tasks

Figure 1: Overview of the WAM-Nav paradigm and performance. (a) Methodological comparison: WAM-Nav jointly models action generation and latent visual foresight within a unified framework. (b) Quantitative performance across three navigation tasks, with method colors matching (a).

recent research has actively explored visual imagination and world-model-based navigation strategies. These modular approaches generally fall into two distinct paradigms. The first paradigm prioritizes generating future visual subgoals, which are subsequently utilized to infer control actions via inverse dynamics models [14, 15]. The second paradigm focuses on trajectory evaluation, where candidate paths are first generated and then scored by a world model that imagines future visual states[16, 17, 18].

Although existing approaches have improved robotic navigation to varying degrees, fundamental limitations remain across current paradigms. Reactive end-to-end mapping methods, which directly map observations to actions, do not explicitly predict future observations and can encounter local optima in cluttered environments. Modular decoupled approaches provide explicit foresight, but their staged action decision-making and future imagination can introduce additional inference cost and sensitivity to prediction errors. While existing World-Action Models have shown promising joint visual-action modeling capabilities in robot manipulation [19, 20, 21], multi-step autoregressive visual generation can be challenging for navigation due to inference cost and prediction uncertainty under large viewpoint changes. Furthermore, existing navigation methods differ in their support for target specifications [9, 10, 11, 12, 16, 14, 18], and adapting the goal interface can require changes to the policy or training procedure. Although NavDP [13] supports multiple goal-oriented navigation tasks, its goal interface differs from our separate visual-semantic and geometric conditioning queries.

Motivated by these advances, we propose WAM-Nav, a Latent World-Action Model for embodied visual navigation that jointly learns action generation and latent visual foresight, enabling more robust and foresighted navigation decisions while retaining practical inference efficiency, as illustrated in Figure 1. Instead of relying on sequential decoupled modules, WAM-Nav utilizes a shared Diffusion Transformer (DiT) for asymmetric joint diffusion to concurrently generate long-horizon actions and short-horizon visual foresight. This joint modeling avoids multi-step autoregressive visual rollouts and couples visual and action tokens throughout the shared DiT. To further guide generation and preserve trajectory consistency, we introduce Dual-Stream Contextual Conditioning (DSCC), which integrates episode-level ego-motion history with sequential visual observations. Combined with a unified goal alignment module that preserves balanced representations across goal types, WAM-Nav naturally supports Image-Goal, Point-Goal, and No-Goal exploration within a single policy without complex architecture switching. We conduct extensive evaluations on cluttered and physically realistic benchmarks, including the *ClutterScenes* and *InternScenes*. Experimental results demonstrate that WAM-Nav achieves strong zero-shot generalization, particularly on Image-Goal and Point-Goal

navigation where it improves success rates by **15.7%** and **3.3%** relative to NavDP, respectively. Furthermore, deployment on physical robot platforms validates its capability for sim-to-real transfer, achieving an average **85%** task success rate across diverse indoor and outdoor environments.

2 Related Work

2.1 Learning-based Embodied Visual Navigation

In contrast to traditional mapping-and-planning methods, the end-to-end learning framework has emerged as a dominant paradigm for embodied visual navigation [5, 22, 23, 24, 25]. Recent foundation models have significantly advanced this field: GNM [9] demonstrated cross-embodiment generalization using a lightweight CNN policy [26], while ViNT [10] extended navigation horizons via Transformer-based sub-goal planning and flexible goal interfaces. Subsequently, NoMaD [11] introduced diffusion policies [27] to model multi-modal action distributions, and NavDP [13] incorporated a critic to score and select generated trajectories. Despite these advances, such reactive policies directly map observations to actions without explicitly modeling their future perceptual consequences [14, 28]. `WAM-Nav` addresses this limitation by integrating predictive visual foresight directly into the policy, conditioning action generation on near-future perceptual information.

2.2 World Models and Visual Imagination for Navigation

Integrating visual imagination facilitates foresight-driven navigation [29, 30], and many systems adopt modular architectures to connect future prediction with motion planning. For instance, Pathdreamer [31] synthesizes future indoor observations, while DeepExplorer [32] hallucinates future image features for exploration and motion planning. NWM [16] uses predicted observations to evaluate candidate trajectories from a navigation policy. Similarly, NavigateDiff [14] employs diffusion models [33] to imagine future visual sub-goals, which are subsequently executed by a separate inverse dynamics policy. These approaches connect visual imagination and action selection through separate stages, making downstream decisions dependent on the predicted visual states.

Joint visual–action modeling has also been explored in driving by Epona [34] and in manipulation by Video Policy [35] and UVA [36]. Building on these advances, `WAM-Nav` jointly models action generation and latent visual foresight for navigation. It pairs a long action horizon with short visual prediction, coupling their tokens throughout a shared DiT conditioned on visual and ego-motion history.

3 Methodology

As illustrated in Figure 2, `WAM-Nav` consists of three key components: (1) *Unified Goal Alignment*, (2) *Dual-Stream Contextual Conditioning*, and (3) *Asymmetric Action-Foresight Generation*. Specifically, to enable balanced performance across diverse navigation tasks, we first project heterogeneous goal specifications into a unified alignment space. We then construct a dual-stream spatiotemporal context by jointly encoding the robot's sequential visual observations and episode-level ego-motion history. Both streams are explicitly modulated by goal-relevant information and aggregated into a compact conditioning context C. Conditioned on C, a shared DiT jointly generates future action trajectories and latent visual foresight through asymmetric denoising.

3.1 Unified Goal Alignment

To enable balanced performance across Image-Goal, Point-Goal, and No-Goal exploration within a single policy, `WAM-Nav` introduces a unified goal alignment mechanism. Our design preserves modality-specific goal information through two projections with different conditioning roles, complementing prior unified goal interfaces [10, 13]. Given a goal input g, `WAM-Nav` encodes it into two complementary goal embeddings: a visual semantic query $g_V \in \mathbb{R}^D$ for visual memory retrieval, and a geometric query $g_G \in \mathbb{R}^D$ for trajectory-level directional guidance.

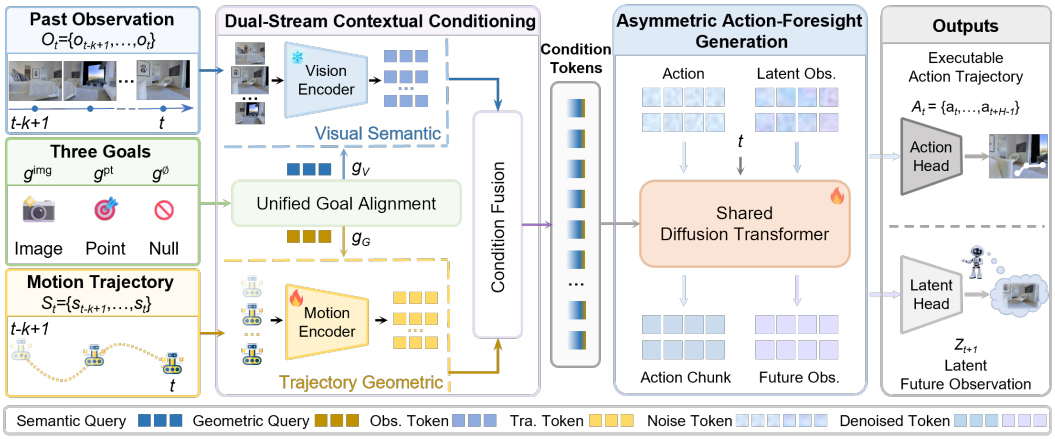


Figure 2: Architecture overview of the WAM-Nav framework. Heterogeneous navigation goals are explicitly routed into visual-semantic (g_V) and trajectory geometric (g_G) queries. These queries contextually modulate historical RGB sequences and relative ego-motion trajectories, synthesizing a compact conditioning context C. Conditioned on C, a shared DiT performs asymmetric joint generation of future control actions and latent visual foresight.

Formally, the goal input g is first transformed into a base embedding e_g via a modality-specific feature extractor $E_\phi(\cdot)$, and then projected into the two functional token spaces:

$$g_V = \psi_V(e_g), \quad g_G = \psi_G(e_g), \quad \text{with} \quad e_g = E_\phi(g) \tag{1}$$

where $E_\phi(\cdot)$ is instantiated according to the goal modality, including a Vision Transformer trained from scratch for image goals, sinusoidal positional encoding for relative coordinates, and a masked zero-state representation for goal-free exploration. The projection operators $\psi_V(\cdot)$ and $\psi_G(\cdot)$ are learnable linear mappings that map the base embedding into the visual semantic and geometric spaces, respectively.

This alignment design preserves modality-specific goal information while providing a unified interface across heterogeneous navigation tasks, supporting a single policy across different goal settings and producing goal-conditioned context for the subsequent Section 3.2, where it serves as conditioning for the joint modeling of action and visual foresight.

3.2 Dual-Stream Contextual Conditioning

While visual history provides spatial cues for obstacle avoidance, it does not explicitly encode the agent's recent motion. To incorporate this temporal context, WAM-Nav introduces Dual-Stream Contextual Conditioning (DSCC) over a historical sliding window from $t-k+1$ to t. We dynamically construct the conditioning context C by fusing a goal-modulated visual memory stream (capturing local spatial geometry) with a trajectory-aware kinematic history stream (summarizing recent motion). Fusing these streams provides the shared DiT with a spatiotemporal conditioning context that links local visual geometry with recent trajectory dynamics for action generation.

Goal-Modulated Visual Memory. Historical RGB observations $\mathcal{O}_t$ are processed by DINO-v2 [37] into a memory tensor $V \in \mathbb{R}^{B\times(k\cdot P)\times D}$, where B represents the batch size, k is the historical window length, and P is the patch count per frame. To selectively reinforce spatial tokens relevant to the target, the visual query $g_V \in \mathbb{R}^{B\times D}$ computes independent relevance scores for each patch token via a scaled dot-product:

$$\alpha_{i,p} = \sigma\left(\frac{\langle g_V, V_{i,p}\rangle}{\sqrt{D}}\right), \qquad i = 1,\ldots,k, \quad p = 1,\ldots,P \tag{2}$$

where σ is the sigmoid function and $V_{i,p}$ denotes patch p of history frame i (batch index omitted). The visual memory is then residually updated:

$$\tilde{V} = V + \alpha \odot V \tag{3}$$

Trajectory-Aware Motion History. To incorporate recent ego-motion and facilitate transfer across scenes, the motion trajectory stream processes the executed pose sequence $\mathcal{S}_t$. Absolute poses are converted into coordinate-invariant relative displacements and heading changes in the current egocentric frame, yielding a relative kinematics sequence $\tilde{\mathcal{S}}_t = \{(\Delta x_i, \Delta y_i, \Delta\theta_i)\}_{i=t-k+1}^{t}$ (detailed in Appendix C.1).

This relative motion sequence $\tilde{\mathcal{S}}_t$ is embedded and processed by a causal Transformer encoder $H \in \mathbb{R}^{B\times k\times D}$ to capture temporal kinematic profiles. To align motion history with current target directionality, we query H with the geometric goal embedding $g_G \in \mathbb{R}^{B\times D}$ via cross-attention to extract a consolidated kinematic vector o_{kin}:

$$o_{\text{kin}} = \text{CrossAttn}(g_G, H, H) \in \mathbb{R}^{B\times D} \tag{4}$$

The vector o_{kin} summarizes the agent's historical motion relative to its target directionality.

Condition Fusion via Cross-Attention. To let motion context guide visual retrieval, the visual and trajectory streams are integrated via a multi-layer Transformer Decoder to formulate the unified conditioning context $C \in \mathbb{R}^{B\times N_c\times D}$. Specifically, the extracted kinematic token o_{kin} is projected to bias a set of learnable query embeddings $Q_c \in \mathbb{R}^{N_c\times D}$:

$$C = \text{TransformerDecoder}\left(Q_c + \phi(o_{\text{kin}}), \tilde{V}, \tilde{V}\right) \tag{5}$$

where $\phi(\cdot)$ denotes a linear projection. Architecturally, this fusion mechanism utilizes the agent's motion context (o_{kin}) to actively query the goal-modulated visual spatial context ($\tilde{V}$). The resulting conditioning context C integrates the retrieved visual features with goal-directed motion information to guide joint action–visual generation.

3.3 Asymmetric Action-Foresight Generation

Conditioned on the unified conditioning context C, `WAM-Nav` jointly models the future action trajectory $\mathbf{A}_t = \{a_t, \ldots, a_{t+H_{\text{act}}-1}\}$ over a long planning horizon H_{act} and its corresponding latent visual foresight $\mathbf{Z}_{t+1:t+H_{\text{vis}}} = \{z_{t+1}, \ldots, z_{t+H_{\text{vis}}}\}$ over a short prediction horizon H_{vis} ($H_{\text{vis}} \le H_{\text{act}}$) within a shared representation space. The future visual state $z_i = \mathcal{E}(o_i)$ is compressed via a pre-trained Stable Diffusion VAE [38] into a compact grid of N latent patches.

To optimize this generative process, we employ the flow-matching [39] training strategy to learn the joint conditional distribution $p_\theta(\mathbf{A}, \mathbf{Z} \mid C)$ via asymmetric joint diffusion. This strategy formulates the continuous probability paths as straight lines that map standard Gaussian priors $(\mathbf{A}_0, \mathbf{Z}_0) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to the empirical data manifold. At any flow time $\tau \in [0, 1]$, the interpolated states are defined as:

$$\mathbf{A}_\tau = (1-\tau)\mathbf{A}_0 + \tau\mathbf{A}_1, \quad \mathbf{Z}_\tau = (1-\tau)\mathbf{Z}_0 + \tau\mathbf{Z}_1 \tag{6}$$

where the corresponding target velocity fields are given by $u_A = \mathbf{A}_1 - \mathbf{A}_0$ and $u_Z = \mathbf{Z}_1 - \mathbf{Z}_0$.

The architectural backbone for this asymmetric generative process is a shared DiT. The noised sequences $\mathbf{A}_\tau$ and $\mathbf{Z}_\tau$ are tokenized, concatenated, and processed through a multi-layer DiT. Crucially, within each DiT block, these heterogeneous tokens undergo a shared self-attention, enabling interaction between action and latent visual representations at every layer. Both streams cross-attend to the conditioning context C, with the timestep τ modulated via adaptive LayerNorm (adaLN) layers to regress the joint velocity fields:

$$\hat{u}_A, \hat{u}_Z = f_\theta(\mathbf{A}_\tau, \mathbf{Z}_\tau, \tau, C). \tag{7}$$

This shared parameterization allows latent visual foresight to guide action generation through token interaction and joint velocity matching. Unlike manipulation-oriented WAMs [19, 20], where future visual changes are often local and object-centric, navigation involves large egocentric viewpoint changes; long visual rollouts can be costly and difficult to predict reliably. `WAM-Nav` therefore adopts an asymmetric design: long-horizon actions provide a continuous trajectory plan, while short-horizon latent foresight supplies near-future perceptual context without overextending visual prediction.

3.4 Model Training and Inference

We train WAM-Nav end-to-end by minimizing a joint objective consisting of flow-matching velocity regression and multi-modal goal alignment:

$$\mathcal{L}_{\text{total}} = \mathbb{E}_{\tau,\mathbf{A}_0,\mathbf{Z}_0}\left[\|\hat{u}_A - u_A\|_2^2 + \lambda_{\text{img}}\|\hat{u}_Z - u_Z\|_2^2\right] + \lambda_{\text{align}}\mathcal{L}_{\text{align}} \tag{8}$$

where $\tau \sim \mathcal{U}(0,1)$, and $\lambda_{\text{img}}, \lambda_{\text{align}}$ are scalar weighting coefficients. The term $\mathcal{L}_{\text{align}}$ represents a symmetric contrastive InfoNCE loss [40]. Image and point goals corresponding to the same endpoint form positive pairs; other cross-modal pairs in the batch form negatives. No-Goal samples are excluded. By aligning cross-space projections of equivalent objectives, this loss encourages consistent directional and visual cues across goal modalities.

During online deployment, WAM-Nav follows a receding-horizon control loop. Consistent with NavDP [13], at each step, conditioned on the current conditioning context C, the policy samples 16 candidate trajectories. Following prior generative navigation policies [11], the first sampled trajectory is selected for execution.

4 Experimental Results

We evaluate WAM-Nav through five questions: **Q1** multi-task generalization, **Q2** qualitative trajectory and foresight comparisons, **Q3** computational efficiency, **Q4** component contributions, and **Q5** zero-shot transfer to physical platforms.

4.1 Simulation Experiments

Training Datasets and Setup. Consistent with prior works [12, 13], WAM-Nav is trained on the large-scale visual navigation dataset VLN-N1 [41, 42]. VLN-N1 is built upon six categories of 3D scene assets, including Replica [43], Matterport3D [44], Gibson [45], 3D-FRONT [46], HSSD [47], and HM3D [48]. Leveraging an extensive domain randomization pipeline, the dataset provides over 400 hours of collision-free and smooth first-person navigation trajectories collected in simulation, comprising more than 200K trajectories in total. WAM-Nav is trained with a learning rate of 1.5×10^{-4} and the total training cost is approximately 8×120 GPU hours. Additional implementation details are provided in Appendix D.

Zero-Shot Evaluation Platform. We evaluate WAM-Nav and all baselines in a zero-shot setting on a simulation benchmark built upon IsaacSim, using the wheeled robot ClearPath Dingo as the navigator. To assess zero-shot transfer to this new benchmark, all methods, including WAM-Nav, are evaluated with their pretrained weights, without training or finetuning on the evaluation benchmark. We additionally report NoMaD and NWM adapted to the full VLN-N1 training set, separately from the original-checkpoint comparison. The benchmark includes two categories of environments: *ClutterScenes* and *InternScenes* [13, 41, 49]. The *ClutterScenes* contain 10 *easy* scenes and 10 *hard* scenes. Both are constructed from randomly generated layouts with cluttered obstacles, while the *hard* scenes feature denser obstacle distributions. The *InternScenes* include 20 *home* scenes and 20 *commercial* scenes. *Home* scenes are characterized by narrow passages and cluttered layouts, whereas *commercial* scenes cover representative indoor categories such as *supermarkets* and *restaurants*. For each scene, we randomly sample 100 navigation episodes, yielding a benchmark of 6,000 evaluation episodes.

Metrics. For Image-Goal and Point-Goal navigation, we evaluate **Success Rate (SR)** and **Success weighted by Path Length (SPL)** to measure task completion and navigation efficiency. For No-Goal exploration, we report episode **Time** and explored **Area**; exploration episodes end at 120 seconds or earlier upon collision or local trapping. Higher values indicate better performance for all metrics. In goal-conditioned tasks, collisions, timeouts, trapping and missed goals count as failures.

Table 1: Original-checkpoint evaluation across three navigation tasks. All overall averages are unweighted macro averages of the four scene categories; M_1 and M_2 are defined within each task block.

Method	ClutterScenes				InternScenes				Average	
	Easy		Hard		Home		Commercial			
	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2
Task 1: Image-Goal Navigation (M_1: SR ↑, M_2: SPL ↑)										
GNM	26.9	25.9	20.0	19.2	7.7	7.3	10.5	10.3	16.3	15.7
ViNT	15.0	13.9	16.5	15.2	6.2	6.0	12.9	12.2	12.7	11.8
NoMaD	8.3	6.9	7.3	6.9	6.7	6.2	11.7	10.7	8.5	7.7
NWM	12.7	12.0	8.7	8.4	4.0	3.9	6.9	6.6	8.1	7.7
NavDP	49.7	48.4	47.5	46.3	29.8	27.7	46.6	43.3	43.4	41.4
Ours	**66.1**	**64.5**	**55.6**	**53.4**	**30.7**	**29.4**	**48.3**	**45.6**	**50.2**	**48.2**
Task 2: Point-Goal Navigation (M_1: SR ↑, M_2: SPL ↑)										
DD-PPO	0.0	0.0	0.0	0.0	0.4	0.4	5.3	5.2	1.4	1.4
iPlanner	89.4	88.4	80.3	78.8	43.0	40.6	54.6	52.8	66.8	65.2
ViPlanner	80.2	80.1	64.7	64.5	45.0	43.2	63.7	61.9	63.4	62.4
NavDP	92.3	90.5	87.4	84.9	60.0	55.6	71.4	68.2	77.8	74.8
Ours	**93.8**	**91.6**	**88.9**	**86.4**	**61.4**	**58.6**	**77.4**	**75.3**	**80.4**	**78.0**
Task 3: No-Goal Exploration (M_1: Area (m^2) ↑, M_2: Time (s) ↑)										
GNM	42.8	17.1	29.6	12.5	19.8	30.8	19.2	28.1	27.9	22.1
ViNT	55.3	20.3	46.6	18.9	20.8	31.5	22.4	27.2	36.3	24.5
NoMaD	119.4	48.1	85.7	36.6	21.7	27.8	23.4	29.7	62.6	35.6
NavDP	315.6	112.7	274.1	106.2	34.1	35.0	43.9	36.2	166.9	72.5
Ours	**320.4**	**115.2**	**281.7**	**108.6**	**36.8**	**38.3**	**45.5**	**40.2**	**171.1**	**75.6**

Compared Methods. To comprehensively evaluate the effectiveness of `WAM-Nav`, we benchmark our model against a diverse set of strong baselines tailored to different navigation tasks. Specifically, for **Image-Goal** navigation, we compare `WAM-Nav` against GNM [9], ViNT [10], NoMaD [11], NWM [16], and NavDP [13]. For **Point-Goal** navigation, the baselines include DD-PPO [24], iPlanner [50], ViPlanner [51], and NavDP [13]. Finally, for **No-Goal** exploration, we compare against GNM [9], ViNT [10], NoMaD [11], and NavDP [13] (detailed configurations are provided in Appendix E).

Result Analysis. For **Q1**, Table 1 summarizes the zero-shot evaluation results across Image-Goal, Point-Goal, and No-Goal exploration. `WAM-Nav` achieves the best average performance across the three task settings, reaching 50.2% SR / 48.2% SPL on Image-Goal, 80.4% SR / 78.0% SPL on Point-Goal, and 171.1 m^2 explored area in No-Goal exploration. The consistent gains across both *ClutterScenes* and *InternScenes* support zero-shot generalization across diverse obstacle layouts and semantic scene complexity. The same policy handles all three goal settings, combining latent visual foresight with motion-aware conditioning.

For **Q2**, Figure 3 qualitatively compares NavDP and `WAM-Nav` on the Image-Goal task. The top-down trajectories and egocentric projections show the planned paths through cluttered areas, while decoded foresight is presented alongside ground-truth observations. Together, these examples illustrate how `WAM-Nav` pairs long-horizon action plans with near-future visual predictions.

For **Q3**, Table 2 shows that `WAM-Nav` remains efficient for online navigation. Despite its frozen visual priors increasing the total parameter count, its trainable parameters are comparable to NavDP, with lower per-decision computation (0.7 vs. 1.3 TFLOPs). Its 0.26 s inference latency introduces modest overhead over NavDP while avoiding NWM's costly multi-candidate visual rollouts (1.43 s, 8.3 TFLOPs).

For **Q4**, Table 5 (Appendix F.2) evaluates trajectory conditioning and latent foresight. Latent prediction alone improves SR from 42.1% to 45.7%. Motion trajectories alone improve performance in

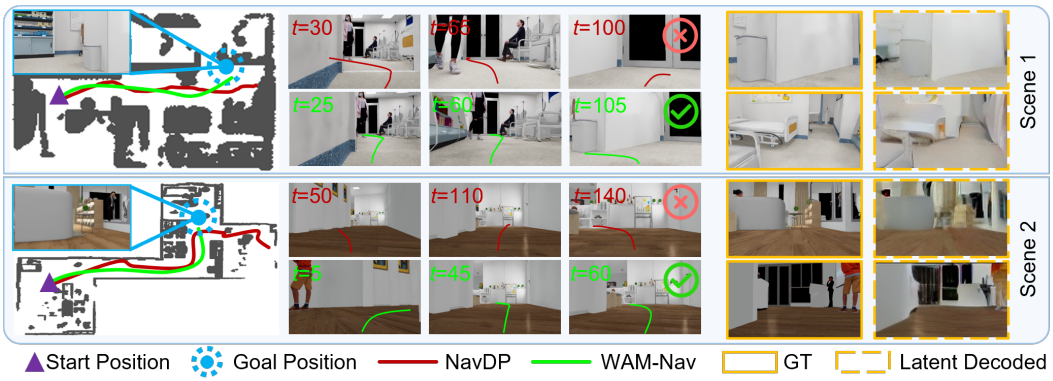


Figure 3: **Qualitative comparison on image-goal navigation.** As shown in the 2D top-down trajectories and egocentric path projections, we show example trajectory differences between NavDP and `WAM-Nav`. Decoded visual foresight is shown alongside ground-truth (GT) observations for qualitative comparison.

Table 2: Computational efficiency comparison across navigation methods.

Method	Inference Latency (s) ↓	Total Params	Trainable Params	TFLOPs ↓
NWM	1.43	154.4M	68.3M	8.3
NavDP	**0.16**	136.4M	136.4M	1.3
Ours	0.26	234.9M	129.2M	**0.7**

ClutterScenes but degrade in the more semantically complex *InternScenes*. Combining both yields the best result (50.2% SR / 48.2% SPL), supporting their complementary roles. Controlled fusion and coupling variants are reported in Appendix F.4.

Table 3: Image-Goal controls on the same held-out benchmark. SR/SPL are macro averages (%). Adapted baselines use the full VLN-N1 training set. Ablation deltas are measured against the same full WAM-Nav configuration.

Configuration	SR	SPL	ΔSR
NoMaD (VLN-N1 adapted)	39.3	36.8	–
NWM (VLN-N1 adapted)	42.7	41.5	–
WAM-Nav (full)	**50.2**	**48.2**	–
w/o $\mathcal{L}_{\text{align}}$	38.6	37.3	−11.6
Direct Concat fusion	35.4	33.8	−14.8
Simple Transformer fusion	41.9	40.2	−8.3
Decoupled DiT	45.9	44.6	−4.3
Partially shared DiT	47.5	45.4	−2.7

Table 3 compares adapted baselines and component designs. WAM-Nav exceeds VLN-N1-adapted NWM by 7.5/6.7 SR/SPL points. Removing goal alignment reduces SR by 11.6 points, while removing latent prediction reduces SR by 6.3 points (Table 5), highlighting both components' contributions.

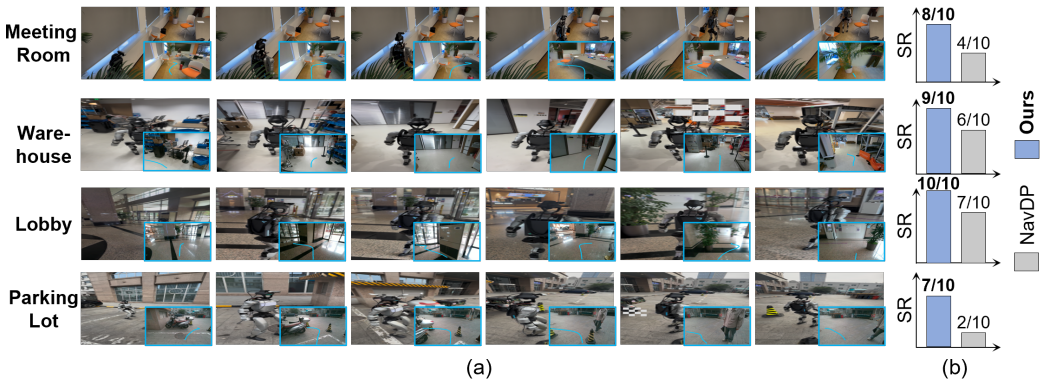


Figure 4: **Zero-shot deployment of `WAM-Nav` in real-world environments.** (a) Visualization of `WAM-Nav` navigating across four representative indoor and outdoor scenes, with the first-person trajectory planning visualization shown in the bottom-right corner of each image. (b) Quantitative comparison of `WAM-Nav` and NavDP in each real-world scenario.

4.2 Real-World Deployment

For **Q5**, we deploy `WAM-Nav` in a zero-shot manner on a real-world Unitree G1 humanoid robot equipped with an Intel RealSense D455 camera. Evaluation is conducted across four indoor and outdoor environments: a *meeting room*, *warehouse*, *lobby*, and *parking lot*, with 10 trials per scene. The meeting room and lobby use Image-Goal, the warehouse uses No-Goal, and the parking lot uses Point-Goal; both methods receive identical goals, starts, sensing and control. Real-world No-Goal success requires no obstacle collision throughout a 120-second trial. As shown in Figure 4(a), `WAM-Nav` consistently predicts feasible trajectories under diverse layouts and lighting conditions, achieving an average 85% success rate. The higher success rate than NavDP across these scenes supports effective zero-shot sim-to-real transfer. Additional protocol details are in Appendix G.

5 Conclusion

This paper presents `WAM-Nav`, a Latent World-Action Model for embodied visual navigation that unifies action generation and latent visual foresight within a shared generative framework. By using asymmetric action-foresight generation, `WAM-Nav` pairs long-horizon action trajectories with short-horizon visual foresight, avoiding multi-step autoregressive visual rollouts at inference. Its dual-stream conditioning further integrates sequential visual observations with episode-level ego-motion history, conditioning action generation on both perceptual and motion context. Together with unified goal alignment, the same policy supports Image-Goal, Point-Goal, and No-Goal exploration. Extensive zero-shot experiments on *ClutterScenes* and *InternScenes*, controlled ablations, and real-world deployment on a humanoid robot demonstrate strong task performance, effective sim-to-real transfer, and practical potential for embodied navigation.

6 Limitations

During real-world deployment of `WAM-Nav`, we identify two primary failure modes. First, the limited camera height and field of view restrict perception of near-field obstacles, which may lead to inaccurate trajectory prediction or delayed avoidance. Second, the current policy does not explicitly model robot embodiment, causing mismatches between camera-level obstacle avoidance and full-body traversability. These limitations suggest two promising future directions: adaptive viewpoint control for dynamic perception adjustment, and embodiment-aware training with diverse robot morphologies to improve navigation robustness.

Acknowledgments

This work was jointly supported by National Natural Science Foundation of China (62322607, 62236010 and 62276261), Beijing Natural Science Foundation (L252033).

References

[1] X. Wei, C. Gu, and L. Zhang. NavOL: Navigation policy with online imitation learning. *arXiv preprint arXiv:2605.11762*, 2026.

[2] N. Yang, F. Lu, X. Li, G. Tian, Z. Li, and T. Fu. Transformer-driven semantic-spatial adaptive fusion representation for object-goal navigation. *IEEE Transactions on Automation Science and Engineering*, 22:19135–19150, 2025.

[3] N. Yang, F. Lu, G. Tian, and J. Liu. Long-term active object detection for service robots: Using generative adversarial imitation learning with contextualized memory graph. *IEEE Transactions on Industrial Electronics*, 72(5):5082–5092, 2025.

[4] J. Zhang, A. Li, Y. Qi, M. Li, J. Liu, S. Wang, H. Liu, G. Zhou, Y. Wu, X. Li, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025.

[5] F. Zhu, Y. Zhu, V. C. Lee, X. Liang, and X. Chang. Deep learning for embodied vision navigation: A survey. *arXiv preprint arXiv:2108.04097*, 2021.

[6] C. Campos, R. Elvira, J. J. Gómez, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[7] M. Labbé and F. Michaud. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019.

[8] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov. Learning to explore using active neural SLAM. In *International Conference on Learning Representations (ICLR)*, 2020.

[9] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

[10] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A foundation model for visual navigation. In *Conference on Robot Learning (CoRL)*, pages 711–733, 2023.

[11] A. Sridhar, D. Shah, C. Glossop, and S. Levine. NoMaD: Goal masking diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[12] J. Peng, W. Cai, Y. Yang, T. Wang, Y. Shen, and J. Pang. LoGoPlanner: Localization grounded navigation policy with metric-aware visual geometry. *arXiv preprint arXiv:2512.19629*, 2025.

[13] W. Cai, J. Peng, Y. Yang, Y. Zhang, M. Wei, H. Wang, Y. Chen, T. Wang, and J. Pang. NavDP: Learning sim-to-real navigation diffusion policy with privileged information guidance. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026.

[14] Y. Qin, A. Sun, Y. Hong, B. Wang, and R. Zhang. NavigateDiff: Visual predictors are zero-shot navigation assistants. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[15] F. Ni, J. Hao, S. Wu, L. Kou, J. Liu, Y. Zheng, B. Wang, and Y. Zhuang. Generate subgoal images before act: Unlocking the chain-of-thought reasoning in diffusion model for robot manipulation with multimodal prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13991–14000, 2024.

[16] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[17] M. Zhang, W. Shen, F. Zhang, H. Qin, Z. Pei, and Z. Meng. RAE-NWM: Navigation world model in dense visual representation space. *arXiv preprint arXiv:2603.09241*, 2026.

[18] Y. Dong, F. Wu, G. Chen, L. Kong, Q. Hu, Y. Zhou, X. Zhu, J. Sun, J.-Y. He, Q. Dai, A. G. Hauptmann, and Z.-Q. Cheng. Towards unified world models for visual navigation via memory-augmented planning and foresight. *arXiv preprint arXiv:2510.08713*, 2025.

[19] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[20] H. Bi, H. Tan, S. Xie, Z. Wang, S. Huang, H. Liu, R. Zhao, Y. Feng, C. Xiang, Y. Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

[21] L. Li, Q. Zhang, Y. Luo, S. Yang, R. Wang, F. Han, M. Yu, Z. Gao, N. Xue, X. Zhu, Y. Shen, and Y. Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

[22] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3357–3364. IEEE, 2017.

[23] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9339–9347. IEEE, 2019.

[24] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations (ICLR)*, 2020.

[25] Y. Zeng, H. Ren, S. Wang, J. Huang, and H. Cheng. NaviDiffusor: Cost-guided diffusion model for visual navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11994–12001, 2025.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, pages 1097–1105, 2012.

[27] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. C. M. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023.

[28] W. Shen, Z. Meng, J. Ma, M. Zhou, and D. Xiang. An efficient and multi-modal navigation system with one-step world model. *arXiv preprint arXiv:2601.12277*, 2026.

[29] H. Zhang, S. Liang, L. Chen, Y. Li, Y. Xu, Y. Zhong, F. Zhang, and H. Li. Sparse video generation propels real-world beyond-the-view vision-language navigation. *arXiv preprint arXiv:2602.05827*, 2026.

[30] D. Nie, X. Guo, Y. Duan, R. Zhang, and L. Chen. WMNav: Integrating vision-language models into world models for object goal navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[31] J. Y. Koh, H. Lee, Y. Yang, J. Baldridge, and P. Anderson. Pathdreamer: A world model for indoor navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14738–14748, 2021.

[32] Y. He, I. Fang, Y. Li, R. B. Shah, and C. Feng. Metric-free exploration for topological mapping by task and motion imitation in feature space. In *Proceedings of Robotics: Science and Systems*, 2023.

[33] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 6840–6851, 2020.

[34] K. Zhang, Z. Tang, X. Hu, X. Pan, X. Guo, Y. Liu, J. Huang, L. Yuan, Q. Zhang, X.-X. Long, X. Cao, and W. Yin. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27220–27230, 2025.

[35] J. Liang, P. Tokmakov, R. Liu, S. Sudhakar, P. Shah, R. Ambrus, and C. Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.

[36] S. Li, Y. Gao, D. Sadigh, and S. Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

[37] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[38] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[39] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

[40] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[41] Intern Robotics. InternVLA-N1: An open dual-system vision-language navigation foundation model with learned latent plans. Technical report, Shanghai AI Laboratory, 2025. URL https://internrobotics.github.io/internvla-n1.github.io/static/pdfs/InternVLA_N1.pdf.

[42] InternRobotics Contributors. InternData-N1 dataset, 2025. URL https://huggingface.co/datasets/InternRobotics/InternData-N1.

[43] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, et al. The Replica dataset: A digital Replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[44] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017.

[45] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9068–9079, 2018.

[46] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, and H. Zhang. 3D-FRONT: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10933–10942, 2021.

[47] M. Khanna, Y. Mao, H. Jiang, S. Haresh, B. Shacklett, D. Batra, A. Clegg, E. Undersander, A. X. Chang, and M. Savva. Habitat synthetic scenes dataset (HSSD-200): An analysis of 3d scene scale and realism tradeoffs for ObjectGoal navigation. *arXiv preprint arXiv:2306.11290*, 2023.

[48] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, et al. Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied ai. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[49] H. Wang, J. Chen, W. Huang, Q. Ben, T. Wang, B. Mi, T. Huang, S. Zhao, Y. Chen, S. Yang, et al. GRUtopia: Dream general robots in a city at scale. *arXiv preprint arXiv:2407.10943*, 2024.

[50] F. Yang, C. Wang, C. Cadena, and M. Hutter. iPlanner: Imperative path planning. In *Proceedings of Robotics: Science and Systems (RSS)*, Daegu, Republic of Korea, July 2023.

[51] P. Roth, J. Nubert, F. Yang, M. Mittal, and M. Hutter. ViPlanner: Visual semantic imperative learning for local navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5243–5249. IEEE, 2024.

[52] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

A Overview

The appendix is organized as follows:

- Section B provides the formal problem definition.
- Section C provides additional details of the proposed method.
- Section D presents the model configurations and training details of `WAM-Nav`.
- Section E introduces the details of the baseline methods.
- Section F includes additional experimental results and ablation studies.
- Section G describes the hardware setup and deployment details for real-world experiments.
- Section H provides the visualization of a failure case.

B Problem Definition

We consider mapless visual navigation under three goal modalities: Image-Goal (g^{img}), Point-Goal (g^{pt}), and No-Goal exploration ($g^{\varnothing}$). At each time step t in a partially observable environment, the agent receives a visual history $\mathcal{O}_t = \{o_{t-k+1}, \ldots, o_t\}$, an action-accumulated ego-motion history $\mathcal{S}_t = \{s_{t-k+1}, \ldots, s_t\}$, and a goal condition $g \in \{g^{\mathrm{img}}, g^{\mathrm{pt}}, g^{\varnothing}\}$.

To move beyond myopic reactive control, `WAM-Nav` formulates navigation as conditional asymmetric joint generation. Conditioned on the compact conditioning context C constructed from $(\mathcal{O}_t, \mathcal{S}_t, g)$, the policy predicts a long-horizon action trajectory $\mathbf{A}_t = \{a_t, \ldots, a_{t+H_{\mathrm{act}}-1}\}$ and a short-horizon latent visual foresight sequence $\mathbf{Z}_{t+1:t+H_{\mathrm{vis}}} = \{z_{t+1}, \ldots, z_{t+H_{\mathrm{vis}}}\}$, where $H_{\mathrm{vis}} \leq H_{\mathrm{act}}$. The learning objective is to model:

$$p_\theta(\mathbf{A}_t, \mathbf{Z}_{t+1:t+H_{\mathrm{vis}}} \mid C). \tag{9}$$

This formulation jointly models a long action chunk and a short future visual sequence without requiring an autoregressive visual rollout.

Algorithm 1 Online Ego-Motion History and Egocentric Conversion

Require: window length k; executed action a_{t-1}; pose accumulator s_{t-1}; pose buffer $\mathcal{B}$
Ensure: egocentric relative-motion sequence $\tilde{\mathcal{S}}_t = \{(\Delta x_i, \Delta y_i, \Delta\theta_i)\}_{i=t-k+1}^{t}$

$s_t \leftarrow s_{t-1} + a_{t-1}$
Append s_t to $\mathcal{B}$; if $|\mathcal{B}| > k$, remove the oldest pose
if $|\mathcal{B}| < k$ **then**
  $\mathcal{S}_t \leftarrow$ zero-pad the front of $\mathcal{B}$ to length k
else
  $\mathcal{S}_t \leftarrow \mathcal{B}$
end if
$s_t = (x_t, y_t, \theta_t) \leftarrow$ the last pose in $\mathcal{S}_t$
$\tilde{\mathcal{S}}_t \leftarrow \emptyset$
for each $s_i = (x_i, y_i, \theta_i)$ in $\mathcal{S}_t$ **do**
  $\delta x \leftarrow x_i - x_t, \quad \delta y \leftarrow y_i - y_t$
  $\Delta x \leftarrow \cos\theta_t \cdot \delta x + \sin\theta_t \cdot \delta y$
  $\Delta y \leftarrow -\sin\theta_t \cdot \delta x + \cos\theta_t \cdot \delta y$
  $\Delta\theta \leftarrow \mathrm{atan2}(\sin(\theta_i - \theta_t), \cos(\theta_i - \theta_t))$
  Append $(\Delta x, \Delta y, \Delta\theta)$ to $\tilde{\mathcal{S}}_t$
end for

C Method Details

C.1 Trajectory-Aware Motion History

While goal-modulated visual memory provides spatial context over $\mathcal{O}_t$, it does not explicitly encode kinematic momentum; the motion trajectory stream therefore maintains an action-accumulated ego-motion history $\mathcal{S}_t = \{s_{t-k+1}, \ldots, s_t\}$ (Sec. B). At deployment, each executed action a_{t-1} updates the current planar pose s_t, which is appended to a length-k sliding buffer and zero-padded at the front when the episode is shorter than k steps. The buffer is then re-expressed in the current egocentric frame—re-centering on s_t, rotating translations by θ_t, and wrapping heading differences—so that the stream receives a complete relative-motion sequence $\tilde{\mathcal{S}}_t = \{(\Delta x_i, \Delta y_i, \Delta\theta_i)\}_{i=t-k+1}^{t}$ rather than absolute scene coordinates. Algorithm 1 details this online procedure.

C.2 Architecture of the Shared DiT Block

As introduced in the **Asymmetric Action-Foresight Generation** subsection, the shared DiT predicts the joint flow-matching velocity fields $\hat{u}_A$ and $\hat{u}_Z$ from the interpolated action and latent visual states $(\mathbf{A}_\tau, \mathbf{Z}_\tau)$, conditioned on the conditioning context C and flow timestep τ. Figure 5 illustrates a single DiT block, stacked L times.

At each block, $\mathbf{A}_\tau$ and $\mathbf{Z}_\tau$ are tokenized into parallel action and image streams. The timestep embedding τ is injected into each stream via separate adaLN-Zero branches. The modulated tokens then pass through a shared self-attention layer, enabling early interaction between action and latent visual tokens. After the residual update, both streams perform cross-attention over the conditioning context C using shared cross-attention and FFN weights with stream-specific modulation. The updated features are propagated through stacked blocks, and finally projected to $\hat{u}_A$ and $\hat{u}_Z$. This design enables consistent action–foresight generation through shared parameters and cross-modal coupling.

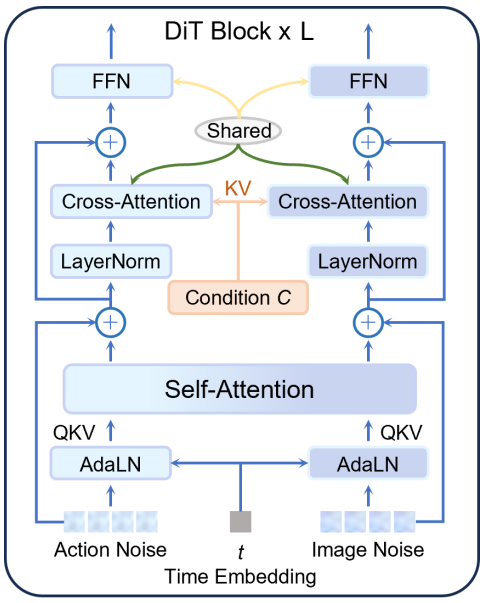


Figure 5: Architecture of the shared DiT block for asymmetric action-foresight generation. Noised action tokens and latent visual foresight tokens are modulated through separate adaLN branches, coupled via shared self-attention, and grounded on the conditioning context C through shared cross-attention and FFN layers.

C.3 Goal Alignment Training Protocol

Image and point goals associated with the same navigation endpoint are treated as positive cross-modal pairs. All other cross-modal pairs in the minibatch are negatives. The InfoNCE loss is computed symmetrically in the image-to-point and point-to-image directions, with a learnable temperature initialized to 0.07. No-Goal zero representations are excluded from this loss. We set $\lambda_{\text{align}} = 0.1$ to balance the contrastive and velocity-matching loss scales. Removing only $\mathcal{L}_{\text{align}}$ from the full configuration reduces Image-Goal SR/SPL from 50.2/48.2 to 38.6/37.3 (Table 3).

D Implementation Details of WAM-Nav

Table 4: Key hyperparameters of `WAM-Nav`.

Parameter	Value	Parameter	Value
Hidden Dimension	384	Condition Length	64
Memory Window (k)	8	Action Horizon (H_{act})	24
DiT Layers	16	DiT Attention Heads	8
Fusion Layers	4	Fusion Attention Heads	8
Causal Layers	2	Causal Attention Heads	4
Latent Grid Size	16×16	Latent Channels	4
Dropout	0.1	–	–
Batch Size	256	Learning Rate	1.5×10^{-4}
Weight Decay	1×10^{-4}	Warmup Ratio	0.05
λ_{img}	0.25	λ_{align}	0.1
Inference Steps	10	Visual Horizon (H_{vis})	1
Visual Backbone	DINOv2 ViT-S/14	–	–

We train `WAM-Nav` using a batch size of 256 and optimize the model with the AdamW optimizer. The peak learning rate is set to 1.5×10^{-4} with a weight decay of 1×10^{-4}, following a cosine annealing schedule with a 5% linear warmup. During training, the visual backbone (DINOv2 ViT-S/14) and the pre-trained Stable Diffusion VAE are kept frozen to preserve their general-purpose representations, while the goal image encoder, fusion decoder, causal motion encoder, and the shared DiT are trained from scratch. The loss weighting coefficients in Eq. 8 are empirically set to $\lambda_{\text{img}} = 0.25$ and $\lambda_{\text{align}} = 0.1$. At deployment, the flow-matching ODE solver runs for $N_{\text{steps}} = 10$ Euler integration steps, which balances high-quality trajectory generation with real-time responsiveness. The detailed hyperparameter configurations are summarized in Table 4.

E Baseline Details

To comprehensively evaluate `WAM-Nav`, we compare against representative baselines spanning reinforcement learning, reactive planning, foundation models, and generative world models.

- **DD-PPO** [24]: A large-scale distributed reinforcement learning framework for Point-Goal navigation, trained over billions of frames in Habitat and widely adopted as a strong mapless navigation baseline.
- **iPlanner** [50]: An imperative-learning planner that predicts local paths from depth observations, trained with a differentiable planning cost.
- **ViPlanner** [51]: An imperative-learning local planner that uses semantic and depth images to predict trajectories, trained with semantic traversability costs.
- **GNM** [9]: A general navigation foundation model trained on large-scale cross-embodiment datasets for zero-shot transfer and low-level action prediction.
- **ViNT** [10]: A Transformer-based navigation policy that extends GNM by jointly modeling heterogeneous visual navigation data with improved sequence representation.

- **NoMaD** [11]: A diffusion-based action policy that introduces goal masking to unify goal-conditioned navigation and goal-free exploration.
- **NWM** [16]: A generative navigation world model based on conditional DiT that predicts future egocentric observations for foresight-driven planning.
- **NavDP** [13]: A diffusion-based navigation policy that leverages privileged simulation priors for zero-shot sim-to-real transfer.

F Additional Experimental Results

F.1 Main Simulation Results

The complete original-checkpoint results are in Table 1 in the main paper. Categories are weighted equally in all reported benchmark macro averages. The adapted baseline comparison is reported separately in Table 3.

F.2 Main Component Ablation

Table 5: Ablation study of different components of `WAM-Nav` on Image-Goal navigation.

Method		ClutterScenes (Avg.)		InternScenes (Avg.)		Overall Average	
Motion Traj.	Latent Pred.	SR	SPL	SR	SPL	SR	SPL
×	×	51.0	49.9	33.2	31.7	42.1	40.8
×	✓	57.4	55.5	34.0	32.2	45.7	43.9
✓	×	57.9	57.1	30.0	28.6	44.0	42.9
✓	✓	**60.9**	**59.0**	**39.5**	**37.5**	**50.2**	**48.2**

F.3 Additional Ablation Studies

To systematically investigate the contribution of each key component in `WAM-Nav`, we conduct ablation studies on unified goal alignment, dual-stream contextual conditioning, and asymmetric action-foresight generation.

The goal-projection variants in Table 7 are each unified policies trained once on identical data and budget, differing only in goal projection and each handling all three navigation tasks with one checkpoint. The table compares variants that retain only the visual query g_V, only the geometric query g_G, or both streams. Using only g_V favors Image-Goal navigation, while using only g_G performs best on Point-Goal navigation; however, both single-stream variants become less balanced across task types. Combining g_V and g_G achieves the strongest No-Goal exploration performance and maintains competitive Image-Goal and Point-Goal results, supporting our motivation that visual-semantic and geometric goal cues are complementary for a unified multi-task policy.

To validate the remaining design choices, we conduct ablations on the Image-Goal task, including the historical memory window size (k), the DSCC fusion design, the action-foresight coupling architecture, and the visual prediction horizon. As shown in Table 6, both shorter ($k = 4$) and longer ($k = 16$) windows perform worse than $k = 8$ under this protocol. Figure 6 provides a qualitative comparison of the planned trajectories for these memory window sizes. Removing explicit goal injection (*w/o GI*) also causes a clear performance drop, supporting goal-aware modulation in the evaluated configuration. For action-foresight coupling, the fully shared DiT outperforms decoupled and partially shared variants, supporting interaction between action and visual tokens in the evaluated shared model.

Table 8 further validates the asymmetric horizon design. Following NavDP, we keep the action horizon fixed at $H_{\text{act}} = 24$ to preserve long-horizon trajectory continuity, and vary the visual horizon H_{vis}. The best performance is obtained with short-horizon visual foresight ($H_{\text{vis}} = 1$), while longer

Table 6: Ablation studies on memory window size, DSCC module, and Action-Foresight architecture.

Memory Window			**DSCC**			**Coupling**		
Setting	SR (↑)	SPL (↑)	**Variant**	SR (↑)	SPL (↑)	**Variant**	SR (↑)	SPL (↑)
$k = 4$	46.6	44.1	–	–	–	Decoupled DiT	45.9	44.6
$k = 8$ (Ours)	**50.2**	**48.2**	w/o GI	44.5	42.2	Partially shared	47.5	45.4
$k = 16$	44.9	42.3	Ours	**50.2**	**48.2**	Ours	**50.2**	**48.2**



(a) $k = 4$ (b) $k = 8$ (Ours) (c) $k = 16$

Figure 6: Visualization of the effect of different memory window sizes on navigation trajectory planning.

Table 7: Ablation study of unified goal alignment across three navigation tasks.

Variant	Image-Goal		Point-Goal		No-Goal	
	SR (↑)	SPL (↑)	SR (↑)	SPL (↑)	Area (↑)	Time (↑)
Only g_V	**55.6**	**53.2**	62.1	58.6	166.3	69.9
Only g_G	47.9	45.3	**83.5**	**80.8**	168.4	72.5
Ours ($g_V + g_G$)	50.2	48.2	80.4	78.0	**171.1**	**75.6**

Table 8: Ablation study of asymmetric action-foresight horizons on Image-Goal navigation.

Setting	SR (↑)	SPL (↑)
$H_{\text{act}} = 24,\ H_{\text{vis}} = 1$	**50.2**	**48.2**
$H_{\text{act}} = 24,\ H_{\text{vis}} = 4$	46.2	43.8
$H_{\text{act}} = 24,\ H_{\text{vis}} = 8$	39.6	37.3
$H_{\text{act}} = 24,\ H_{\text{vis}} = 24$	30.4	28.1

visual horizons progressively degrade performance. The trend favors near-future visual context for conditioning long-horizon action generation. Appendix F.4 reports matched-representation controls and visual prediction quality.

F.4 Controlled Fusion, Coupling and Horizon Comparisons

All fusion controls use the same data, training budget and Image-Goal evaluation protocol. Direct Concat removes learned fusion; Simple Transformer replaces the directed fusion with symmetric attention. With $H_{\text{vis}} = 1$, Direct Concat, Simple Transformer and DSCC achieve 35.4/33.8, 41.9/40.2 and 50.2/48.2 SR/SPL, respectively (Table 3). Table 9 crosses Simple Transformer and DSCC with visual horizons 1 and 24. Using short-horizon foresight also improves Simple Transformer by 17.2/17.8 percentage points in SR/SPL.

For coupling, the Decoupled DiT uses disjoint action and visual DiTs that share only the conditioning context C. Its visual branch supplies auxiliary supervision through a separate, full-capacity DiT, while the action branch performs denoising independently. The partially shared variant has separate action/visual self-attention and shared cross-attention/FFN. The fully shared variant additionally permits action–visual token interaction in every shared self-attention layer. Their respective

Table 9: Fusion and visual-horizon controls. The action horizon is 24 throughout. Image-Goal SR/SPL are four-category macro averages (%).

Fusion	H_{vis}	SR	SPL
Simple Transformer	1	41.9	40.2
Simple Transformer	24	24.7	22.4
DSCC	1	**50.2**	**48.2**
DSCC	24	30.4	28.1

SR/SPL are 45.9/44.6, 47.5/45.4 and 50.2/48.2. Fully shared versus partially shared (+2.7/+2.8 points) assesses direct token interaction; fully shared versus decoupled (+4.3/+3.6 points) assesses the overall shared-coupling design.

Visual-horizon variants use a padded/masked $H_{\max} = 24$ representation, matching dimensionality, capacity and tensor-level compute. Table 10 evaluates decoded predictions against simulator-executed ground-truth observations. Both LPIPS [52] and patch-DINO similarity [37] degrade with longer horizons, consistent with the task-performance trend. VAE-only reconstruction provides a reference for compression fidelity. The prediction-quality trend parallels the navigation results, favoring short visual foresight when longer-horizon prediction becomes less reliable.

Table 10: Visual prediction quality on Image-Goal navigation. Lower LPIPS and higher patch-DINO similarity are better.

Setting	LPIPS ↓	Patch-DINO ↑
$H_{\text{vis}} = 1$	0.21	0.56
$H_{\text{vis}} = 8$	0.24	0.52
$H_{\text{vis}} = 24$	0.33	0.47
VAE-only reconstruction	0.039	0.705

F.5 Goal-Modulated Patch Relevance

Figure 7 visualizes the goal-conditioned patch scores $\alpha_{k,p}$ introduced in Section 3.2. Each 16×16 score map is overlaid on the corresponding RGB observation at six selected decision steps. Display-only normalization is applied; warmer colors denote higher relevance. The maps illustrate how goal-conditioned relevance varies across visual patches as the robot moves.

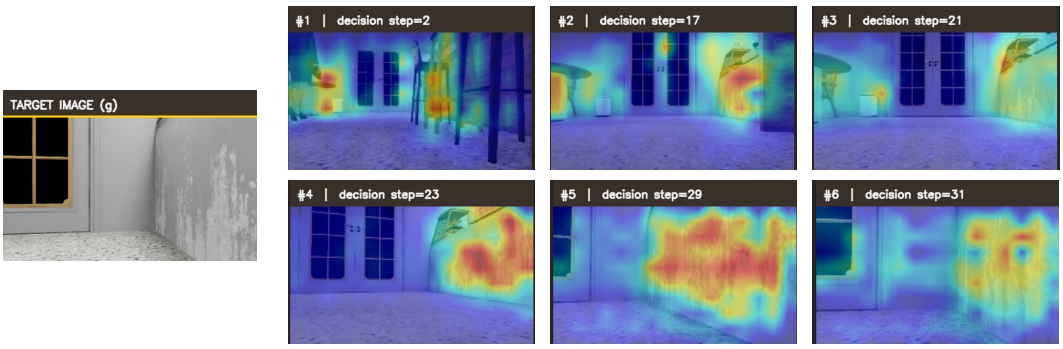


Figure 7: Target image (left) and goal-modulated patch relevance (right, in row-major order) at decision steps 2, 17, 21, 23, 29 and 31. Warmer colors indicate higher α after display-only normalization.

Table 11: Generalization performance across different embodiments.

Embodiment	Method	Image-Goal		Point-Goal		No-Goal	
		SR (↑)	SPL (↑)	SR (↑)	SPL (↑)	Area (↑)	Time (↑)
Dingo	NavDP	43.4	41.4	77.8	74.8	166.9	72.5
	Ours	**50.2**	**48.2**	**80.4**	**78.0**	**171.1**	**75.6**
G1	NavDP	37.2	35.6	60.3	56.8	133.0	65.2
	Ours	**42.9**	**42.5**	**64.8**	**63.1**	**147.6**	**68.3**
H2	NavDP	32.7	31.3	47.9	46.5	142.4	67.6
	Ours	**35.5**	**34.9**	**52.7**	**50.5**	**145.4**	**70.6**

F.6 Comparison with NavDP

We first evaluate the zero-shot adaptability of a single `WAM-Nav` policy across diverse robot embodiments without any parameter retraining. As summarized in Table 11, experiments are conducted on a wheeled Dingo platform together with Unitree G1 and H2 humanoid robots in the IsaacSim environment. The quantitative results demonstrate strong cross-embodiment generalization. In particular, on the G1 platform, `WAM-Nav` consistently outperforms NavDP across all evaluated task settings, achieving a 42.9% SR on Image-Goal navigation. On the more challenging H2 humanoid platform, `WAM-Nav` also achieves higher performance in all navigation tasks. These results support policy transfer across the evaluated robot platforms without retraining.

G Real-World Deployment Details

G.1 Hardware setup

As shown in Figure 8, our experimental platform is built on the Unitree G1 humanoid robot. The robot is equipped with an Intel RealSense D455 camera mounted on a pan-tilt servo; the policy receives RGB observations from fixed viewpoints. A backpack-mounted onboard computer equipped with an RTX 4060 GPU performs real-time perception and control. A power transformer provides stable power supply for the onboard devices. This hardware setup enables real-time perception, computation, and execution for our experiments.

G.2 Deployment Details

During deployment, the Intel RealSense D455 depth camera is mounted on the pan-tilt servo and configured with a 20° downward pitch angle to improve the observation of ground-level obstacles. The camera captures observations at 30 Hz; the policy uses the RGB stream. The WAM-Nav model is deployed as a server on the backpack-mounted onboard computer equipped with an RTX 4060 GPU for online inference. The model occupies approximately 1.3 GB of GPU memory, and the inference frequency is set to 1 Hz. The client module is also deployed on the onboard computer. It is responsible for forwarding the incoming observations to the server and receiving the predicted outputs. The predicted trajectories are tracked using Model Predictive Control (MPC), which further converts them into velocity commands for the robot motors. During experiments, the controller updates at 50 Hz, while each action has an execution period of 0.1 s; these are distinct from the 1 Hz policy inference frequency. Image goals are used in the meeting room and lobby, No-Goal exploration in the warehouse, and point goals in the parking lot. WAM-Nav and NavDP use the same starts, goals, sensing and controller. In the real-world warehouse trials, a trial succeeds if no obstacle collision occurs throughout 120 seconds. In the parking lot, the required path detours right around a motorcycle and cones with the left route blocked; observed NavDP failures included late collisions and repeated left-side trapping.

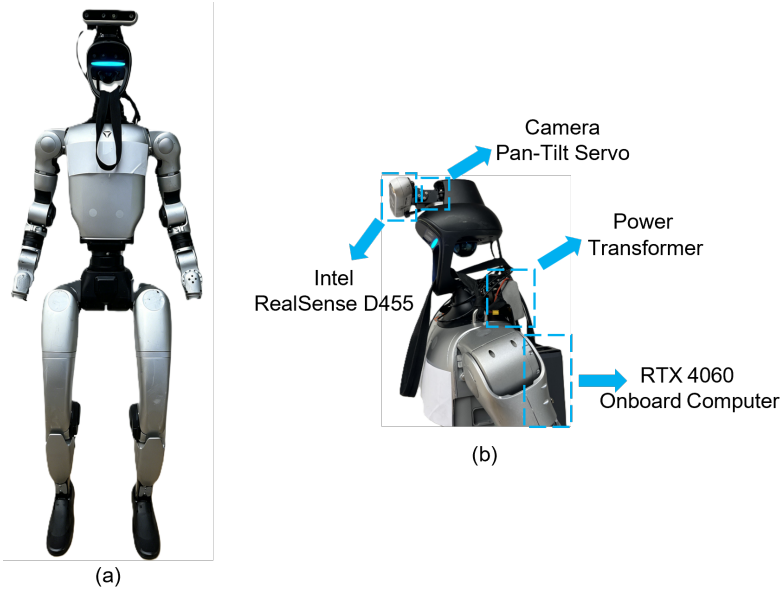


Figure 8: Robot Setup. (a) Unitree G1. (b) Detailed view of the hardware components.

H Additional Failure Case Analysis

In Figure 9, we visualize several typical failure cases. Case (a) shows that when the robot is close to a low obstacle, the limited field of view results in a weak perception of the obstacle, causing the robot to overlook it and eventually collide. Another common type of failure is illustrated in (b) and (c): when obstacles are located on the left or right side of the robot, the robot's body shape is not taken into account. The planned camera path can appear traversable while leaving insufficient clearance for the robot body, resulting in body-obstacle collisions.

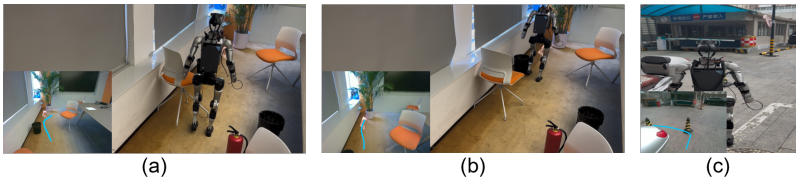


Figure 9: Failure Cases. The bottom-left inset shows the first-person view of the planned trajectory.