# TIC-VLA: A Think-in-Control Vision-Language-Action Model for Robot Navigation in Dynamic Environments

Zhiyu Huang*, Yun Zhang*, Johnson Liu, Rui Song, Chen Tang, Jiaqi Ma
University of California, Los Angeles
{zhiyuh, yun666, jiaqima}@ucla.edu

## Abstract

*Robots in dynamic, human-centric environments must follow language instructions while maintaining real-time reactive control. Vision-language-action (VLA) models offer a promising framework, but they assume temporally aligned reasoning and control, despite semantic inference being inherently delayed relative to real-time action. We introduce **Think-in-Control (TIC)-VLA**, a latency-aware framework that explicitly models delayed semantic reasoning during action generation. TIC-VLA defines a delayed semantic-control interface that conditions action generation on delayed vision-language semantic states and explicit latency metadata, in addition to current observations, enabling policies to compensate for asynchronous reasoning. We further propose a latency-consistent training pipeline that injects reasoning inference delays during imitation learning and online reinforcement learning, aligning training with asynchronous deployment. To support realistic evaluation, we present **DynaNav**, a physics-accurate, photo-realistic simulation suite for language-guided navigation in dynamic environments. Extensive experiments in simulation and on a real robot show that TIC-VLA consistently outperforms prior VLA models while maintaining robust real-time control under multi-second reasoning latency. Project website: https://ucla-mobility.github.io/TIC-VLA/*

## 1. Introduction

Robots operating in real-world, human-centric environments must react to dynamic scenes while following high-level natural language instructions [6]. Vision-language-action (VLA) models [10, 16, 46] have emerged as a promising paradigm by unifying perception, language understanding, and control within a single learning-based system. By incorporating large vision-language models (VLMs), these approaches enable semantic grounding, task reasoning, and instruction following [3, 4, 8, 28, 54, 56]. However, existing VLA systems rely on a hidden and impractical assumption: reasoning and control are temporally aligned. Vision-language reasoning is produced intermittently and often with substantial delay, while control must operate continuously as the robot moves and the environment changes. As a result, semantic representations frequently correspond to past world states, yet are consumed by the policy as if they were current, introducing systematic misalignment between thinking and control. This temporal mismatch is particularly pronounced on mobile robots with limited computation, where VLM inference can take seconds while control loops run at tens of hertz.

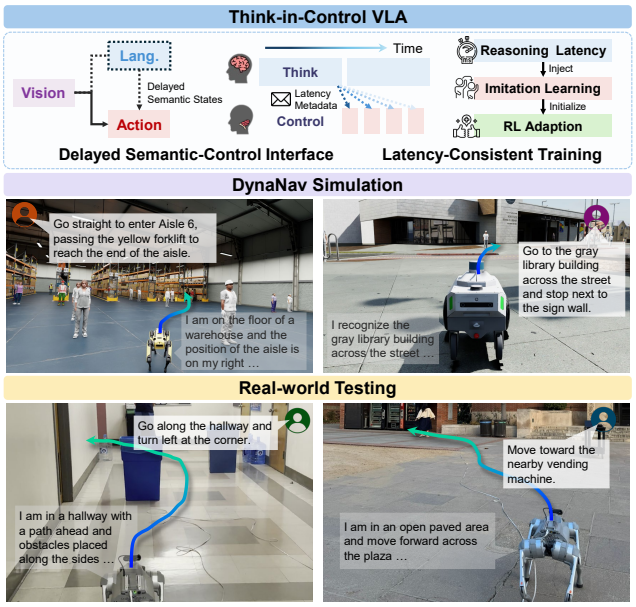


Figure 1. TIC-VLA enables real-time, language-conditioned navigation by decoupling slow vision-language reasoning from fast reactive control via a delayed semantic-control interface. A latency-consistent training strategy improves robustness under variable reasoning delays. Performance is demonstrated in the DynaNav simulation and real-world indoor and outdoor navigation tasks.

Most prior work on embodied navigation sidesteps this issue. Classical vision-language navigation (VLN) abstracts navigation into discrete viewpoint transitions, ignor-

ing embodiment, dynamics, and timing [11, 17, 23, 36, 47, 49]. Recent VLA systems rely on powerful GPUs and often pause execution during reasoning inference [7, 51], which is impractical in dynamic environments. Dual-system VLA architectures decouple reasoning and control [20, 42], but still assume that semantic outputs are temporally fresh, implicitly treating inference latency as negligible. We argue that latency in reasoning is not merely an engineering inefficiency but a fundamental modeling problem: delayed semantic information is neither represented nor accounted for during policy learning. Consequently, policies trained under idealized synchronous supervision can degrade severely when deployed under realistic inference latency.

To this end, we introduce **Think-in-Control VLA**, a latency-aware framework that explicitly exposes inference delay to the control policy. Rather than enforcing real-time constraints on semantic reasoning, TIC-VLA defines a *delayed semantic-control interface* that allows reasoning to proceed asynchronously while enabling robust real-time control. Specifically, the reasoning module produces delayed latent semantic representations together with explicit latency and ego-motion offset metadata, while the action policy conditions on this delayed semantic-control interface. Importantly, architectural decoupling alone is insufficient. Policies trained under idealized synchronous supervision fail when deployed with delayed semantic inputs. We therefore propose a *latency-consistent training pipeline* in which inference delays are explicitly injected during imitation learning and reinforcement learning. This enables the policy to learn to be robust to delayed semantic information, rather than overfitting to unrealistic training conditions.

To support realistic and reproducible evaluation, we develop **DynaNav**, a simulation suite featuring realistic rendering, dynamic human participants, physics-based execution, and diverse indoor and outdoor scenarios. DynaNav supports teleoperated data collection, online RL, and benchmarking. Experiments in both simulation and real-world deployment demonstrate that our proposed TIC-VLA model achieves improved and robust navigation under inference latency. Fig. 1 illustrates the key design of TIC-VLA and its performance in simulation and real-world environments. The primary contributions can be summarized as:

1. We introduce TIC-VLA with a delayed semantic-control interface that enables integration of temporally misaligned semantic features with real-time control.
2. We propose a latency-consistent training pipeline that aligns learning with asynchronous inference at deployment, yielding robust navigation under variable delays.
3. We present DynaNav, a realistic simulation suite and benchmark for language-guided navigation in dynamic environments, and we demonstrate TIC-VLA's strong performance in simulation and real-world.

## 2. Related Work

**Learning-based Visual Navigation**. Recent advances in robot navigation have shifted from traditional map-based pipelines toward end-to-end learning-based models. Diffusion policies [2, 18], imitation and reinforcement learning methods [15, 30, 44], and world modeling approaches [1, 29] have demonstrated strong performance in navigating complex environments without relying on maps. The integration of LLMs and VLMs into navigation tasks [13, 46, 48, 53, 55] has further expanded robots' semantic understanding and open-vocabulary reasoning capabilities, allowing them to follow flexible natural language commands. However, in most of these works, VLMs are employed as auxiliary modules rather than being fully integrated into the navigation pipeline. This limitation has motivated the development of VLA models, which unify perception, instruction, reasoning, and planning within a single framework.

**VLA for Navigation**. Recent studies increasingly employ VLA models for robotic navigation. Representative methods span direct action prediction and intermediate planning: NaVid [51] predicts next-step action from monocular RGB inputs, while NaVILA [7] generates mid-level linguistic actions executed by a visual locomotion policy. TrackVLA [41] combines language-based recognition with diffusion-based trajectory planning. Several works further explore generalist VLA frameworks: OmniVLA [16] supports multiple forms of instruction conditioning, including egocentric poses, images, and natural language, while NavFoM [52] demonstrates strong cross-embodiment performance. More recent systems address temporal reasoning and real-time execution, such as StreamVLN [43], MobileVLA [19], and dual-system VLA approaches like DualVLN [42], which balance deliberative reasoning and reactive control via asynchronous inference. Despite these advances, most existing VLA-based navigation models implicitly assume negligible inference latency, often relying on powerful GPUs or blocking execution during reasoning inference. In contrast, TIC-VLA is designed for latency-aware execution and robust on-device deployment.

**Navigation in Dynamic Environments.** Another line of work focuses on navigation in social and dynamic settings. Social-LLaVA [34] fine-tunes VLMs for social navigation, while Narrate2Nav [35] incorporates implicit language reasoning, social cues, and human intent into visual representations. Vi-LAD [12] distills socially compliant navigation knowledge from large VLMs into lightweight policies for real-time deployment. However, these methods typically rely on specialist policies at inference time and do not keep an in-the-loop VLM during execution. Evaluation is further limited by existing simulators. Arena [38] and UrbanSim [45] support dynamic agents but lack language-conditioned navigation. SocialNav-SUB [33] evaluates real-world so-

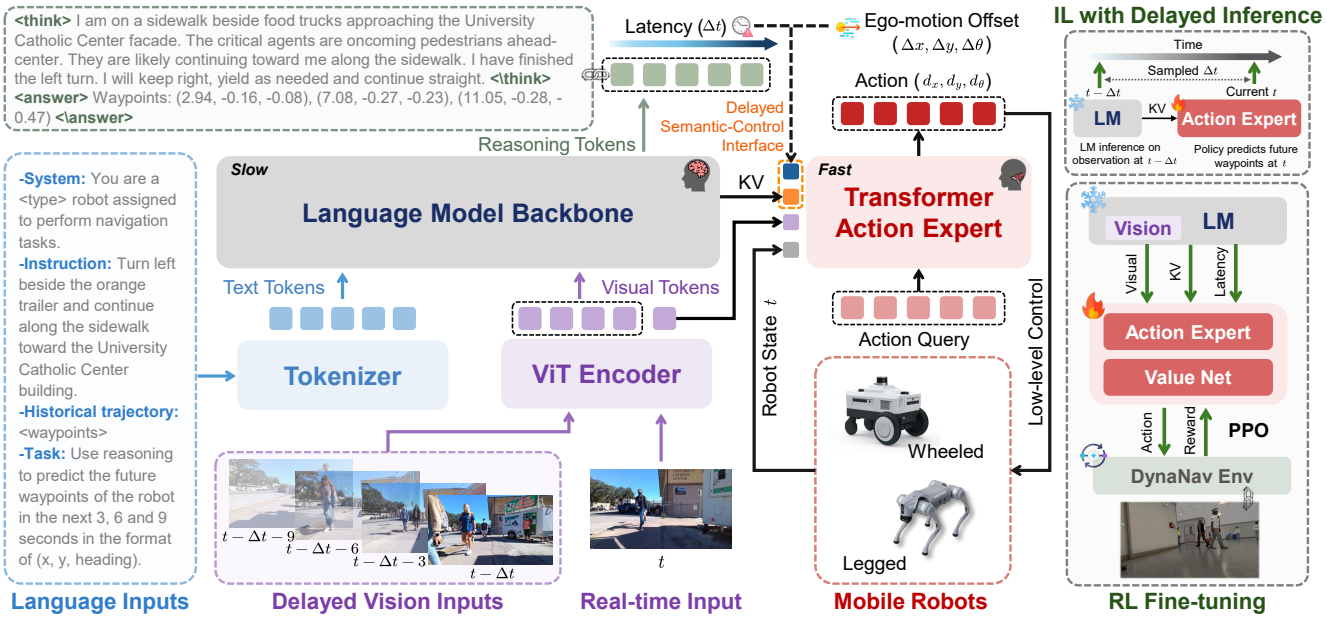


Figure 2. Overview of **TIC-VLA**. The architecture adopts a decoupled dual-system design with a fast action expert and a slow reasoning VLM. A shared vision encoder provides real-time observations to the policy and time-lagged observations to the VLM, where the delay arises naturally from slow inference. The delayed semantic-control interface (including delayed VLM KV cache features and latency metadata) is explicitly recorded. The Transformer-based action expert takes as input the current observation, robot state, and delayed semantic-control interface data to generate actions from learnable action queries via cross-attention. Multi-stage training combines imitation learning with delayed inference and reinforcement learning to ensure robustness to real-world, time-sensitive conditions.

cial navigation without a controllable simulator, while HA-VLN [9] relies on simplified observations and abstracted control. In contrast, *DynaNav* provides a physics-accurate simulation suite for language-guided navigation in human-centric environments.

# 3. Method

## 3.1. Problem Formulation

We consider an instruction-following navigation problem in which an embodied agent must follow natural language instructions to navigate dynamic environments. At each control timestep $t$, the agent receives: (1) a natural language instruction and context $\mathcal{I}$, specifying the navigation goal and historical trajecoty; (2) an egocentric observation history $\mathcal{O}_t = \{x_0, \ldots, x_t\}$, consisting of RGB frames $x_t \in \mathbb{R}^{H\times W\times 3}$; and (3) the robot state $s_t \in \mathbb{R}^3$, encoding ego-motion information such as linear velocity and angular velocity. Conditioned on these inputs, the agent must output an action $\mathbf{a}_t$ at each timestep to safely and efficiently progress toward the goal. The environment evolves in response to the agent's actions, and the episode continues until the agent reaches the goal or terminates.

We consider settings where a large-scale VLM is employed to interpret the natural-language instructions and provide semantic guidance. While such models are powerful, their reasoning often introduces non-negligible inference latency $\Delta t$. As a result, semantic outputs may become temporally misaligned with the agent's current observations and state, creating a key challenge for real-time navigation.

## 3.2. Think-in-Control VLA

An overview of the TIC-VLA framework is shown in Fig. 2. TIC-VLA adopts a dual-system execution paradigm in which high-level semantic reasoning and low-level control operate asynchronously. Crucially, rather than treating this as an architectural contribution, we explicitly model the resulting inference delay as part of the control problem. The key design principle is to expose reasoning latency to the action policy and train the policy to act under delayed semantic observations. We employ InternVL3-1B [57] as the vision-language backbone for semantic and instruction understanding. At a high level, a VLM performs semantic reasoning over delayed visual context and language instructions, while a reactive action policy executes at a high control frequency and never blocks on VLM inference. The action policy conditions on cached semantic representations together with explicit latency and ego-motion metadata, allowing it to interpret the delayed semantic information in the correct temporal context. This latency-aware semantic-control coupling enables robust navigation despite asynchronous and delayed reasoning updates.

**VLM Semantic Reasoning.** We formalize inference latency as a core variable in the system. We define the *ef-*

*fective reasoning latency* as $\Delta t = t_{\text{infer}} + t_{\text{elapse}} \geq 0$, which accounts for both the VLM inference time ($t_{\text{infer}}$) and the elapsed time since the last completed reasoning update ($t_{\text{elapse}}$). At the current timestep $t$, the VLM operates on visual observations anchored at time $t - \Delta t$, rather than the current frame. Given a set of historical frames $\mathcal{X}^{\text{vlm}}_{t-\Delta t} = \{x_{t-\Delta t-\delta} \mid \delta \in \{0, 3, 6, 9\}\}$, the VLM performs semantic reasoning conditioned on the instruction $\mathcal{I}$. The resulting output, denoted $\mathcal{R}_{t-\Delta t}$, encodes high-level scene understanding, critical object identification, intent prediction, and future target waypoints derived from delayed observations. The reasoning results, including predicted waypoints, are generated relative to the time of inference start $t-\Delta t$, rather than the current control timestep. This explicit temporal anchoring allows the downstream policy to know when the reasoning was produced, rather than treating them as instantaneous or noisy signals.

**Latency-Aware Action Policy**. The action policy $\pi_\theta$ runs at a high frequency and is responsible for real-time planning. At each timestep $t$, it conditions on four categories of inputs: (1) the current visual observation $x_t$; (2) the current robot state $s_t$; (3) the most recent semantic hidden state $\mathcal{S}_{t-\Delta t}$ produced by the VLM (i.e., the last-layer key-value cache); (4) explicit latency metadata, including the effective latency $\Delta t$ and the corresponding motion offsets $\Delta \mathbf{p}_t = (\Delta x, \Delta y, \Delta \theta)$ accumulated since reasoning generation. Providing both delayed semantic states and latency metadata establishes a *delayed semantic-control interface*: semantic features describe a past state of the world, while the control policy is responsible for reinterpreting them in the current frame. This allows the policy to reason consistently about delayed semantics as the robot moves during inference. The policy outputs a short horizon of future actions:

$$\mathbf{a}_t = \{a_t^1, \ldots, a_t^T\} = \pi_\theta\big(\mathcal{S}_{t-\Delta t}, x_t, s_t, \Delta t, \Delta \mathbf{p}_t\big), \quad (1)$$

where each $a_t^i \in \mathbb{R}^3$ represents a continuous action. The action chunks are integrated into a short-horizon trajectory, and a target point is chosen for execution.

As shown in Fig. 3(a), the action policy utilizes a dedicated action query token that attends to the scene context through a stack of cross-attention Transformer layers. Visual tokens and VLM cache features are first projected into a shared latent space via MLP layers, while the robot state and latency metadata are encoded and added with positional embeddings. These inputs are concatenated as key-value tokens, and the updated action query representation is passed through an MLP to generate the action outputs.

**Asynchronous Semantic Reasoning and Control.** TIC-VLA operates in a closed-loop asynchronous manner. The VLM periodically updates semantic reasoning based on delayed visual inputs, while the action policy continuously executes without waiting for inference to complete. The cached VLM hidden state is updated as follows:

$$(\mathcal{S}_t^{\text{cache}}, \mathcal{R}_t^{\text{cache}}) = \begin{cases} (\mathcal{S}_t, \mathcal{R}_t), \text{if inference finishes at } t, \\ (\mathcal{S}_{t^-}^{\text{cache}}, \mathcal{R}_{t^-}^{\text{cache}}), \text{otherwise}, \end{cases} \quad (2)$$

where $t^-$ denotes the most recent timestep prior to $t$ at which inference was completed. At every control timestep, the action policy conditions on the most recent cached VLM hidden state $\mathcal{S}_{t-\Delta t} = \mathcal{S}_t^{\text{cache}}$. In addition, by conditioning action generation on latency metadata and ego-motion, TIC-VLA could reinterpret stale semantic information in the current state. An illustration of the asynchronous inference process is shown in Fig. 3(d), and latency is measured as the sum of two parts: VLM reasoning inference time and the elapsed time since the last finished inference.

### 3.3. Latency-Consistent Training Pipeline

We adopt a three-stage training pipeline designed to enforce *latency consistency* between training and deployment. An overview of the pipeline is provided in Fig. 3(c).

**Supervised Fine-tuning of the VLM.** We first fine-tune the VLM on structured semantic reasoning data collected from both simulation and real-world environments. Training samples are automatically annotated using GPT-5. Given past and future image sequences, the robot's positional context, and the corresponding trajectory, GPT-5 produces: (1) a long-horizon instruction describing the navigation goal, and (2) a concise, structured reasoning trace capturing critical objects identification, intention prediction, and resulting action. Details are provided in the supplementary material.

During this stage, the vision encoder is kept frozen. The language model is trained to produce reasoning tokens and waypoint predictions conditioned on visual tokens and instructions. We optimize the standard autoregressive cross-entropy loss $\mathcal{L}_l$ over the target token sequence. We mix waypoint-only and scene-reasoning-augmented targets during training for flexible prompting at inference time.

**Imitation Learning under Reasoning Latency.** To compensate for the uncertain delay in semantic reasoning, we perturb originally aligned and synchronous demonstrations to synthesize training data with delayed semantic reasoning. Specifically, we sample reasoning delays $\Delta t$ uniformly from $[0, 10]$ seconds and condition the policy on: (1) the current image input and robot state, (2) KV cache features from the delayed VLM reasoning output, (3) explicit latency metadata. This exposes the policy to a range of temporally misaligned semantic representations during training. The action policy $\pi_\theta$ is trained via imitation learning using human demonstration trajectories. As low-level control actions are not available, we integrate the predicted actions forward to obtain positions $(x, y, \theta)$ over the prediction horizon and compare them against ground-truth trajec-

tories using a smooth $L_1$ loss:

$$\mathcal{L}_a = \frac{1}{T}\sum_{i=1}^{T} \text{SmoothL1}\big(\hat{p}_t^{(i)} - p_t^{(i)}\big), \tag{3}$$

where $\hat{p}_t^{(i)}$ is the predicted pose at sub-timestep $i$, and $p_t^{(i)}$ is the ground-truth pose.

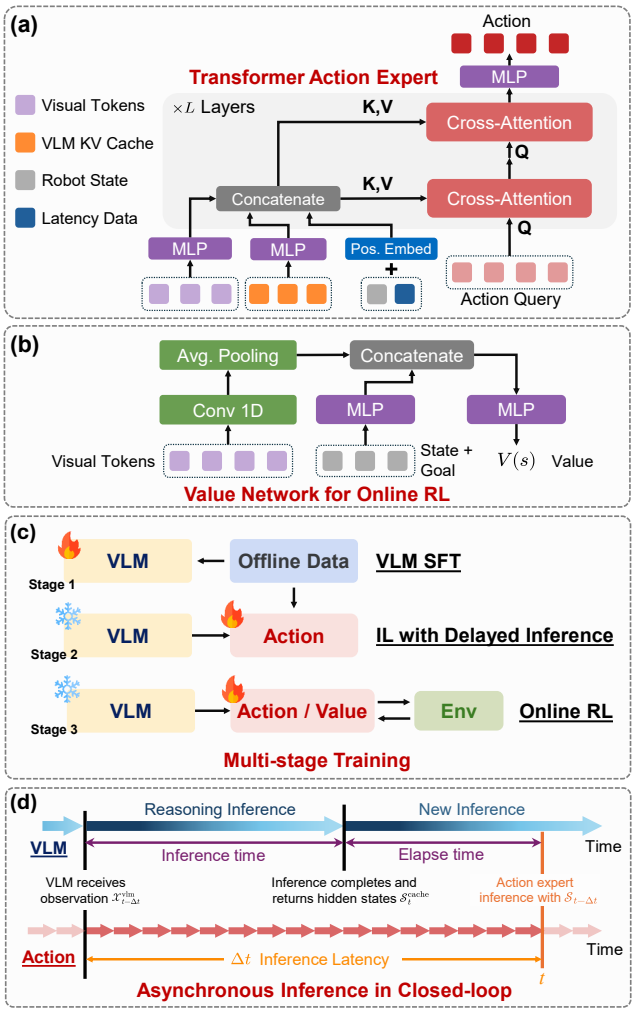


Figure 3. Details of TIC-VLA action policy structure, training, and asynchronous execution. (a) Latency-aware action policy that predicts action chunks from multimodal inputs. (b) Value network used during online reinforcement learning. (c) Three-stage latency-consistent training pipeline combining VLM supervision, imitation learning, and reinforcement learning. (d) Asynchronous inference and control with explicit latency modeling.

**Reinforcement Learning with Asynchronous Guidance.** While imitation learning with synthesized reasoning delays provides a strong initialization that accounts for inference latency, the resulting training data distribution remains mismatched to the closed-loop distribution induced by coupled, asynchronous reasoning-action interactions. Motivated by prior work [5, 25, 31], we fine-tune only the action policy using RL while keeping the vision encoder and language model frozen. This allows the policy to learn to interpret delayed VLM guidance, handle dynamic agents, and mitigate variability introduced by asynchronous inference.

We construct a simulation environment with dynamic human participants and train the policy using Proximal Policy Optimization (PPO) [37]. The value network, shown in Fig. 3(b), takes as input the current image tokens, the goal position, and the robot state, and outputs the estimated state value. The policy outputs a Gaussian action distribution, where the action derived from the predicted trajectory is the mean, and the standard deviation is learned during training. The reward function is defined as:

$$r_t = w_g r_t^{\text{goal}} + w_p r_t^{\text{progress}} + w_c r_t^{\text{collision}} + w_s r_t^{\text{speed}}, \tag{4}$$

where $r_t^{\text{goal}}$ rewards reaching the target, $r_t^{\text{progress}}$ encourages progress toward the goal, $r_t^{\text{collision}}$ penalizes collisions with humans or static obstacles, and $r_t^{\text{speed}}$ penalizes both excessively slow motion and overly fast speed. $w_g, w_p, w_c$, and $w_s$ are weights for these terms.

To further improve robustness under asynchronous deployment, we inject stochastic inference delays following each VLM update to mimic the latency characteristics observed on edge hardware. This enforces consistency between training and execution conditions. Additional details are provided in the supplementary material.

### 3.4. DynaNav

Evaluating our proposed latency-aware VLA framework requires more realistic navigation benchmarks than classic VLN benchmarks such as R2R [26], VLN-CE [23], and RxR [24], which operate in small indoor environments and abstract navigation as viewpoint transitions without physical interaction. VLN-PE [40] and GRUtopia [39] support embodied evaluation but do not model navigation among dynamic human participants, while SocialHM3D [14] and HA-VLN [9] incorporate human agents with limited visual realism. To fill this gap, DynaNav provides a realistic benchmark integrating language-guided navigation, large-scale scenes, diverse human agents, and physics-based robot control.

**Simulation Environments.** We construct simulation environments in Isaac Sim [21], with realistic, controllable dynamic interactions. Four representative scenes (i.e., warehouse, hospital, office, and outdoor sidewalk) are designed to capture a broad range of navigation contexts. Human behaviors are modeled using Isaac Sim's built-in human simulation tools, supporting behaviorally plausible pedestrian movement. Our simulation setup supports both wheeled (Nova Carter) and quadruped (Boston Dynamics Spot) robots. We develop custom robot behavior scripts that allow two modes of operation: (1) Human teleoperation mode,

which enables manual control for collecting expert demonstrations, and (2) End-to-end model control mode, which allows direct control of the robot with end-to-end planning models.

**Designed for Scalable RL Training.** We employ Isaac Lab [32] and leverage its GPU-accelerated simulation to build environments for scalable reinforcement learning training. We implement a custom human behavior control script to generate human movements within the environment. Human-robot and robot-scene interactions are fully physics-based with realistic contact dynamics. This setup allows us to run a number of parallel environments, enabling scalable and efficient RL training.

**Diverse Benchmark Tasks.** We develop a comprehensive benchmark of 85 test cases to evaluate navigation performance across diverse conditions. Tasks vary along three dimensions: (1) *Crowd Density*, ranging from empty spaces to densely populated settings, capturing different levels of dynamic complexity. (2) *Navigation Distance*, adjusted from short-term navigation to long-horizon planning, reflecting increasing navigation difficulty. (3) *Scene Type*, with evaluation conducted across four distinct environments to assess robustness to varying spatial layouts and human behaviors. For each task, standardized initial and goal positions are specified with a language instruction. Additional details are provided in the supplementary material.

# 4. Experiments

## 4.1. Experimental Setup

**Datasets**. We train the model using three datasets featuring dynamic human-robot interactions: (1) SCAND [22], which contains 8.7 hours of robot-driven trajectories across diverse social environments; (2) GND [27], which comprises over 11 hours of recorded data collected in various campus environments; (3) DynaNav simulation dataset, collected using our designed dynamic simulation environments, containing 5.1 hours of robot navigation data across multiple scene types.

**Implementation Details**. For VLM SFT, we employ full-parameter training due to the compact size of TIC-VLA. Training is performed using Distributed Data Parallel on eight NVIDIA L40S GPUs, with a batch size of 2 per GPU. AdamW is used as the optimizer with a cosine learning rate schedule, initialized at $2 \times 10^{-5}$. For training the action expert, we increase the batch size to 16 per GPU and set the initial learning rate to $2 \times 10^{-4}$. RL fine-tuning of the action policy is conducted on a single NVIDIA L40S GPU for 400 iterations across three tasks in three environments. Additional details, including hyperparameters and data preprocessing, are provided in the supplementary material.

**Baselines.** We evaluate TIC-VLA against two categories methods. Point-goal navigation policies are included as reference baselines to contextualize task difficulty: (1) a vanilla Behavior Cloning (BC) policy that maps RGB observations and point-goal commands directly to actions; (2) a vanilla RL policy trained on RGB observations and point-goal commands; (3) NavDP [2], a point-goal-conditioned diffusion-based navigation policy. The primary language-guided navigation baselines are listed below: (1) NaVILA [7], a hierarchical VLA model that translates language instructions into mid-level commands; (2) Uni-NaVid [50], a unified video-based VLA model trained across multiple navigation tasks; and (3) DualVLN [42], a dual-system VLA model. These baselines are fine-tuned on the same datasets to ensure a fair and controlled comparison.

**Evaluation metrics**. We adopt the following evaluation metrics: (1) Navigation Error (NE): the final distance between the agent and the goal; (2) Success Rate (SR): the percentage of episodes in which the agent stops within 1 meter of the goal; (3) Success weighted by Path Length (SPL): SR weighted by the ratio between the shortest path length and the actual path length, penalizing inefficient trajectories; (4) Collision Rate (CR): the percentage of episodes in which the agent collides with static obstacles or humans, quantifying the safety of navigation behavior.

**Real-world experimental setup**. To evaluate real-time performance on edge devices, we deploy the model on two platforms with different power budgets: an NVIDIA Jetson Orin NX (25W) and an RTX 4060 Laptop GPU (50W), representing typical edge computing capabilities. An RTX A6000 GPU is used only when the baselines cannot run on these devices. The navigation policy is executed on a Unitree Go2 quadruped robot in real-world navigation tasks. We employ FlashAttention to improve inference efficiency. Performance is measured by the average success rate. An episode is considered a failure if manual intervention is required to prevent collisions.

## 4.2. Simulation Testing

**Performance on the DynaNav benchmark.** All experiments are conducted on an NVIDIA L40S GPU, with the action policy running at 10 Hz and asynchronous VLM reasoning at 0.5 Hz. Results are summarized in Tab. 1. Despite relying solely on egocentric observations and language instructions, without privileged goal or map information, TIC-VLA achieves strong performance. Notably, point-goal methods bypass vision-language inference and therefore incur no reasoning latency. TIC-VLA (no RL) performs comparably to NavDP, a point-goal method with privileged state access, and outperforms BC and RL baselines trained without language. With additional RL fine-tuning, TIC-VLA achieves the highest success rate (SR) and the lowest collision rate (CR). TIC-VLA also substantially outperforms prior VLA/VLN methods (Uni-NaVid, NaVILA, DualVLN), which are designed for abstract naviga-

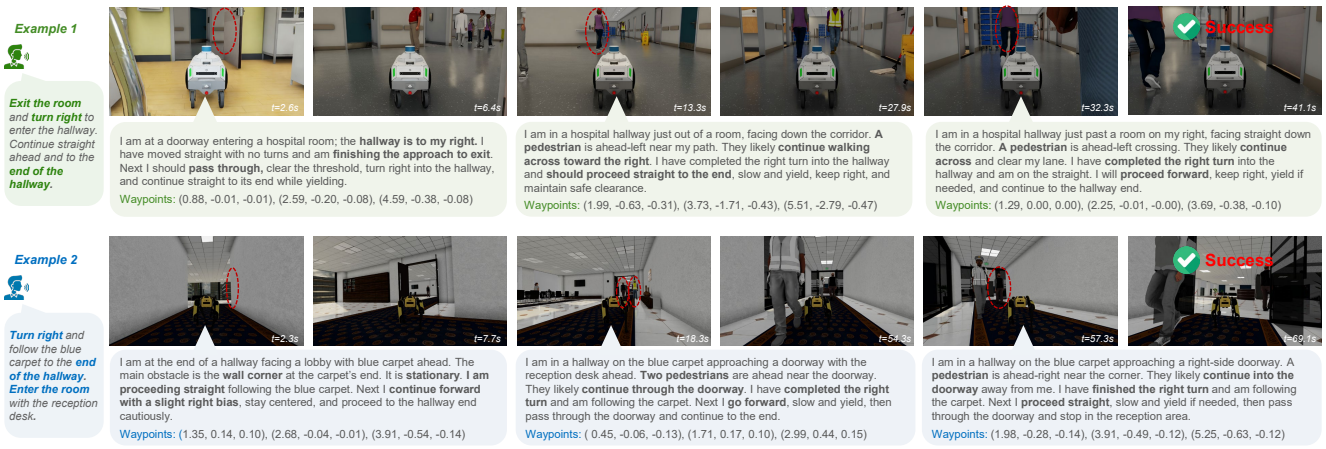


Figure 4. Qualitative results of TIC-VLA closed-loop performance in DynaNav hospital (top) and office (bottom) environments.

tion with discrete actions, highlighting their limitations in physically realistic, dynamic environments. Blocking control with synchronous VLM inference leads to a marked performance drop even under modest latency, underscoring the importance of TIC-VLA's asynchronous design. Fig. 4 provides qualitative examples of accurate reasoning and interactive navigation in dynamic environments.

Table 1. Performance of TIC-VLA and baseline methods on the DynaNav benchmark. BC, RL, and NavDP are goal point-based.

| Method | NE (↓) | SR (↑) | SPL (↑) | CR (↓) |
|---|---|---|---|---|
| BC Policy | 9.96 | 45.88 | 41.52 | 35.29 |
| RL Policy | 12.20 | 30.59 | 28.45 | 36.47 |
| NavDP | **8.61** | 54.12 | **52.62** | 30.59 |
| Uni-NaVid | 15.90 | 22.35 | 19.61 | 49.41 |
| NaVILA | 17.20 | 28.24 | 25.51 | 48.24 |
| DualVLN | 16.45 | 30.59 | 27.82 | 47.06 |
| **TIC-VLA (Sync.)** | 16.31 | 32.94 | 29.64 | 41.18 |
| **TIC-VLA (no RL)** | 10.85 | 47.06 | 42.41 | 34.12 |
| **TIC-VLA** | 10.55 | **55.29** | 50.29 | **28.24** |

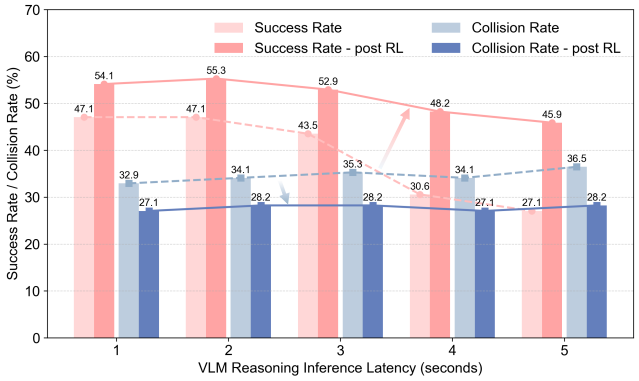


Figure 5. The effect of VLM asynchronous reasoning inference latency in TIC-VLA on task performance.

Table 2. Influence of semantic interface and latency training.

| Interface | Latency | NE (↓) | SR (↑) | SPL (↑) | CR (↓) |
|---|---|---|---|---|---|
| Waypoint | × | 21.17 | 16.47 | 15.89 | 47.06 |
| Waypoint | ✓ | 20.32 | 22.35 | 18.34 | 42.35 |
| KV Cache | × | 16.74 | 30.59 | 28.31 | 40.00 |
| KV Cache | ✓ | **10.85** | **47.06** | **42.41** | **34.12** |

**Latency Robustness Analysis.** Fig. 5 reports performance under increasing VLM reasoning inference latency for TIC-VLA in simulation, before and after RL fine-tuning. As latency increases, the IL-based action expert exhibits a noticeable decline in success rate, indicating higher sensitivity to delayed reasoning updates. In contrast, the RL-fine-tuned policy maintains consistently higher success rates across all latency settings, demonstrating improved robustness to inference latency. This result highlights the effectiveness of RL fine-tuning in mitigating latency-induced performance degradation. Collision rates are insensitive to inference latency, suggesting that the asynchronous policy preserves reactivity independent of reasoning speed.

**Influence of Semantic-Control Interface and Latency Awareness.** We evaluate the impact of different delayed semantic-control interfaces and latency-awareness on TIC-VLA. Specifically, we compare interface variants that use waypoint-based guidance and KV-cache-based features, each trained with and without explicit latency-aware modeling and training. As shown in Tab. 2, using KV-cache features significantly improves navigation success, and latency-awareness enhances performance under asynchronous inference. The waypoint-based interface leads to inferior performance due to its sparsity and potential inconsistency with the agent's local observations. Combining both the KV cache feature interface and latency-aware modeling and training achieves the best overall performance.

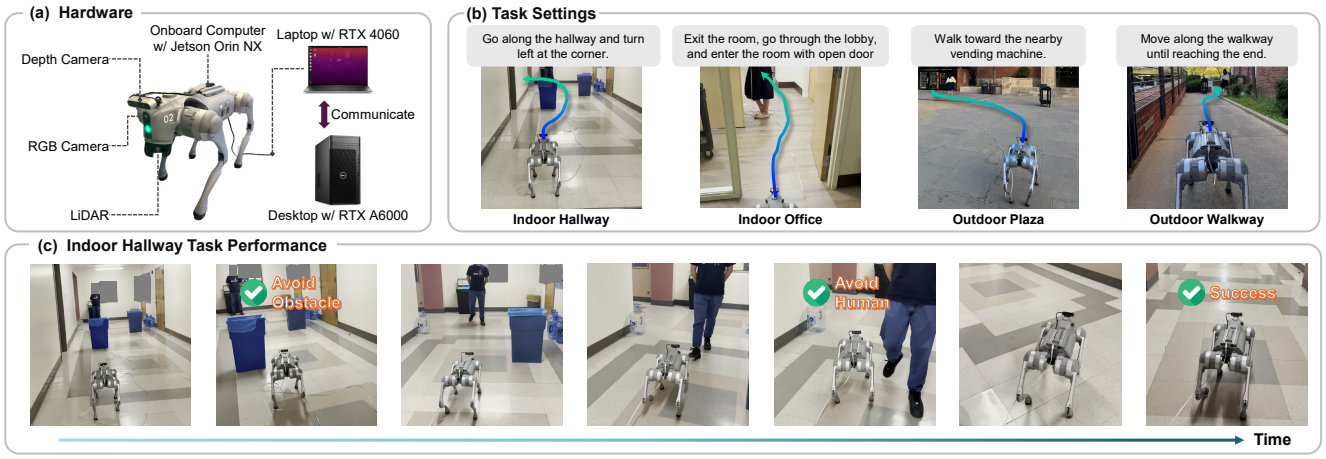


Figure 6. Real-world evaluation of TIC-VLA. (a) Hardware configuration, including the robot platform and computation setup. (b) Designed indoor and outdoor vision-language navigation tasks. (c) Qualitative results from an indoor hallway navigation task, showing the robot following natural language instructions while avoiding obstacles and humans and reaching the goal.

### 4.3. Real-world Testing

We design four real-world navigation tasks: (1) an indoor hallway with dynamic human traffic and static obstacles, (2) an office environment with cluttered layouts, (3) an outdoor plaza environment, and (4) an outdoor walkway scenario with uneven terrain. Task descriptions and hardware configurations are illustrated in Fig. 6. All evaluations are conducted in a zero-shot setting, without additional training data. For each task, we perform five trials and report the average success rate. During deployment, real-time RGB images from the front-facing camera of a Unitree Go2 robot are streamed to TIC-VLA for inference.

Table 3. Real-world testing results. Runtimes for the dual system are reported as (x/x) for the action policy and VLM reasoning.

| Method | Platform | Success Rate (↑) | Runtime (ms) |
|---|---|---|---|
| TIC-VLA (no RL) | 4060 | 0.70 | – |
| TIC-VLA | 4060 | **0.85** | 85.73/3430.73 |
| TIC-VLA | Orin NX | 0.75 | 120.27/4831.73 |
| TIC-VLA | A6000 | 0.80 | 32.70/1681.66 |
| Dual-VLN (7B) | A6000 | 0.50 | 299.92/1534.67 |
| NaVILA (7B) | A6000 | 0.35 | 4106.62 |

We compare TIC-VLA against DualVLN [42] and NaVILA [7], with results summarized in Tab. 3. TIC-VLA outperforms prior VLA baselines despite operating with significantly smaller models and lower compute budgets. RL fine-tuning further improves performance, particularly in scenarios with dynamic human interactions. TIC-VLA maintains high success rates when deployed on edge hardware (Jetson Orin NX) under multi-second reasoning latency, validating the effectiveness of explicit latency modeling for real-time control. Moreover, deployment on a remote server yields only limited performance gains due to communication delays.

### 4.4. Ablation Study

**Influence of Reasoning at Test Time.** We evaluate the effect of explicit VLM reasoning during inference by comparing the model (after RL fine-tuning) with and without reasoning-token outputs. As shown in Tab. 4, enabling reasoning improves all navigation metrics but incurs increased inference latency (0.5 Hz in simulation). Disabling reasoning significantly reduces VLM forward overhead (4 Hz in simulation), at the cost of significantly degraded performance. These results indicate that explicit reasoning at test time improves task success, while the proposed latency-aware design mitigates its associated overhead. Additional ablation results are provided in the supplementary material.

Table 4. Effect of VLM reasoning at test time.

| Inference | NE (↓) | SR (↑) | SPL (↑) | CR (↓) |
|---|---|---|---|---|
| W/o Reasoning | 14.23 | 40.00 | 34.22 | 25.88 |
| W/ Reasoning | **10.55** | **55.29** | **50.29** | **28.24** |

## 5. Conclusions

We present TIC-VLA, a latency-aware vision-language-action framework designed to explicitly address the temporal mismatch between computationally intensive semantic reasoning and real-time control. By introducing a delayed semantic-control interface and training policies under realistic inference latency, TIC-VLA enables robust and effective language-guided navigation under substantial delay. Extensive simulation and real-world experiments demonstrate that TIC-VLA consistently outperforms prior VLA approaches in both robustness and task performance. In future work, we plan to extend this framework to robot manipulation tasks, improve reasoning-action alignment, and evaluate its effectiveness in more dynamic and complex environments.

## References


[1] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025. 2

[2] Wenzhe Cai, Jiaqi Peng, Yuqiang Yang, Yujian Zhang, Meng Wei, Hanqing Wang, Yilun Chen, Tai Wang, and Jiangmiao Pang. Navdp: Learning sim-to-real navigation diffusion policy with privileged information guidance. *arXiv preprint arXiv:2505.08712*, 2025. 2, 6

[3] Mateo Guaman Castro, Sidharth Rajagopal, Daniel Gorbatov, Matt Schmittle, Rohan Baijal, Octi Zhang, Rosario Scalise, Sidharth Talia, Emma Romig, Celso de Melo, et al. Vamos: A hierarchical vision-language-action model for capability-modulated and steerable navigation. *arXiv preprint arXiv:2510.20818*, 2025. 1

[4] Bhargav Chandaka, Gloria X Wang, Haozhe Chen, Henry Che, Albert J Zhai, and Shenlong Wang. Human-like navigation in a world built for humans. *arXiv preprint arXiv:2509.21189*, 2025. 1

[5] Hanyang Chen, Mark Zhao, Rui Yang, Qinwei Ma, Ke Yang, Jiarui Yao, Kangrui Wang, Hao Bai, Zhenhailong Wang, Rui Pan, et al. Era: Transforming vlms into embodied agents via embodied prior learning and online reinforcement learning. *arXiv preprint arXiv:2510.12693*, 2025. 5

[6] Ziyi Chen, Yingnan Guo, Zedong Chu, Minghua Luo, Yanfen Shen, Mingchao Sun, Junjun Hu, Shichao Xie, Kuan Yang, Pei Shi, et al. Socialnav: Training human-inspired foundation model for socially-aware embodied navigation. *arXiv preprint arXiv:2511.21135*, 2025. 1

[7] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024. 2, 6, 8

[8] Wonje Choi, Woo Kyung Kim, Minjong Yoo, and Honguk Woo. Embodied cot distillation from llm to off-the-shelf agents. In *Proceedings of the 41st International Conference on Machine Learning*, pages 8702–8721, 2024. 1

[9] Yifei Dong, Fengyi Wu, Qi He, Heng Li, Minghan Li, Zebang Cheng, Yuxuan Zhou, Jingdong Sun, Qi Dai, Zhi-Qi Cheng, et al. Ha-vln: A benchmark for human-aware navigation in discrete-continuous environments with dynamic multi-human interactions, real-world validation, and an open leaderboard. *arXiv preprint arXiv:2503.14229*, 2025. 3, 5

[10] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025. 1

[11] Ainaz Eftekhar, Rose Hendrix, Luca Weihs, Jiafei Duan, Ege Caglar, Jordi Salvador, Alvaro Herrasti, Winson Han, Eli VanderBil, Aniruddha Kembhavi, et al. The one ring: a robotic indoor navigation generalist. *arXiv preprint arXiv:2412.14401*, 2024. 2

[12] Mohamed Elnoor, Kasun Weerakoon, Gershom Seneviratne, Jing Liang, Vignesh Rajagopal, and Dinesh Manocha. Vilad: Vision-language attention distillation for socially-aware robot navigation in dynamic environments. *arXiv preprint arXiv:2503.09820*, 2025. 2

[13] Chen Gao, Liankai Jin, Xingyu Peng, Jiazhao Zhang, Yue Deng, Annan Li, He Wang, and Si Liu. Octonav: Towards generalist embodied navigation. *arXiv preprint arXiv:2506.09839*, 2025. 2

[14] Zeying Gong, Tianshuai Hu, Ronghe Qiu, and Junwei Liang. From cognition to precognition: A future-aware framework for social navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9122–9129. IEEE, 2025. 5

[15] Honglin He, Yukai Ma, Wayne Wu, and Bolei Zhou. From seeing to experiencing: Scaling navigation foundation models with reinforcement learning. *arXiv preprint arXiv:2507.22028*, 2025. 2

[16] Noriaki Hirose, Catherine Glossop, Dhruv Shah, and Sergey Levine. Omnivla: An omni-modal vision-language-action model for robot navigation. *arXiv preprint arXiv:2509.19480*, 2025. 1, 2

[17] Junjun Hu, Jintao Chen, Haochen Bai, Minghua Luo, Shichao Xie, Ziyi Chen, Fei Liu, Zedong Chu, Xinda Xue, Botao Ren, et al. Astranav-world: World model for foresight control and consistency. *arXiv preprint arXiv:2512.21714*, 2025. 2

[18] Zichao Hu, Chen Tang, Michael J Munje, Yifeng Zhu, Alex Liu, Shuijing Liu, Garrett Warnell, Peter Stone, and Joydeep Biswas. Composablenav: Instruction-following navigation in dynamic environments via composable diffusion. *arXiv preprint arXiv:2509.17941*, 2025. 2

[19] Ting Huang, Dongjian Li, Rui Yang, Zeyu Zhang, Zida Yang, and Hao Tang. Mobilevla-r1: Reinforcing vision-language-action for mobile robots. *arXiv preprint arXiv:2511.17889*, 2025. 2

[20] InternRobotics. Internvla-n1: An open dual-system navigation foundation model with learned latent plans. https://huggingface.co/InternRobotics/InternVLA-N1, 2025. Accessed: 2025-10-10. 2

[21] Isaac-sim development team. IsaacSim: An open-source robotics simulation platform on NVIDIA Omniverse. https://github.com/isaac-sim/IsaacSim, 2025. Accessed: 2025-10-09; version v5.0.0. 5

[22] Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7 (4):11807–11814, 2022. 6

[23] Jacob Krantz, Erik Wijmans, Arjun Majundar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision and language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 5

[24] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical*

*Methods in Natural Language Processing (EMNLP)*, pages 4392–4412, 2020. 5

[25] Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025. 5

[26] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. 2019. 5

[27] Jing Liang, Dibyendu Das, Daeun Song, Md Nahid Hasan Shuvo, Mohammad Durrani, Karthik Taranath, Ivan Penskiy, Dinesh Manocha, and Xuesu Xiao. Gnd: Global navigation dataset with multi-modal perception and multi-category traversability in outdoor campus environments. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2383–2390. IEEE, 2025. 6

[28] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025. 1

[29] Wei Liu, Huihua Zhao, Chenran Li, Joydeep Biswas, Billy Okal, Pulkit Goyal, Yan Chang, and Soha Pouya. X-mobility: End-to-end generalizable navigation via world modeling. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7569–7576. IEEE, 2025. 2

[30] Wei Liu, Huihua Zhao, Chenran Li, Joydeep Biswas, Soha Pouya, and Yan Chang. Compass: Cross-embodiment mobility policy via residual rl and skill synthesis. *arXiv preprint arXiv:2502.16372*, 2025. 2

[31] Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025. 5

[32] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025. 6

[33] Michael J Munje, Chen Tang, Shuijing Liu, Zichao Hu, Yifeng Zhu, Jiaxun Cui, Garrett Warnell, Joydeep Biswas, and Peter Stone. Socialnav-sub: Benchmarking vlms for scene understanding in social robot navigation. *arXiv preprint arXiv:2509.08757*, 2025. 2

[34] Amirreza Payandeh, Daeun Song, Mohammad Nazeri, Jing Liang, Praneel Mukherjee, Amir Hossain Raj, Yangzhe Kong, Dinesh Manocha, and Xuesu Xiao. Social-llava: Enhancing robot navigation through human-language reasoning in social spaces. *arXiv preprint arXiv:2501.09024*, 2024. 2

[35] Amirreza Payandeh, Anuj Pokhrel, Daeun Song, Marcos Zampieri, and Xuesu Xiao. Narrate2nav: Real-time visual navigation with implicit language reasoning in human-centric environments. *arXiv preprint arXiv:2506.14233*, 2025. 2

[36] Sonia Raychaudhuri and Angel X Chang. Semantic mapping in indoor embodied ai-a survey on advances, challenges, and future directions. *Transactions on Machine Learning Research*, 2025. 2

[37] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 5

[38] Volodymyr Shcherbyna, Linh Kastner, Diego Diaz, Huu Giang Nguyen, Maximilian Ho-Kyoung Schreff, Tim Seeger, Jonas Kreutz, Ahmed Martban, Zhengcheng Shen, Huajian Zeng, et al. Arena 4.0: A comprehensive ros2 development and benchmarking platform for human-centric navigation using generative-model-based environment generation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9138–9144. IEEE, 2025. 2

[39] Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, et al. Grutopia: Dream general robots in a city at scale. *arXiv preprint arXiv:2407.10943*, 2024. 5

[40] Liuyi Wang, Xinyuan Xia, Hui Zhao, Hanqing Wang, Tai Wang, Yilun Chen, Chengju Liu, Qijun Chen, and Jiangmiao Pang. 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*, pages 9455–9465, 2025. 5

[41] Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. Trackvla: Embodied visual tracking in the wild. *arXiv preprint arXiv:2505.23189*, 2025. 2

[42] Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. *arXiv preprint arXiv:2512.08186*, 2025. 2, 6, 8

[43] Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025. 2

[44] Jingda Wu, Yanxin Zhou, Haohan Yang, Zhiyu Huang, and Chen Lv. Human-guided reinforcement learning with sim-to-real transfer for autonomous navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12): 14745–14759, 2023. 2

[45] Wayne Wu, Honglin He, Chaoyuan Zhang, Jack He, Seth Z Zhao, Ran Gong, Quanyi Li, and Bolei Zhou. Towards autonomous micromobility through scalable urban simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27553–27563, 2025. 2

[46] Zhuo Xu, Hao-Tien Lewis Chiang, Zipeng Fu, Mithun George Jacob, Tingnan Zhang, Tsang-Wei Edward Lee, Wenhao Yu, Connor Schenck, David Rendleman, Dhruv Shah, et al. Mobility vla: Multimodal instruction navigation with long-context vlms and topological graphs. In *8th Annual Conference on Robot Learning*, 2024. 1, 2

[47] Zhuoyuan Yu, Yuxing Long, Zihan Yang, Chengyan Zeng, Hongwei Fan, Jiyao Zhang, and Hao Dong. Correctnav:

Self-correction flywheel empowers vision-language-action navigation model. *arXiv preprint arXiv:2508.10416*, 2025. 2

[48] Mingfeng Yuan, Letian Wang, and Steven L Waslander. Opennav: Open-world navigation with multimodal large language models. *arXiv preprint arXiv:2507.18033*, 2025. 2

[49] Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, and Xing Wei. Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. *arXiv preprint arXiv:2509.22548*, 2025. 2

[50] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024. 6

[51] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024. 2

[52] Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025. 2

[53] Siqi Zhang, Yanyuan Qiao, Qunbo Wang, Longteng Guo, Zhihua Wei, and Jing Liu. Flexvln: Flexible adaptation for diverse vision-and-language navigation tasks. *arXiv preprint arXiv:2503.13966*, 2025. 2

[54] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025. 1

[55] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7641–7649, 2024. 2

[56] Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025. 1

[57] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 3