Computer Systems Engineering

Capstone Report - Fall 2025

Command-Controlled Robot Dog Using Vision-Language-Action Models

Mandy Liu (ml8305@nyu.edu)

Supervised by
Yik-Cheung Tam (yt2267@nyu.edu)

Preface

My academic journey at NYU Shanghai has evolved from theoretical quantum computing research and data science internships into a focused passion for the intersection of software and hardware. While I valued the rigor of developing machine learning models, I was drawn to a dynamic environment where algorithms could directly interact with physical components. This drive, alongside hands-on embedded systems projects, has led me to pursue a capstone project that synthesizes these experiences to advance robotic autonomy.

This project aims to integrate robust semantic reasoning within robotic systems to purposefully perform complex actions informed by their environment. Current literature suggests most robotic systems are optimized either for pure spatial awareness (e.g., mapping and localization) or for action-oriented behaviors (e.g., manipulating objects or executing simple fixed commands).

As such, the project focuses on developing a hierarchical architecture for quadrupedal robots that seamlessly integrates both spatial understanding and adaptive action comprehension. The system achieves this by implementing a Vision-Language-Action (VLA) pipeline that utilizes Vision-Language Models (VLMs) for high-level semantic reasoning, alongside real-time safety and precision provided by YOLOv8 and LiDAR sensors to understand its environment. This methodology allows the robot to achieve robust, socially aware navigation without requiring resource-intensive model fine-tuning.

Acknowledgements

This project wouldn't have been possible without the guidance and advice provided by Professor Yik-Cheung Tam. Thank you for all your help and support throughout the semester. In addition, I would like to thank Professor Prométhée Spathis for his feedback within my documentation and reports that went towards this paper. Lastly, I would like to acknowledge NYU Shanghai for the resources and space they have provided that allowed me to pursue this project.

Abstract

Although most robotic systems can carry out predefined actions like "turn left" or "greet" these behaviors are not autonomous and lack contextual awareness of its environment. As such, integrating Vision-Language Models (VLMs) into embodied robotic systems offers advanced semantic reasoning. However, due to the high inference latency and lack of real-time safety guarantees of VLMs, this project proposes using a hierarchical Vision-Language-Action (VLA) pipeline that utilizes YOLOv8 and Intel RealSense camera data to address this control trade-off.

By employing a dual-rate control methodology, the robot is conceptually partitioned in two parts: "slow-brain" and "fast-reflex." The "slow-brain" utilizes the Qwen3-VL VLM to perform semantic reasoning to determine an appropriate action based on training examples that are sent as part of its prompting. The "fast-brain" operates on a separate high-frequency thread using YOLOv8 and the Intel RealSense camera data to identify target subjects (e.g., person), execute real-time servoing, enforce collision avoidance, and override VLM commands in safety-critical situations. This command-controlled, hierarchical handoff mechanism effectively solves a piece of the latency-safety problem, demonstrating a scalable and deployable framework for VLM-guided robotic autonomy.

Given the limited time-frame (approx. 3 months), the project reduces the robot's capabilities to only greet humans and avoid obstacles. By focusing on a simplified set of capabilities, it makes training more feasible and allows for a proof of concept.

Keywords

Keywords: **Hierarchical Control; Vision-Language-Action; Vision-Language-Model; Few-Shot Learning; Dual-Rate Control**

Contents

1 Introduction

This project investigates the development of a robotic dog (Go2 Unitree Systems) capable of executing high-level semantic goals through the integration of a hierarchical Vision-Language-Action (VLA) Pipeline. A key challenge in this domain lies in bridging the gap between the VLM's powerful semantic reasoning and the robot's ability to perform real-time safe action within its physical environment. Current limitations of direct VLM integration include high inference latency and the absence of reliable depth awareness necessary for collision avoidance [1, 2]. At present, the robotic dog functions primarily through remote control or predefined action-based programming, underscoring the need for greater autonomy and adaptability through structured AI integration.

Guiding Research Questions

The guiding research questions of this project are as follows:

1. How can dual-rate control —splitting a robot into a "slow brain" for semantic reasoning and a "fast reflex" for real-time safety—– improve the reliability of VLM-guided robotic actions?
2. How can a hierarchical architecture bring together high-level reasoning and real-time spatial awareness to close the gap that usually exists in robotic design?
3. How can a simplified greeting-and-avoidance task demonstrate the effectiveness of combining VLM-based reasoning with fast, reflexive perception modules?

To address the control trade-off, the project implements a Hierarchical VLA Pipeline where the VLM performs high-level semantic reasoning (the "slow brain"), and the YOLOv8/RealSense sensors provide real-time tracking and safety (the "fast reflex"). This enables the robot to ground language-derived semantic goals in perception, making it possible to translate tasks, such as autonomously greeting a human and navigating obstacles, into immediate, safe, and actionable steps.

The objective of the project is to design and implement a wireless autonomous robotic system that integrates VLM semantic reasoning into a hierarchical control loop to achieve a higher level of socially aware human-robot interaction. The anticipated outcome is a robotic dog capable of situational adaptation and the execution of environment-specific tasks within a simplified scope (greeting humans and avoidance). For instance, when traversing an empty corridor, the system autonomously prioritizes exploration by identifying open paths. However, upon detecting a human, it dynamically shifts its behavioral state to 'social interaction.' It then utilizes real-time depth data to approach the subject safely and executes a "greet" gesture only when a specific proximity threshold (1.2m) is reached, thereby demonstrating the successful grounding of high-level semantic intent into precise physical action.

2 Related Work/ Literature Review

The literature review begins by examining the foundation of semantic robotics, Vision-Language Models (VLMs), which integrate LLM reasoning with visual perception to enable deep scene understanding for embodied AI. This capability evolved into Vision-Language-Action (VLA) models, the current state-of-the-art, which unify perception and control in an end-to-end network. Examples include NaviLA for navigation and LoHoVLA for manipulation [3, 4]. However, these VLA models are large, with high inference latency and a dependence on massive datasets. These characteristics create significant limitations for real-time safety and adaptability [1]. This motivates the need for the devolvement of a hierarchical VLA pipeline in autonomous robotics proposed in this work.

While these VLM and VLA architectures are compelling, the development of suitable evaluation metrics remains an open challenge. Existing robotic benchmarks, such as CALVIN and BARKOUR, offer relevant task evaluations but exhibit crucial gaps relative to this project. CALVIN primarily focuses on stationary or manipulative agents and lacks the complex locomotion component necessary for quadruped-based applications [5]. Conversely, BARKOUR evaluates spatial navigation and motor precision but does not incorporate high-level semantic or language-guided reasoning [6]. Therefore, to accurately assess the performance of the Dual-Rate Control system and the reliability of the VLM-YOLOv8 handoff, we developed a set of custom, task-specific metrics focused on the frequency of successful actions, including approaching humans, following humans, and greeting humans.

2.1 Vision Language Models (VLM)

Traditionally, computer vision models have relied on convolutional neural networks (CNNs) to identify and classify objects; however, these models are limited in that they cannot translate visual understanding into purposeful actions, require large labeled datasets, and often need retraining for even slight changes in the environment [1, 7]. In contrast, large language models (LLMs) enable machines to understand and generate text-based information, but they are restricted to processing language and lack the ability to perceive or reason about the physical world [2, 8]. With the emergence of multi-modal approaches, it has become possible to combine these two paradigms into Vision-Language Models (VLMs) that allow agents to recognize the physical world and interpret language describing the environment. Hence through VLMs, agents gain the capability to see images and understand natural language, enabling reasoning and linking visual input to text.

2.1.1 VLMs for Semantic Grounding

Recent advancements in Vision-Language Models (VLMs) have significantly influenced robotics, particularly in enabling high-level semantic understanding to guide low-level control. SpatialVLM and SpatialRGPT enhance traditional VLMs with explicit spatial reasoning capabilities, allowing models to perceive relative distances, directions, and object positions in the environment [9, 10]. This spatial awareness enables the generation of mid-level action goals that are directly compatible with hierarchical Vision-Language-Action (VLA) pipelines. Complementing this, Vision-Language Model Predictive Control (VLM-PC) leverages VLMs for commonsense reasoning and multi-step planning in legged robots, enabling adaptive navigation through dynamic, unstructured environments [11, 12]. As such, VLM-PC allows legged robots to not only interpret commands but also to anticipate and plan for environmental challenges, improving safety, reliability, and autonomy in real-world navigation tasks. By integrating high-level perception with structured, spatially grounded goals, these models serve as a bridge between semantic understanding and precise low-level motor execution, making them highly suitable for deployment on platforms such as the Unitree Go2.

2.2 Vision Language Action Models (VLA)

Building on the success of VLMs, recent research emphasizes hierarchical frameworks that translate high-level visual and language understanding into mid-level action goals, which are then executed by low-level controllers, enabling robots to interact effectively with the physical world.

While Vision-Language Models (VLMs) enable agents to perceive their environment and understand descriptive language, they do not directly translate this understanding into motor actions necessary for autonomous behavior [1]. Traditional robotic control systems, including those based on reinforcement learning, often rely on predefined action scripts, which can be inflexible, environment-specific, and difficult to generalize across tasks [1, 13]. Vision-Language Action (VLA) models address this limitation by integrating visual perception, language comprehension, and action generation within a single framework. By grounding natural language instructions in visual observations, VLAs enable robots to interpret commands such as “walk near the couch and greet the people sitting there” and convert them into executable motion sequences. This multimodal integration allows for adaptive, context-aware behaviors that bridge the gap between perception and action, offering a pathway toward more autonomous and flexible robotic systems [1, 3, 4].

2.2.1 NaVILA

NaVILA (Legged Robot Vision-Language-Action Model for Navigation) exemplifies a hierarchical approach for legged robots, in which high-level language and visual inputs are processed to perform generic navigation tasks [3]. Built on SpatialVLM and SpatialRGPT, researchers created fine-tuned VLM to output a mid-level action (VLA) in the language of “turn right 30 degrees” [9, 10]. The generated mid-level actions were then trained on Unitree Go2 locomotion policy to follow these instructions for execution. With a 17 percent improvement on classic VLN benchmarks using NaVILA, it enabled spatial understanding and adaptive movement of complex environments[3]. By bridging high-level reasoning with low-level control, NaVILA allows the Go2 robot to perform context-aware, language-driven navigation while maintaining balance, avoiding obstacles, and dynamically responding to environmental changes.

2.2.2 LoHoVLA

On the other hand, LoHoVLA represents a hierarchical approach for robotic systems that integrates high-level language and visual inputs to perform interactive tasks[4]. Built upon a large pretrained video and web Vision-Language Model (VLM), it combines high-level task planning with low-level motion control within a unified framework, enabling the robot to bridge semantic reasoning and motor execution[14]. The system first infers linguistic sub-tasks from input observations and specified goals, which are then used to guide the generation of mid-level actions (e.g, “stack all the blocks in the red zone”) that can be executable by a low-level manipulation policy. Given LoHoVLA outperformed Vanilla VLA on classification and matching tasks (e.g, put-block-into-matching-bowl task), there is empirical evidence that LoHoVLA could enable Go2 to accurately identify objects and execute the correct response[4]. Overall, LoHoVLA exemplifies how pretrained vision-language models can be effectively applied to robotic systems to unify high-level reasoning with real-time motor control.

2.3 Benchmarks

Existing benchmarks have played a crucial role in evaluating the capabilities of embodied AI systems across different modalities of perception, reasoning, and control. The CALVIN benchmark [5] provides a comprehensive framework for assessing continuous control and multi-step manipulation tasks. It emphasizes goal-conditioned generalization by testing agents on unseen combinations of spatial configurations and object arrangements, making it a strong standard for evaluating interactive manipulation and task decomposition in robotic systems. However, CALVIN primarily focuses on stationary or manipulative agents and lacks the locomotion component necessary for quadruped-based applications.

In contrast, the Barkour benchmark [6] is specifically designed for quadruped robots, providing standardized metrics for agility, balance, and traversal performance across varied terrains. Barkour evaluates spatial navigation and motor precision but does not incorporate high-level semantic or language-guided reasoning. While both benchmarks have significantly advanced evaluation standards in embodied AI, they remain limited in assessing interactive, language-conditioned tasks that require the integration of perception, reasoning, and dynamic control.

Table 1: Comparative Capabilities of Embodied AI Models and Benchmarks

Feature	**VLM**	**NaviLA**	**LoHoVLA**	**CALVIN**	**BARKOUR**
Semantic Reasoning	✓	✓	✓	×	×
Quadruped Locomotion	×	✓	×	×	✓
Real-Time Safety/Avoidance	×	∼	∼	×	∼
End-to-End Latency	Low	High	High	N/A	N/A

✓ = Yes, × = No, ∼ = Partially or Limited Scope, N/A = Not Applicable

3 Methodology

The development of the hierarchical VLA pipeline followed a phased approach, progressing from simulation-based SDK framework analysis, to isolated camera testing and object detection, and finally, full physical integration. To achieve the stated goals within the limited time frame, the Unitree SDK2 Python API (via the `unitree_sdk2` package) were used to control the motors rather than manually programming each individual joint. Specifically, the `SportClient` and `ObstaclesAvoidClient` modules were used to execute actions in the environment and perform collision avoidance using the robot's on-board LiDAR sensor. Similar to pressing a button on the remote control, the API calls by themselves are not considered autonomous behavior.

In order to achieve autonomous behavior, the Intel RealSense camera D435i was utilized to capture both color RGB frames and depth information. Through the camera's data, the information can be passed to either YOLOv8 or Qwen3-VL for inference. To ensure the robot prioritizes its goal of behaving like a social dog and greeting humans, YOLOv8-nano and YOLOv8-small were used for fast person detection. These detections override VLM inferences when necessary, allowing the robot to focus on navigating safely toward the person.

In the event no persons are detected within its view, the VLM will take control to make an inference and decide on a decision about its environment A training dataset consisting of image–action pairs (e.g., "wall: turn left") was collected prior to deployment, enabling the VLM to reference previously seen situations and infer appropriate actions when encountering new ones.

3.1 Simulation and Unitree SDK2 Python API Client Analysis

Initial development focused on using the MuJoCo physics engine to visualize Unitree Go2's kinematic behavior and understand the functionalities of the Unitree SDK2 Python API. The simulation environment was deployed on an Ubuntu Foxy/ WSL architecture that was connected to an Xbox controller to interact with the simulation interface. (Fig. 1).

However, due to discrepancies between MuJoCo SDK and Unitree SDK implementations, connecting high-level semantic commands to low-level motor control proved challenging Due to this mismatch, manually coding the robot's joints for basic actions such as walking forward would have been required, but this was outside the project's domain knowledge (Fig. 2). Before pivoting away from physics engines, NVIDIA's IsaacSim engine was also taken into consideration, but due to hardware limitations, this alternative was not pursued.

In the end, to simplify experimentation and validation of logic flow before deployment, the inferred action was printed to the terminal. Thus, the focus shifted towards developing the hierarchical architecture between YOLOv8 and Qwen3-VL's ability to identify objects and generate text-based action. Despite the limitations, the simulation provided critical insight into the robot's joint movement and kinematic constraints before physical deployment.

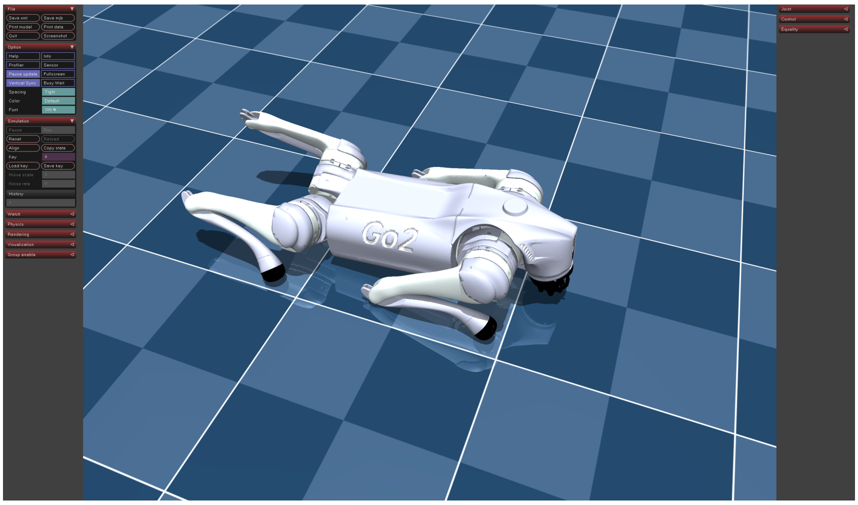

Figure 1: Unitree Go2 Robot in MuJoCo Simulator

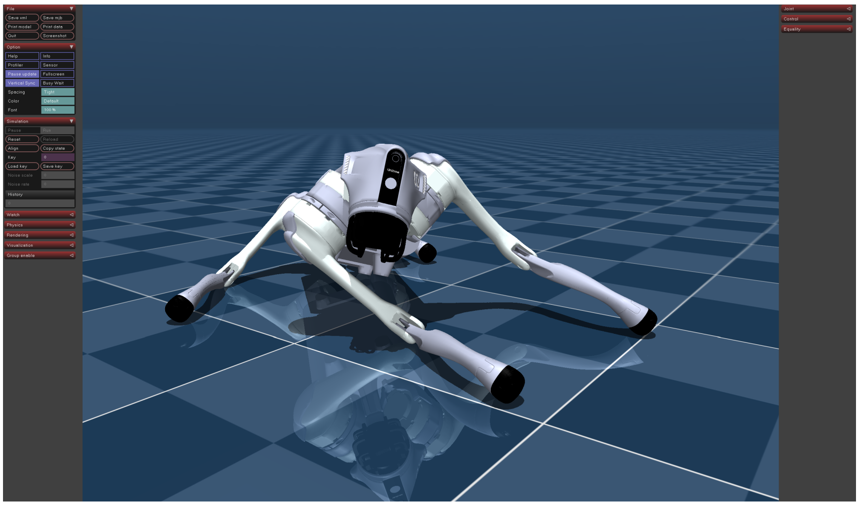

Figure 2: Difficulty with precise motor control in MuJoCo.
(Click the image to view the video)

3.2 MuJoCo Simulation and SDK Framework Analysis

Since the primary focus of the physics engine was on low-level motor control rather than high-level semantic directives, the validation environment was pivoted to a terminal-based abstraction. Through terminal-based abstraction, a safe environment was created for testing decision logic, exploring the capabilities of YOLOv8 and Qwen3-VL, and experimenting with the Intel RealSense camera. In this setup, actions associated with a perceived environment were printed to the terminal (e.g., “if ‘person’ detected, then print ‘sit’”) rather than executed by the motors. This approach allowed rapid prototyping of decision logic without risking hardware damage (Fig. 3).

Initially, we incorporated the COM15K depth-camera dataset from Hugging Face to evaluate Qwen3-VL’s inference capabilities. The COM15K dataset provides extensive information about distances, spatial structures, and obstacle proximity, making it a suitable starting point to assess how effectively Qwen3-VL could interpret images. Although Qwen3-VL could make strong inferences about object identity, it could only estimate object distances. Since the project aimed to incorporate real-time navigation, this limitation was addressed using YOLOv8 in conjunction with depth data from the Intel RealSense camera.

Using the Intel RealSense camera, the system verified the integration of YOLOv8 for real-time object detection and the Qwen3-VL API for semantic reasoning. YOLOv8 was selected for the “Fast Reflex” layer due to its high inference speed on embedded hardware and high mean Average Precision (mAP). The mAP metric validates the model’s ability to maintain high detection accuracy and precise bounding box localization across varying confidence thresholds, which is critical for dynamic object tracking [15]. Conversely, Qwen3-VL was chosen for the "Slow Brain" due to its robust zero-shot capabilities. "Zero-shot" refers to the model’s ability to recognize objects and understand complex scenes in new environments without requiring model retraining [16]. While we leverage this zero-shot perception, we later utilize few-shot prompting (providing small sets of image-action examples) to steer this understanding into specific, structured navigational commands.

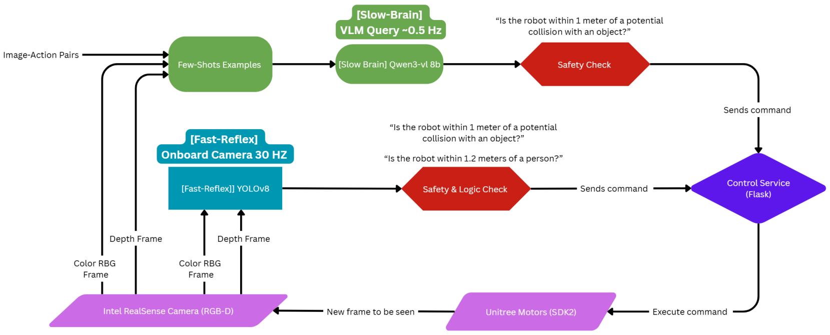

Figure 3: Hierarchical VLA Pipeline with Dual-Rate Strategy Diagram

To synthesize these perceptual inputs into autonomous behavior, we developed a hierarchical VLA pipeline that processes image streams into actionable commands. Drawing inspiration from the Voyager autonomous agent framework [17], which utilizes a trained curriculum to prioritize essential tasks, our system enforces a strict hierarchy of goals. The primary objective is defined as "find and greet humans," which supersedes all other environmental interactions. For example, if the system detects both a "potted plant" and a "human" in the same frame, the hierarchical logic dictates that the robot must ignore the plant and prioritize the "approach human" sub-task. If no humans are visible, the system defaults to a "search" behavior rather than idling, ensuring continuous progress toward the primary social goal.

3.3 Physical Deployment and Hierarchical Integration

The final phase involved deploying the Hierarchical VLA Pipeline onto the physical Unitree Go2, utilizing a Dual-Rate Control strategy to bridge the frequency gap between semantic reasoning and motor execution (Fig. 4). To enable autonomous perception, the Intel RealSense camera was integrated to capture synchronized RGB frames and depth information, which are passed to either the YOLOv8 or Qwen3-VL inference engines. The system is split into two distinct computational units:

1. **The Fast Reflex (Onboard):** The robot runs a local Flask-based Control Service. This service manages the SportClient for movement and the ObstaclesAvoidClient for safety. To ensure the robot prioritizes its social goal of greeting humans, YOLOv8-nano and YOLOv8-small are used for high-speed person detection (30Hz). These detections serve as a "reflex," overriding VLM inferences when a human is identified to allow the robot to focus on navigating safely toward the person.

2. **The Slow Brain (Server-Side):** In the event no persons are detected within the robot's view, the VLM (Qwen3-VL) takes control to infer environmental context and make navigational decisions. A training dataset consisting of image–action pairs (e.g., “wall: turn left”) was collected prior to deployment and provided to the VLM. This enables the VLM to reference previously seen situations and infer appropriate exploratory actions when encountering new environments.

This separation ensures that network latency from the VLM does not compromise the robot's real-time stability, while the VLM provides the high-level reasoning required for autonomous exploration.

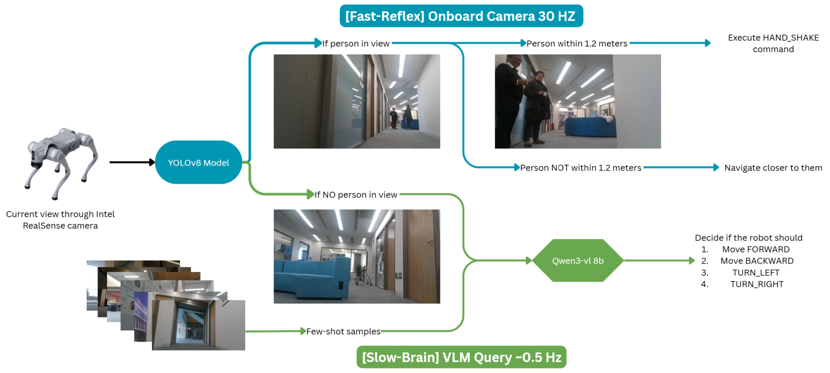
[Fast-Reflex] Onboard Camera 30 HZ
If person in view
Person within 1.2 meters
Execute HAND_SHAKE command
Person NOT within 1.2 meters
Navigate closer to them
YOLOv8 Model
Current view through Intel RealSense camera
If NO person in view
Qwen3-vl 8b
Decide if the robot should
1. Move FORWARD
2. Move BACKWARD
3. TURN_LEFT
4. TURN_RIGHT
Few-shot samples
[Slow-Brain] VLM Query ~0.5 Hz

Figure 4: Visual Pipeline Description

4 Results

The tracking performance of YOLOv8 nano and YOLOv8 small was evaluated across different static person placements (Center, Left, Right). YOLOv8 nano achieved a perfect success rate of 100% for the Center and Right placements, but a slightly lower rate of 90% for the left placement, suggesting a minor sensitivity to lateral positioning. Average latency for the nano model ranged from 304.21 ms to 356.62 ms, with collision rates remaining at 0% across all positions. The average distance maintained from the person varied between 847.3 mm and 984.3 mm, indicating that nano tended to track slightly closer than the intended greeting distance of 1.2m. VLM assistance was only employed for the left placement, coinciding with the slight drop in success rate, which may suggest that the model required additional semantic reasoning when the person was off-center (Fig. 2).

YOLOv8 small, in contrast, achieved a uniform 100% success rate across all placements, demonstrating improved robustness in tracking. Although it exhibited higher average latency ranging from 606.20 ms to 653.22 ms, the model maintained average distances ranging from 901.2 mm to 1009.4 mm, generally closer to the target greeting distance of 1.2 meters. In addition, the model had improved performance for off-center placements. Collision rates remained at 0%, and the selective use of VLM for the left placement mirrors the pattern seen with the nano variant, suggesting similar reliance on semantic reasoning for lateral tracking challenges. Overall, these results indicate a trade-off between model size, tracking accuracy, and inference speed, where the smaller model is faster but slightly less consistent in off-center placements, while the larger model is robust but slower (Fig. 3).

Navigation capabilities of Qwen3-VL-8b were evaluated in a controlled indoor environment with varying numbers of hallways traversed. Latency per run ranged from 244.50 ms to 620.30 ms, with collision rates between 10% and 23.3%. Stagnation occurred in all tested runs, indicating that the model struggled to maintain continuous progress in the absence of human intervention. The variation in latency and collision rate across different hallway traversals suggests that navigation complexity increases with environmental variability, and that the system may be sensitive to action space limitations. These findings highlight the balance between processing latency and robust navigation performance in indoor environments (Fig. 4).

4.1 Experimentation protocol

The goal of experimentation is to systematically evaluate the performance of the proposed hierarchical VLA pipeline across two primary components: the fast-reflex perception module (YOLOv8) and the slow-brain semantic reasoning module (Qwen3-VL). The measurements are designed to quantify responsiveness, safety under controlled conditions, and task completion. A dataset of over 100 image-action pairs were captured prior to experimentation such that sample examples can be passed to Qwen3-VL.

4.1.1 Environment Setup

1. **Environment:** A controlled indoor environment measuring 5 m × 5 m, with clearly marked regions (Left, Center, Right) relative to the robot's starting position, where a human subject can be placed. The space is also connected to multiple hallways.

2. **Robot Configuration:** The Unitree Go2 is equipped with an Intel RealSense D435i camera, LiDAR, and servo mechanisms, in addition to running the hierarchical VLA pipeline.

3. **Repetitions:** Each test configuration is repeated $N = 10$ times to ensure statistical reliability and account for network latency variance.

4.1.2 Evaluation Metrics

Performance is quantified using four key metrics:

1. **System Latency:** Time elapsed between visual input and command generation

2. **Collision Rate:** Frequency of physical contact with obstacles

3. **Navigation Stagnation:** Instances where the robot entered a conflicting command loop or "deadlock"

4. **Task Success Rate:** Successful approach to within $1.2m$ followed by a greeting and turning away action

4.1.3 Individual Component Analysis Procedures

To validate the dual-rate architecture, the system layers were evaluated in isolation:

1. **Fast Reflex Evaluation (YOLOv8):** The tracking accuracy was tested by moving a human subject through the three spatial zones (Left, Center, Right). We measured the system's ability to re-center the target using real-time visual servoing without incurring collisions. The models used for evaluation were YOLOv8 nano and YOLOv8 small.

2. **Slow Brain Reasoning (Qwen3-VL):** The semantic reasoning capabilities were assessed by the robot's ability to explore its environment safely, avoiding collisions and preventing navigation stagnation. This evaluation focused on the balance between inference latency and decision-making accuracy, specifically observing whether the robot navigated effectively without repeating movement patterns (e.g., moving forward, then backward, then forward again). Notably, the robot could only perform 30 discrete actions before being reset to its starting position. The model used for this assessment was Qwen3-VL-8b.

Table 2: Capabilities of YOLOv8 nano to Track Static Person

Person Placement	**Success Rate (%)**	**Avg Latency (ms)**	**Collision Rate (%)**	**Avg Greeting Dist (mm)**	**VLM Used?**
Center	100%	304.63	0%	984.3	×
Left	90%	356.62	0%	865.8	✓
Right	100%	304.21	0%	847.3	×

✓ = Yes, × = No

Table 3: Capabilities of YOLOv8 small to Track Static Person

Person Placement	**Success Rate (%)**	**Avg Latency (ms)**	**Collision Rate (%)**	**Avg Greeting Dist (mm)**	**VLM Used?**
Center	100%	622.58	0%	922.3	×
Left	100%	653.22	0%	1009.4	✓
Right	100%	606.20	0%	901.2	×

✓ = Yes, × = No

Table 4: Capabilities of Qwen3-VL-8b to Navigate

# Hallways Traversed	Latency per Run (ms)	Collision Rate (%)	Stagnation Occur
1	412.00	10%	✓
0	244.50	23.3%	✓
2	620.30	13.3%	✓

✓ = Yes, × = No

5 Discussion

The primary challenges encountered during this project centered around navigation using Vision-Language Models (VLMs), strict time constraints, and the workload limitations inherent to a single-person effort. Most existing Vision-Language-Action (VLA) models rely on large datasets to achieve robust performance [3, 4]. Collecting and annotating such datasets was infeasible within the three-month timeframe, which necessitated developing a solution that could operate effectively with limited data. Our approach leveraged pre-trained YOLOv8 models combined with selective VLM assistance, enabling the robot to maintain accurate tracking and approximate the desired greeting distance without requiring extensive retraining or massive data collection efforts. This represents an advantage over traditional VLA pipelines that depend heavily on large-scale dataset availability, making our method more practical for rapid prototyping and small-scale deployment.

An important observation from our experiments is the complementary potential of YOLO and the VLM for navigation. Currently, the VLM primarily guides the robot only when no person is detected, resulting in navigation that can be somewhat aimless when trying to interact with other objects in the environment. However, if the VLM could cross-reference objects identified by YOLO, the system could lock onto specific targets and navigate toward them with purpose, rather than moving blindly. This coordinated approach would combine YOLO's fast and reliable object detection with the VLM's semantic reasoning, potentially improving goal-directed navigation and reducing stagnation or unnecessary movements.

Despite these strengths, certain limitations remain. Navigation with the VLM exhibited higher latency and occasional stagnation, particularly in complex hallway configurations or off-center placements of targets. The discrete nature of the robot's movement, constrained by inference and action delays, further limits smooth continuous navigation. While parameter tuning—such as adjusting inference intervals or movement step sizes—could help reduce latency and improve responsiveness, the solution still falls short of fully continuous and fluid navigation. In comparison, some existing VLA models achieve smoother control and higher success rates in complex environments due to end-to-end fine-tuning on large datasets, highlighting an area where our approach could be further improved.

Given the experience gained from this project, several potential directions emerge for future exploration. One avenue involves developing a more integrated pipeline where YOLO and the VLM work together to identify objects and navigate toward them with purpose. Lightweight fine-tuning and simulation-based pretraining could also enhance navigation strategies without requiring massive data collection. Additionally, hybrid approaches combining fast object tracking with selective semantic reasoning may strike a practical balance between speed, robustness, and goal-directed behavior. Overall, the project demonstrates that careful model selection and task-specific adaptations can enable effective navigation and human-robot interaction even under significant resource and time constraints.

6 Conclusion

This project successfully demonstrated a hierarchical Vision-Language-Action (VLA) pipeline for a robotic dog that integrates YOLOv8-based perception with VLM semantic reasoning to achieve autonomous and socially aware behaviors. YOLOv8 small provided robust tracking across different person placements and maintained distances closer to the target greeting range of 1.2 meters. YOLOv8 nano offered faster inference but was slightly less consistent in off-center positions.

VLM guidance allowed goal-directed navigation when no person was detected, but the current operation is limited to discrete movement and occasional stagnation in complex environments. The combination of YOLO and VLM shows promise. YOLO ensures real-time obstacle avoidance and object tracking, while the VLM enables high-level semantic reasoning. With further integration, the system could focus on specific objects and navigate purposefully rather than moving blindly.

Overall, the project demonstrates that semantic goals can be successfully grounded into safe and actionable robotic behaviors within practical time and data constraints. Future work could involve tighter integration of YOLO and VLM for object-directed navigation, continuous movement refinement, and simulation-based pretraining to further reduce latency and improve robustness.

References

[1] R. Sapkota, Y. Cao, K. I. Roumeliotis, and M. Karkee, "Vision-language-action models: Concepts, progress, applications and challenges," 2025. [Online]. Available: https://arxiv.org/abs/2505.04769

[2] K. Kawaharazuka, J. Oh, J. Yamada, I. Posner, and Y. Zhu, "Vision-language-action models for robotics: A review towards real-world applications," *IEEE Access*, vol. 13, p. 162467–162504, 2025. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2025.3609980

[3] A.-C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang, "Navila: Legged robot vision-language-action model for navigation," 2025. [Online]. Available: https://arxiv.org/abs/2412.04453

[4] Y. Yang, J. Sun, S. Kou, Y. Wang, and Z. Deng, "Lohovla: A unified vision-language-action model for long-horizon embodied tasks," 2025. [Online]. Available: https://arxiv.org/abs/2506.00411

[5] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," 2022. [Online]. Available: https://arxiv.org/abs/2112.03227

[6] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, N. Batchelor, S. Bohez, F. Casarini, J. E. Chen, O. Cortes, E. Coumans, A. Dostmohamed, G. Dulac-Arnold, A. Escontrela, E. Frey, R. Hafner, D. Jain, B. Jyenis, Y. Kuang, E. Lee, L. Luu, O. Nachum, K. Oslund, J. Powell, D. Reyes, F. Romano, F. Sadeghi, R. Sloat, B. Tabanpour, D. Zheng, M. Neunert, R. Hadsell, N. Heess, F. Nori, J. Seto, C. Parada, V. Sindhwani, V. Vanhoucke, and J. Tan, "Barkour: Benchmarking animal-level agility with quadruped robots," 2023. [Online]. Available: https://arxiv.org/abs/2305.14654

[7] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, no. 4, p. 99, 2024. [Online]. Available: https://doi.org/10.1007/s10462-024-10721-6

[8] F. Stella, C. D. Santina, and J. Hughes, "How can llms transform the robotic design process?" *Nature Machine Intelligence*, vol. 5, pp. 561–564, 2023. [Online]. Available: https://doi.org/10.1038/s42256-023-00669-7

[9] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Driess, P. Florence, D. Sadigh, L. Guibas, and F. Xia, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," 2024. [Online]. Available: https://arxiv.org/abs/2401.12168

[10] A.-C. Cheng, H. Yin, Y. Fu, Q. Guo, R. Yang, J. Kautz, X. Wang, and S. Liu, "Spatialrgpt: Grounded spatial reasoning in vision language models," 2024. [Online]. Available: https://arxiv.org/abs/2406.01584

[11] A. S. Chen, A. M. Lessing, A. Tang, G. Chada, L. Smith, S. Levine, and C. Finn, "Commonsense reasoning for legged robot adaptation with vision-language models," 2024. [Online]. Available: https://arxiv.org/abs/2407.02666

[12] W. Zhao, J. Chen, Z. Meng, D. Mao, R. Song, and W. Zhang, "Vlmpc: Vision-language model predictive control for robotic manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2407.09829

[13] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, T. Wang, S. Marks, C.-R. Segerie, M. Carroll, A. Peng, P. Christoffersen, M. Damani, S. Slocum, U. Anwar, A. Siththaranjan, M. Nadeau, E. J. Michaud, J. Pfau, D. Krasheninnikov, X. Chen, L. Langosco, P. Hase, E. Bıyık, A. Dragan, D. Krueger, D. Sadigh, and D. Hadfield-Menell, "Open problems and fundamental limitations of reinforcement learning from human feedback," 2023. [Online]. Available: https://arxiv.org/abs/2307.15217

[14] C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, H. Zhang, and M. Zhu, "Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2410.06158

[15] A.-R. A. Gamani, I. Arhin, and A. K. Asamoah, "Performance evaluation of yolov8 model configurations, for instance segmentation of strawberry fruit development stages in an open field environment," 2024. [Online]. Available: https://arxiv.org/abs/2408.05661

[16] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu, "Qwen3-vl technical report," 2025. [Online]. Available: https://arxiv.org/abs/2511.21631

[17] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar, "Minedojo: Building open-ended embodied agents with internet-scale knowledge," 2022. [Online]. Available: https://arxiv.org/abs/2206.08853