OpenVLA-Driven High-Level Navigation for Industrial AGVs

A Novel Approach for Semantic Scene Understanding & Control

Team3

November 20, 2025

Motivation

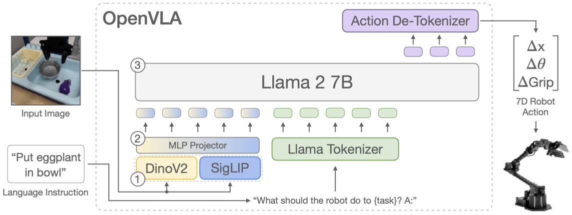
Current Challenges

- **Semantic Blindness**: Traditional AGVs see geometry, not meaning. They cannot distinguish a crouching worker from a pallet.
- **The "Freezing Robot" Problem**: Treats all moving agents as static obstacles, leading to abrupt, panic-like stops.
- **Lack of Negotiation**: Fails to nudge aside or signal intent at intersections.

Proposed Solution

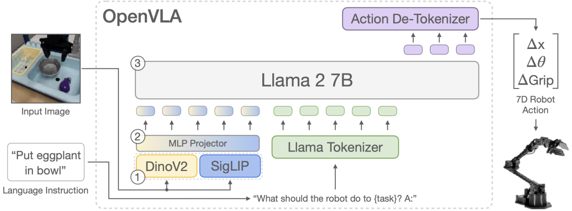
- Integrate the **OpenVLA (Vision-Language-Action)** model.
- Leverage strong visual reasoning as the AGV's "High-Level Brain".
- Empower robot to understand scenes and issue strategic commands.

Gap Analysis: Adapting OpenVLA for Mobility



The Potential vs. The Mismatch

- **Strength:** State-of-the-art generalist policy trained on massive robot datasets.
- **Limitation:** Native output is a **7-DoF Action** (joint angles, gripper) designed for manipulation.
- **Incompatibility:** AGVs require 2D planar navigation (v_x, ω) or discrete decisions, not 7D poses.



Our Adaptation Strategy

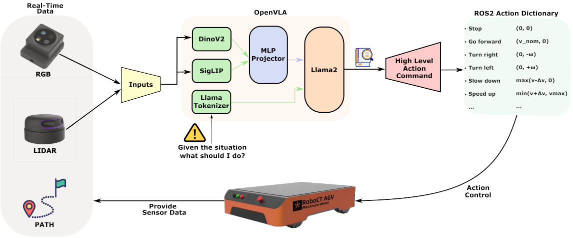
- **Remove:** The *Action De-Tokenizer* (Action Head).
- **Inject:** A text-based **System Prompt** to redirect output to the LLM's language head.
- **Result:** High-level semantic tokens (e.g., "DETOUR") instead of motor noise.

System Architecture Overview

Hybrid Navigation Architecture

Combining the *Modified VLA* with existing industrial controllers:

- **Perception Input:** Front RGB Image + LiDAR BEV.
- **Decision Core:** Modified OpenVLA (Text Output).
- **Mapping Layer:** Translates text tokens to specific control protocols with action dictionary.
- **Control Stack: Proprietary AGV Controller** (e.g., Navitrol / PLC) handling the kinematics.



Action Dictionary & Mapping

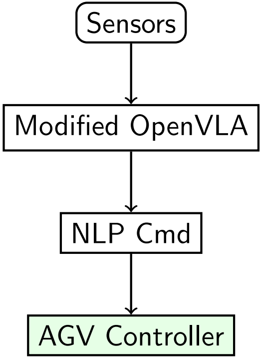
Command	Signal	Scenario
STOP	`vel = 0` / `pause_task`	Unavoidable obstacles / Blocked.
GO	`vel = nominal`	Path clear, resume operation.
YIELD	`vel = 0` (Temp)	Yielding to crossing pedestrians.
DETOUR	`offset_path` / `bias`	Partial blockage, needs lateral shift.

**Note:* *Signals are abstract representations; actual implementation depends on the hardware interface (see Slide 9).*

Control Flow & Safety

Software-Level Compliance

- We utilize the AGV's native controller as the safety enforcement layer.
- **VLA Layer (Strategy):** Determines *Intent* (e.g., "Pass on the left").
- **Native Controller (Execution):** Handles *Kinematics* (e.g., wheel velocity profiles).



Simulation Setup (Isaac Sim)

Simulating the "Black Box" Controller

- Since we do not use Nav2, we implement a **Kinematic Control Node** in Isaac Sim to mimic the real AGV's behavior.
- **Input:** High-level velocity or discrete state commands.
- **Output:** Differential/Omni drive wheel velocities.
- **Goal:** Validate VLA decision logic without relying on complex open-source planners.

Data Collection Workflow

Synthetic Data Generation

- **Domain Randomization:** Auto-generate dynamic obstacles (pedestrians, forklifts).
- **BEV Generation:** Scripts to project 3D point clouds to 2D images.
- **Goal:** Establish "Golden Dataset" (Normal, Crossing, Blocked).

Real-World Integration Strategy

Challenge: The physical AGV runs on a proprietary stack (Navitrol). Bridging OpenVLA (ROS2-based) requires investigation.

Option A: API Bridge
(Preferred)
Use Navitrol's API .

+ Safe & Standardized.
+ Handles kinematics.

Option B: Industrial Proto
(Fallback)
Control via **OPC UA**.

+ Universally supported.
- High latency (bad for reactivity).

Option C: Direct PLC
(Last Resort)
Send analog/digital signals to PLC.

+ Real-time control.
- Re-implementing safety logic is risky.

Phase 1 Action Item: *Establish communication handshake with on-site Navitrol unit.*

Implementation Plan

Phase 1: Integration & Handshake

Build OpenVLA Inference Node; **Investigate AGV API (Navitrol)**; Basic Stop/Go logic.

Phase 2: Simulation Validation

Use Isaac Sim to verify VLA logic against a "Mock Controller" (simulating real AGV response).

Phase 3: Deployment

Deploy on edge compute; Fine-tune latency; Expand Action Dictionary.

Conclusion

Training-Free
No expensive training; direct visual reasoning.

Hardware Agnostic
Decoupled from Nav2; adaptable to Navitrol or PLC.

Future Outlook
Foundation for true Social Navigation.

Q & A

Thank you for listening