UCLA Mobility Lab

ICML International Conference On Machine Learning

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

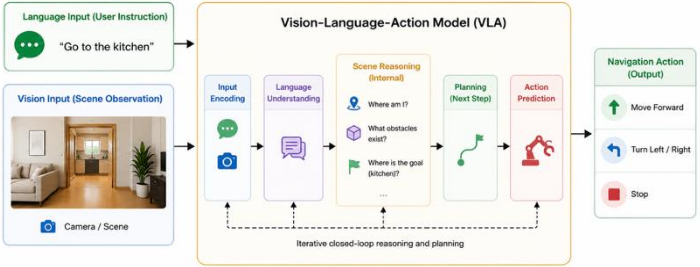
https://ucla-mobility.github.io/TIC-VLA/

Human-centric Robot Navigation

Interact with humans

Scene Understanding and Reasoning

Robot in Human-centric Environments



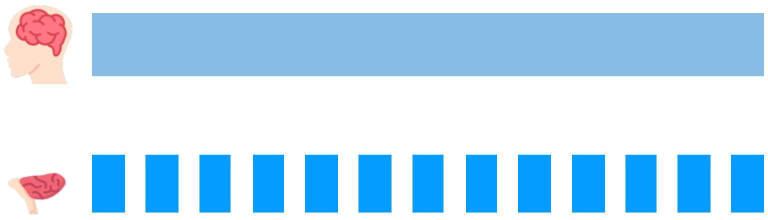
Vision-Lanauge-Action Model

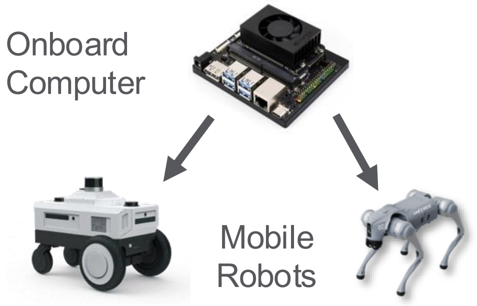
A Frequency Mismatch Between Reasoning and Action

- **Conflict:** Vision-Language Models (VLMs) provide rich semantic reasoning but incur significant latency; robot control requires fast, continuous feedback.
- **Latency Effect:** Inference delays introduce **semantic misalignment** between reasoning and the evolving environment.
- **Safety Risk:** On resource-constrained edge devices, latency is amplified, and robot control is compromised.

Think Slow

Act Fast





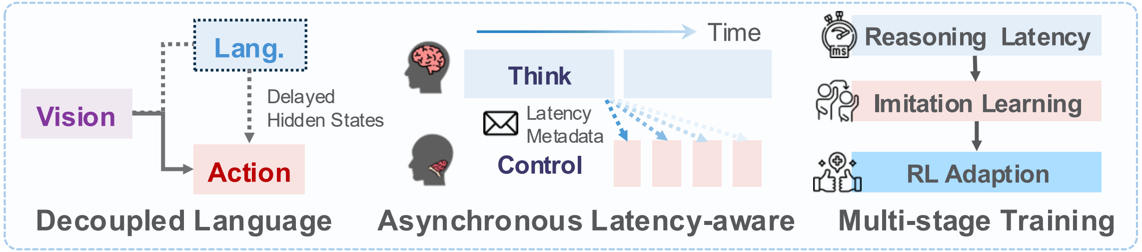
Our Goal: Maintain continuous, stable control while benefiting from deep semantic reasoning.

Think-in-Control (TIC)-VLA

- **Decoupled Language**
Vision-language reasoning separated from control; Uses delayed semantics

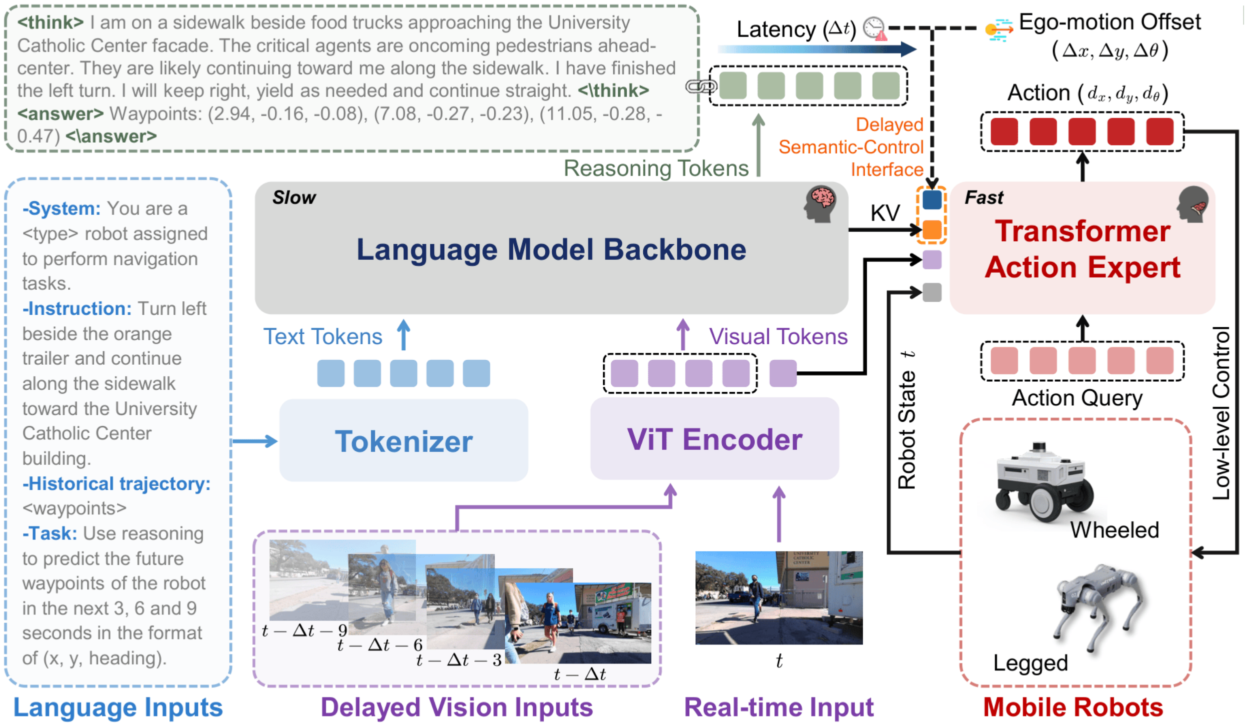
- **Latency-Aware Control**
Reasoning runs over time; Policy uses delayed features + latency/motion

- **Latency-Consistent Training**
Inject delay in training; Imitation to RL adaptation; Robust to latency



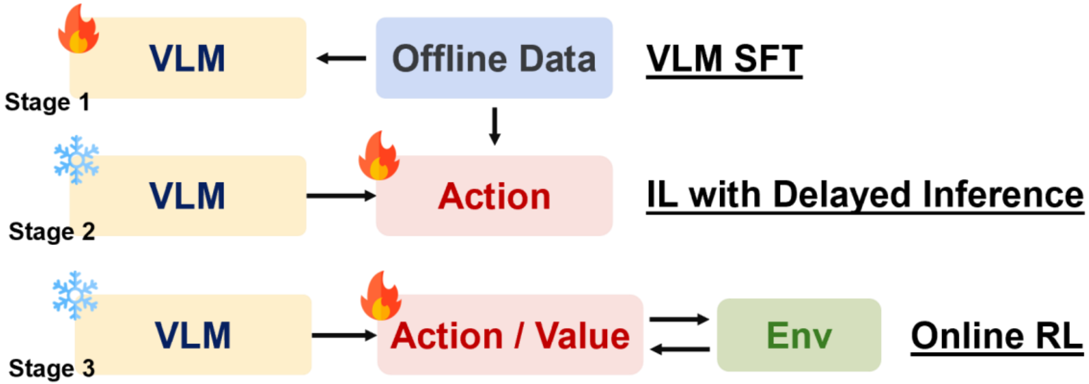
Our Solution: **Latency-aware dual-system design + latency-consistent training**

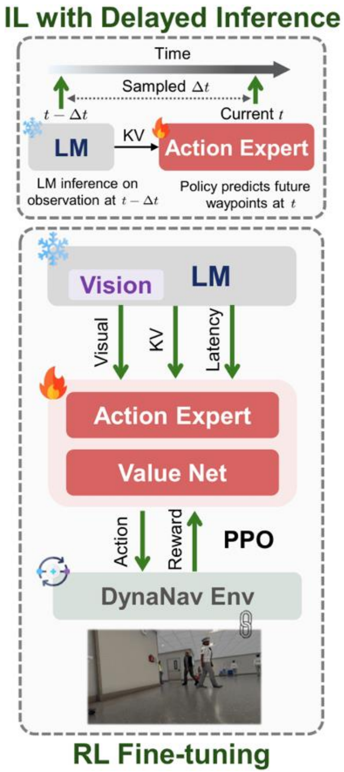
TIC-VLA Framework



- **Decouple reasoning and control**: Separate slow reasoning from fast low-level control.
- **Delayed semantic-control interface**: Action expert uses current observation plus delayed semantic with explicit latency awareness.
- **Asynchronous execution with caching**: Run VLM reasoning asynchronously and cache delayed semantic hidden states.

Latency-Consistent Training

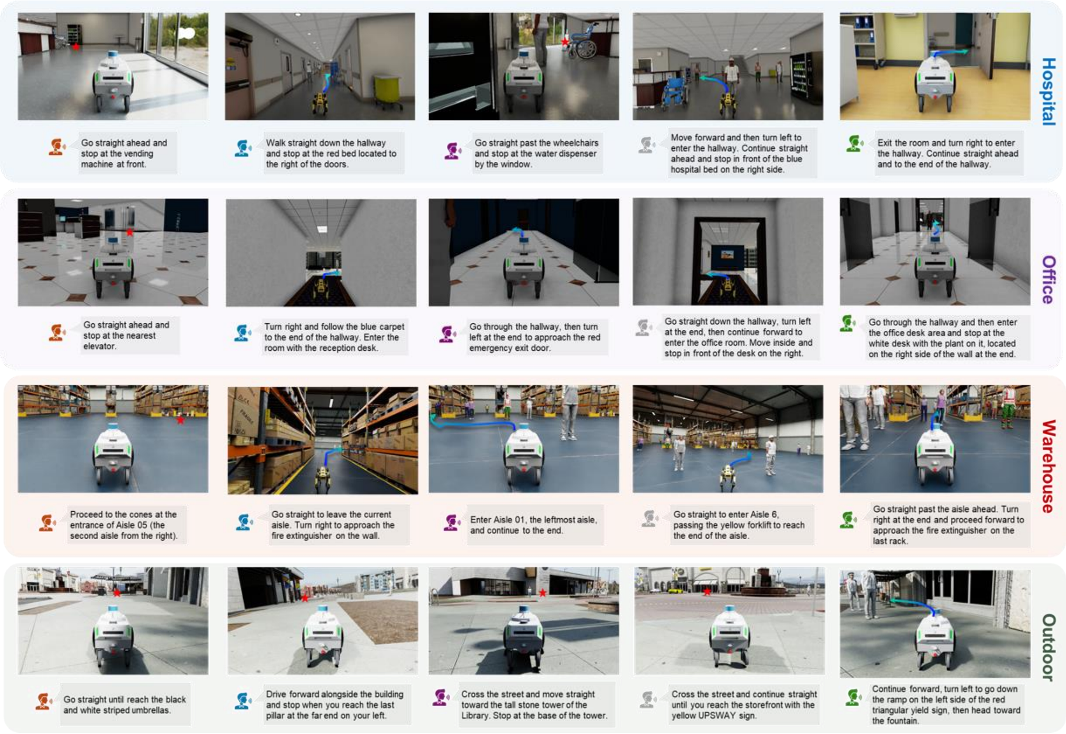




- **Stage 1:** Supervised fine-tuning of the VLM, distilling reasoning traces and knowledge from GPT-5 into a compact 1B-parameter model.
- **Stage 2:** Train the action expert with variable latency, using delayed VLM outputs and providing latency information as input.
- **Stage 3:** Apply reinforcement learning to make the action expert robust to realistic latency and distribution shifts.

UCLA

DynaNav: Realistic Dynamic Navigation Simulation

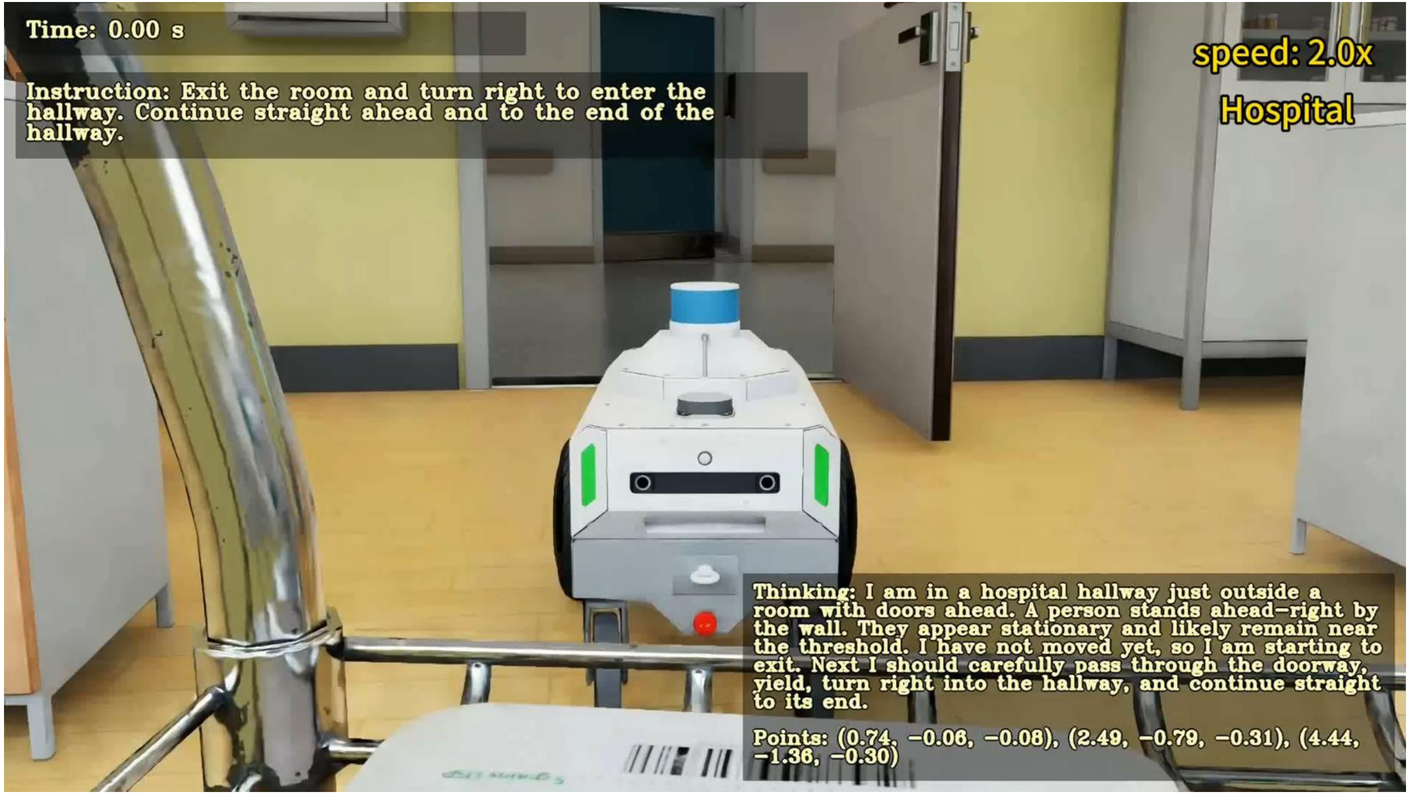


DynaNav Uniquely Provides:

- **Large-scale, photo-realistic, reproduceable environments** with physics-accurate robot control and time constraints.
- **Dynamic human participants** for realistic human-robot interaction.
- **Support for different robots and RL training**, enabling end-to-end evaluation.
- **Systematic benchmark design** spanning crowd density, navigation length, and scenes.

UCLA

Simulation Testing

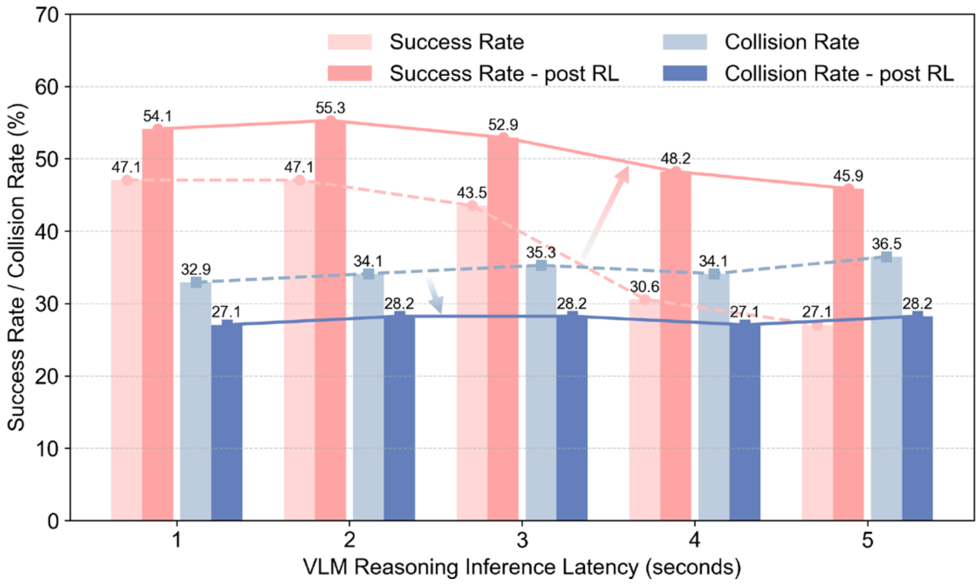


Robust Vision-Language Navigation Performance

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**

TIC-VLA achieves the best performance on the DynaNav benchmark and is comparable to goal-point-based navigation. **RL enhances performance and robustness to reasoning latency.**



Testing Results under Variable Latency

Real-World Testing

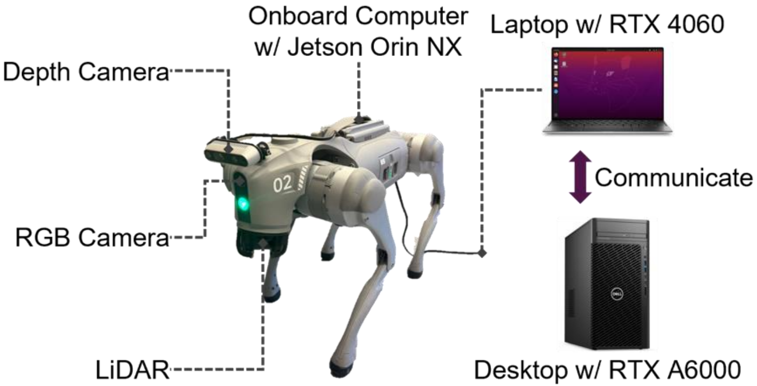
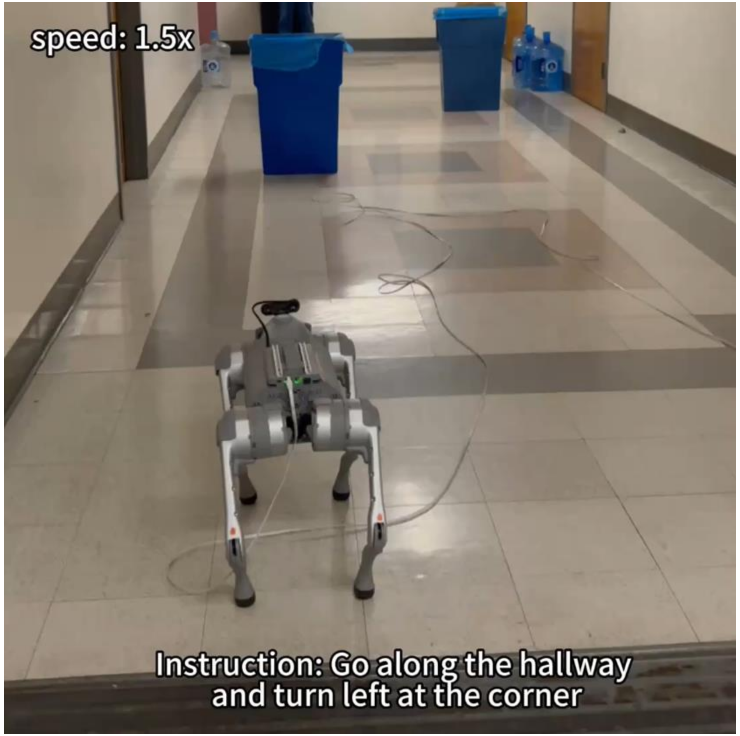


Table 3. Real-world testing results. Runtimes for the dual system are reported as (x/x) for the action policy and VLM reasoning.

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



Thank You