machines

MDPI

Article

A Unified Conditional Policy for Multi-Robot Navigation via LiDAR-to-Vision Distillation

Amir Mahdi Amani [1,*,†], Sajjad Amani [2] and AmirHossein MajidiRad [3]

[1] Software Development Division, Venloer Str. 310-316, 50823 Cologne, Germany
[2] Robotics Division, Seaos Co., Ltd., 1-18-18 Ebisu, Tokyo 150-0013, Japan; s-amani@seaos.co.jp
[3] Department of Engineering Technology, Purdue University Northwest, 2200 169th St., Hammond, IN 46323, USA; amajidir@pnw.edu
* Correspondence: amir.amani@deutsche-netzplanung.de or amirmahdi.amani7th@gmail.com
† Website: https://am-amani.github.io/ (accessed on 28 May 2026).

Abstract

Mobile-robot navigation policies have typically assumed a fixed sensing input and robot platform. In this work, we investigate a teacher–student policy where teachers learn continuous velocity commands from LiDAR-based Twin Delayed Deep Deterministic Policy Gradient, and the student navigates using either LiDAR or camera observations. The student has two modalities with separate feature extractors. The extracted features and robot-state variables are fed into a shared encoder and actor, which map them to linear and angular velocity commands. The generalized two-platform configuration includes a Robot-ID token of scalar type. Training and evaluation were performed in ROS-Gazebo using simulated Pioneer 3-DX and TurtleBot3 Waffle robots. Robot-ID conditioning increased generalized LiDAR success to 90.4% on Pioneer 3-DX and to 91.2% on TurtleBot3 Waffle, up from 71.8% and 79.5%, and camera-based success was 87.3% and 84.0%, respectively. Across four controlled LiDAR-to-vision handover conditions, we achieved an overall success rate of 88.8-90.8%. Overall, 85.2-90.1% of the episodes that were still active at the planned switch were completed without resetting or retraining the policy. The results demonstrate that the conditioning on robot identity is very helpful for cross-platform LiDAR performance and that the shared student policy is able to continue navigation after external scheduling of the active sensing branch. The results provide a controlled, simulation-based feasibility assessment of the proposed policy architecture and establish the basis for subsequent physical robot validation.

Keywords: mobile robots; robot learning; reinforcement learning; autonomous driving; machine learning

Check for updates

Academic Editors: Rui Azevedo Antunes, Luís Brito Palma and Antonios Gasteratos

Received: 29 May 2026
Revised: 22 July 2026
Accepted: 3 August 2026
Published: 13 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Mobile robots used in logistics, inspection, and service tasks require a local control policy that can move toward a goal while avoiding nearby obstacles. LiDAR supplies geometric range information, whereas a camera provides scene appearance cues. The two sensing modalities, therefore, offer different evidence for navigation and fail under different conditions [1,2]. Can a learned control policy operate from either LiDAR or camera input without requiring both streams simultaneously? This question is more specific than broad multimodal perception.

1.1. Background

LiDAR and cameras provide complementary but imperfect information for navigation. Mechanical scanning components can degrade, and weather such as fog, rain, snow, or dust can affect range returns [3–5]. Cameras retain visual scene information but are sensitive to illumination and appearance changes, and monocular images do not provide direct range measurements [2,6]. Transfer between robot platforms introduces a separate challenge because sensor placement, geometry, and control response alter the mapping from an observation to a safe action [7–9]. The present work therefore evaluates a constrained setting: two simulated differential-drive platforms, one shared student policy, and one active sensing modality at a time.

This study evaluates a teacher–student navigation framework in which a single-student policy can operate using either LiDAR or visual observations. Separate LiDAR-based TD3 teachers are trained for the Pioneer 3-DX and TurtleBot3 Waffle platforms in ROS-Gazebo (Figure 1). Their actions supervise a student network with modality-specific feature extractors, a shared encoder, a shared actor, and robot-state inputs. During generalized training, a scalar Robot-ID token is provided with LiDAR samples to distinguish the two platform-specific trajectory sources. The policy is evaluated only on these two simulated platforms; no zero-shot transfer to unseen robots or physical deployment is claimed.

The experiments examine policy imitation, cross-platform conditioning, and LiDAR-to-vision handover in simulated environments with standard lighting and stationary obstacles during each episode. Fog, rain, snow, dust, abrupt illumination changes, and moving agents are not modeled. These conditions motivate the use of alternative sensing modalities, but robustness to such disturbances is not evaluated in the present study.

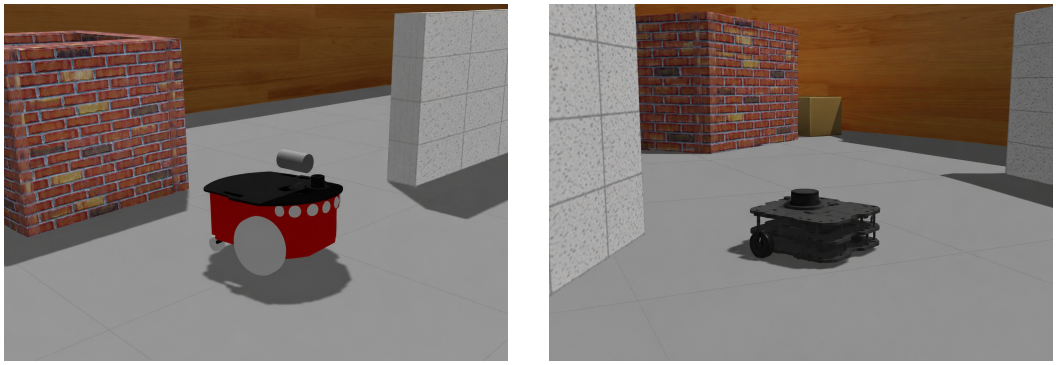


Figure 1. Robot models used in the ROS-Gazebo simulations: Pioneer 3-DX (**left**) and TurtleBot3 Waffle (**right**).

For the single-platform models, the LiDAR student retained performance close to that of its corresponding teacher. On Pioneer 3-DX, the student reached the goal in 91.6% of 1000 episodes, compared with 94.1% for the teacher. On TurtleBot3 Waffle, the student obtained 92.2%, compared with 91.1% for the teacher. The differences were therefore -2.5 and $+1.1$ percentage points, respectively. These results show that supervised action imitation preserved the teacher navigation behavior under the tested LiDAR conditions, although they do not establish statistical equivalence or superiority.

The cross-platform results show a modality-dependent effect. In the generalized model without an identifier, LiDAR success declines on both platforms, whereas vision success remains closer to the single-robot results. Introducing a scalar Robot-ID token restores LiDAR success from 71.8% to 90.4% on Pioneer 3-DX and from 79.5% to 91.2% on TurtleBot3 Waffle. These results support the practical value of explicit platform conditioning for the two evaluated robot configurations; they do not establish that visual observations intrinsically identify every robot embodiment.

The proposed architecture avoids simultaneous LiDAR-vision fusion during execution: the student acts from one selected modality at a time. Controlled evaluation changes the active input from LiDAR to vision within an episode while retaining the same student policy. This protocol demonstrates continuation after an externally scheduled handover; it does not include automatic sensor-health monitoring or automatic failure detection.

Reliable mobile navigation is also relevant to future assistive and rehabilitation-oriented service robots that could operate in rehabilitation clinics, hospitals, long-term care facilities, and home-care settings. Such environments may involve clutter, changing illumination, sensor degradation, and people sharing the same space. Prior work has considered socially assistive robots in post-stroke rehabilitation and mobile robotic assistants in nursing homes [10,11]. The present study does not implement a rehabilitation, healthcare, or patient-interaction system; it contributes only a simulation-based navigation infrastructure that could support future care-oriented mobile robots after human-aware, safety, and physical-system validation.

The contribution of this work is methodological rather than a new navigation theory. It evaluates action-level distillation from platform-specific LiDAR teachers to a shared student that can accept either LiDAR or camera observations. The associated experiments include single-platform baselines, a Robot-ID conditioning ablation, and controlled LiDAR-to-vision handovers. These analyses identify which design choices are supported by the present ROS-Gazebo evidence; they do not claim transfer to arbitrary robot embodiments or deployment-ready resilience to real sensor faults.

For cross-platform navigation, the generalized student receives a scalar Robot-ID token in addition to the modality-specific features, encoded context, and robot state. The results compare the generalized model with and without this conditioning input across the two studied robots. Together with the controlled handover tests, these experiments provide targeted evidence for the design choices under the present simulation protocol.

Section 1.2 places the study within work on navigation perception, continuous-control reinforcement learning, and learned policy transfer. Section 2 describes the teachers, replay-buffer records, student architecture, and action-imitation objective. Section 3 reports the single-student, generalized-student, Robot-ID, and controlled-handover evaluations. Section 4 interprets these results together with the study limitations, and Section 5 summarizes the contribution and the remaining validation steps.

1.2. Related Work

The literature most relevant to this study concerns sensor-specific navigation, multimodal perception, and learned policy transfer. The discussion below focuses on the aspects that motivate a student policy that can execute from a selected modality across two known robot configurations.

Multimodal Perception and Modality Flexibility: LiDAR and RGB cameras are commonly combined through explicit fusion, in which synchronized sensor streams are merged before navigation or obstacle-avoidance decisions are produced [12–14]. Other work uses sensor dropout or robust fusion objectives to limit the effect of missing measurements [15,16]. Those methods address a different operational choice from the one examined here. The present student does not combine LiDAR and vision features at runtime; it receives one selected sensor modality and maps that observation to a shared velocity-command space.

Continuous Control with TD3 in Robotics: TD3 addresses value overestimation in continuous-control actor-critic learning through twin critics, delayed policy updates, and target-policy smoothing [17]. It has been applied to mapless mobile-robot navigation, including settings that use range observations to produce linear and angular velocity

commands [18,19]. In this study, TD3 is not proposed as a new algorithmic contribution. Its role is to produce platform-specific LiDAR teacher policies whose recorded actions provide the labels for student imitation.

Teacher-Student Transfer and Cross-Embodiment Generalization: Teacher–student methods transfer an expert policy or privileged observation into a student through action imitation, auxiliary losses, or related distillation objectives [20–22]. This work uses the simplest of these mechanisms: supervised regression from the selected student observation to the teacher velocity command recorded at the same time step. Transfer across robots remains difficult because kinematics, dimensions, and sensor placement can change the observation-action relationship [7]. Domain randomization and online or meta-adaptation are common responses to that gap [23,24], while broader multi-robot navigation policies have also been reported [8,9,25]. The present scope is more limited: platform-conditioned imitation across Pioneer 3-DX and TurtleBot3 Waffle, with LiDAR or camera execution rather than universal embodiment transfer.

Navigation Focused State of the Art Comparison

Table 1 narrows the comparison to representative studies that explicitly address mobile-robot navigation or collision avoidance. The selected studies cover mapless LiDAR navigation, vision-only navigation, LiDAR-camera fusion, and visual multi-robot navigation. This is a scope comparison, not a shared-benchmark performance ranking: the studies differ in robots, tasks, environments, sensors, datasets, and evaluation protocols. "Not reported" means that the cited study does not report that capability; it does not imply that the approach could not be extended in future work.

The table identifies the targeted gap studied here. Existing navigation systems commonly operate from a fixed LiDAR input, a fixed visual input, or concurrent multimodal sensing. In contrast, the present work evaluates a student that uses LiDAR or vision independently at execution while preserving a shared continuous action space. The contribution is therefore a focused design and evaluation for selectable-modality navigation across two known robot platforms, rather than a claim of universal superiority or a direct numerical comparison against the cited studies.

Table 1 is a scope comparison rather than a shared-benchmark ranking. It places the present study alongside fixed-modality and fused-modality navigation systems, as well as visual multi-robot policies. The specific contribution assessed here is a student policy with selectable LiDAR or camera execution and scalar platform conditioning for two simulated robots. Automatic sensor-failure detection, sensor fusion, and zero-shot transfer to unseen embodiments remain outside the evaluated scope.

Table 1. State-of-the-art scope comparison with navigation studies. This table is not a shared-benchmark numerical comparison.

Study	Runtime Sensing	Navigation Learning Formulation	Multi-Platform Evaluation	LiDAR-Vision Handover Evaluated	Relation to the Present Study
Pfeiffer et al. [26]	2D LiDAR and relative goal	Expert-demonstration pre-training, imitation learning, and reinforcement learning for mapless navigation	Not reported	Not reported	Shares action learning from expert trajectories, but operates from LiDAR at runtime and does not study visual fallback.
Chiang et al. [27]	Noisy LiDAR observations	End-to-end reinforcement-learning navigation with automated reward and architecture selection	Not reported	Not reported	LiDAR-only continuous control; it does not address cross-modal execution.
Devo et al. [28]	Visual observation and target information	Deep reinforcement learning for target-driven visual navigation	Not reported	Not reported	Vision-only navigation; it does not use LiDAR teacher actions or a LiDAR-to-vision handover.
Han et al. [29]	2D LiDAR and monocular camera	Deep reinforcement learning with self-state attention and sensor fusion for collision avoidance	Not reported	Not reported	Uses both sensing streams jointly at runtime, whereas the proposed policy executes from one selected modality.
Shah et al. (GNM) [9]	Visual observations and goal-conditioned context	Multi-robot visual navigation model trained from heterogeneous robot trajectories	Yes	Not reported	Relevant visual multi-robot reference; it does not evaluate LiDAR-to-vision fallback or use LiDAR teacher supervision.
Shah et al. (ViNT) [30]	Visual observations and goal representation	Visual-navigation foundation model trained on navigation data from multiple robot platforms	Yes	Not reported	Relevant visual multi-platform reference; it does not include a LiDAR execution path or a sensor handover protocol.
Proposed framework	**LiDAR or grayscale vision as one selected active modality**	**LiDAR-TD3 teacher actions distilled into one student policy with Robot-ID conditioning**	**Yes: Pioneer 3-DX and TurtleBot3 Waffle**	**Yes: externally scheduled LiDAR-to-vision continuation; no automatic failure detector**	**Evaluated against the internal LiDAR TD3 teachers and student variants under the same Gazebo protocol.**

2. Methodology

The methodology is reported in the order in which the implemented pipeline produces and uses data: LiDAR teachers generate actions in simulation, replay buffers store paired sensor observations and teacher commands, and the student policy learns to reproduce those commands from one active modality at a time. This organization replaces the earlier report-like description with the specific experimental workflow used in this study.

2.1. Overview of the Proposed Approach

This section defines the proposed teacher–student framework as a modality-flexible action-imitation problem. Separate LiDAR-based TD3 teachers generate navigation actions for their respective robot platforms. A single student then learns to reproduce the same continuous action from either LiDAR or camera observations, while using the robot state and, in the generalized setting, a scalar platform-conditioning token. The framework therefore shares a control objective across sensing modalities without requiring sensor-level fusion.

The method is not presented as a complete autonomy stack. It does not include localization, global planning, automatic sensor-health diagnosis, or an online decision module for detecting when a sensor has failed. The contribution is the learned local control component and its evaluation under controlled single-modality and handover protocols.

- **Platform-specific teacher policies**: Separate TD3 policies are trained with LiDAR observations for the Pioneer 3-DX and TurtleBot3 Waffle platforms. They share the same learning architecture but use reward designs tailored to the corresponding navigation setup.
- **Modality-flexible student policy**: Independent LiDAR and camera feature extractors feed a shared encoder and actor, which predict the same linear and angular velocity command from the selected sensing modality.
- **Cross-platform conditioning**: A scalar robot token is appended to the actor input in the generalized baseline, enabling the shared policy to distinguish the two known robot embodiments.

Figure 2 provides an overview of the teacher data collection and student training pipeline. The following subsections specify the teacher policies, stored training tuples, modality-specific perception branches, and common action-prediction objective.

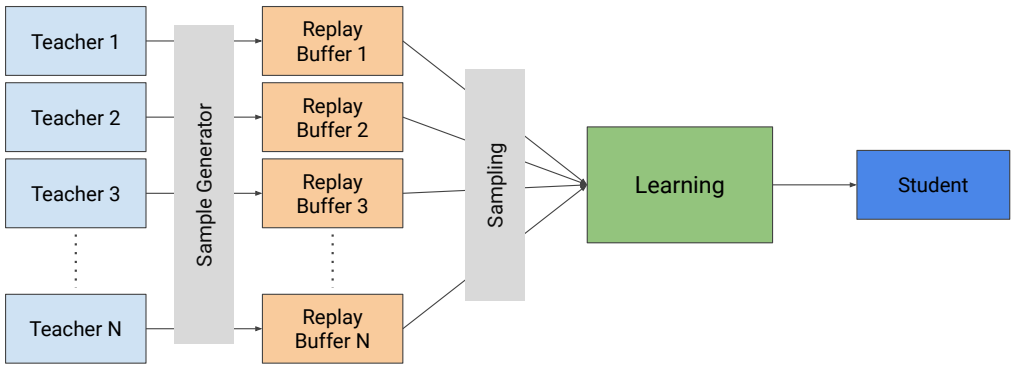


Figure 2. Diagram of the teacher–student framework used in this study.

2.1.1. Teacher–Student Framework

Two platform-specific teacher policies are trained with LiDAR observations using the TD3 algorithm. The Pioneer 3-DX and TurtleBot3 Waffle teachers use the same TD3 architecture but different reward designs because their navigation settings and platform dynamics differ. The Pioneer reward encourages forward velocity while penalizing turning

and unsafe obstacle proximity. The Waffle reward incentivizes non-increasing goal distance and forward velocity while penalizing movement away from the goal, turning, and unsafe obstacle proximity. Thus, the two teachers provide platform-specific action demonstrations; they are not treated as two interchangeable behavior labels supplied to the student. During training and data collection, obstacle locations are randomized at the beginning of each episode.

In the remainder of the paper, "teacher" refers to a fixed LiDAR policy used to label replay-buffer transitions. The teacher is not used during student execution, and the student is not trained to reproduce the teacher's LiDAR features. The supervised target is only the recorded velocity command associated with the synchronized transition.

Each teacher trajectory is stored in a separate platform-associated replay buffer. At each logged control step, the record contains the teacher action, robot state, LiDAR observation, and paired grayscale camera observation. The student can receive either LiDAR or camera data as its active perceptual input, but the supervision target remains the LiDAR-teacher action recorded for that same control step. This is action-level knowledge distillation: the method transfers navigation behavior through a common control space rather than attempting to match LiDAR measurements directly to image pixels.

2.1.2. Conditional Policy and Action-Level Distillation

For robot platform j and modality m, the generalized policy uses a scalar conditioning token $\chi_{m,j}$. In the standard Robot-ID configuration, $\chi_{m,j} = \rho_j$ for both LiDAR and vision. In the asymmetric conditioning evaluation, $\chi_{\mathrm{L},j} = \rho_j$ for LiDAR and $\chi_{\mathrm{V},j} = 0$ for vision. The shared actor receives this token, the direct modality features, the encoded context, and the robot state, and outputs a continuous navigation action $\hat{\mathbf{a}}_t^{m,j} = (\hat{v}_t, \hat{\omega}_t)$. This mapping is formalized in Equation (1).

$$\hat{\mathbf{a}}_t^{m,j} = \pi_\theta\big([\chi_{m,j}, \mathbf{f}_t^m, \mathbf{z}_t^m, \mathbf{r}_t]\big), \qquad \hat{\mathbf{a}}_t^{m,j} \in \mathbb{R}^2. \tag{1}$$

The actor input has 42 elements: one scalar conditioning token, 32 modality-specific features, five shared context features, and four robot-state values. For every stored transition, the teacher action $\mathbf{a}_t^T = (v_t^T, \omega_t^T)$ is the supervision target. The student is optimized for the currently selected modality by minimizing the mean-squared action error, as shown in Equation (2):

$$\mathcal{L}_m(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{\mathbf{a}}_i^{m,j} - \mathbf{a}_i^T \right\|_2^2. \tag{2}$$

For the generalized student, the learning objective can be written across the two modalities and the two platform-specific trajectory buffers. Let $\mathcal{M} = \{\mathrm{L}, \mathrm{V}\}$ denote LiDAR and vision, and let $\mathcal{J} = \{\mathrm{P}, \mathrm{W}\}$ denote Pioneer 3-DX and TurtleBot3 Waffle. The aggregate objective over a complete alternating-modality training cycle is given in Equation (3):

$$\mathcal{L}_{\text{joint}} = \frac{1}{|\mathcal{M}||\mathcal{J}|} \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} \frac{1}{N_{m,j}} \sum_{i=1}^{N_{m,j}} \left\| \pi_\theta\Big(\Big[\chi_{m,j}, F_m(\mathbf{o}_{i,j}^m), E\Big(F_m(\mathbf{o}_{i,j}^m)\Big), \mathbf{r}_{i,j}\Big]\Big) - \mathbf{a}_{i,j}^T \right\|_2^2. \tag{3}$$

This objective formalizes the implemented training design. Modality-specific front ends map LiDAR or vision observations into a shared control pipeline, while balanced sampling from the Pioneer and Waffle replay buffers prevents either platform from dominating the generalized objective. In implementation, camera and LiDAR updates alternate rather than being optimized simultaneously within one gradient step.

The LiDAR and camera CNNs are distinct front ends and therefore preserve modality-specific early processing. Their outputs are passed through the same encoder and actor so that both modalities are trained toward the same continuous navigation action.

The implemented optimization schedule activates one modality at a time. On each even-numbered training step, only the grayscale camera batch is processed by the camera CNN; the camera CNN, shared encoder, and shared actor are updated using the mean-squared error between the predicted action and the stored teacher action. On the following odd-numbered step, only the 20-sector LiDAR batch is processed by the LiDAR CNN; the LiDAR CNN, the same shared encoder, and the same shared actor are updated against the corresponding teacher action target. The inactive modality-specific CNN is not updated during that step. Thus, LiDAR and camera tensors, and their extracted feature vectors, are never concatenated or supplied simultaneously to the actor.

This design provides separate early feature extraction and prevents direct input-level sensor fusion. Alternating updates also give both modalities regular access to the shared encoder and actor, rather than allowing one modality to dominate their optimization. However, because the encoder and actor are intentionally shared, their parameters receive sequential gradients from both modalities. The schedule therefore does not mathematically guarantee complete feature decoupling or eliminate all possible interference within the shared layers. No contrastive, adversarial, orthogonality, or modality-disentanglement loss is used, and no latent-feature correlation analysis is reported. The appropriate claim is consequently modality-specific front ends with balanced sequential optimization toward one shared navigation-action objective.

2.1.3. Advantages of the Proposed Approach

Modality-flexible execution. The student can execute from either a LiDAR or camera observation without requiring simultaneous sensor fusion. The controlled handover evaluation changes the active perception branch during an episode, but the present experiments do not inject mechanical failures, lighting changes, rain, fog, dust, snow, or other environmental sensor degradations.

Platform-conditioned evaluation. The generalized student is evaluated on two known robot configurations. The scalar Robot-ID token allows the shared actor to condition its output on the platform associated with each trajectory. Extending the method to an additional robot would require teacher data, a conditioning choice, and a separate evaluation; the current results do not quantify that effort.

No simultaneous sensor fusion. At execution, the actor receives features from one active modality. This removes the need to synchronize and concatenate LiDAR and camera inputs in this implementation. It does not guarantee independence from all sensor-related errors, nor does it replace a safety analysis for a physical robot.

The architecture reduces the need for a simultaneous fusion pipeline in the controlled ROS-Gazebo setting. Its environmental robustness beyond the tested static-obstacle and standard-lighting conditions remains to be evaluated.

2.2. Training the Teacher Models

Platform-specific teacher policies are trained with TD3 to generate continuous linear- and angular-velocity demonstrations. TD3 is used here as the demonstration generator for the student training dataset, not as a new reinforcement learning contribution.

All training and evaluation are carried out using ROS 1 and Gazebo 11.15.1, with Python 3.8.10. A Python environment class interfaces between the neural network and the simulator, and the active robot model is either Pioneer 3-DX or TurtleBot3 Waffle. Each reset randomizes the robot start pose, goal location, and locations of four cardboard-box

obstacles. The obstacle locations therefore vary between episodes, but the obstacles remain fixed within an episode. The simulator configuration does not model fog, rain, dust, snow, abrupt illumination changes, or moving obstacles. Figure 3 provides an overview of this controlled environment.

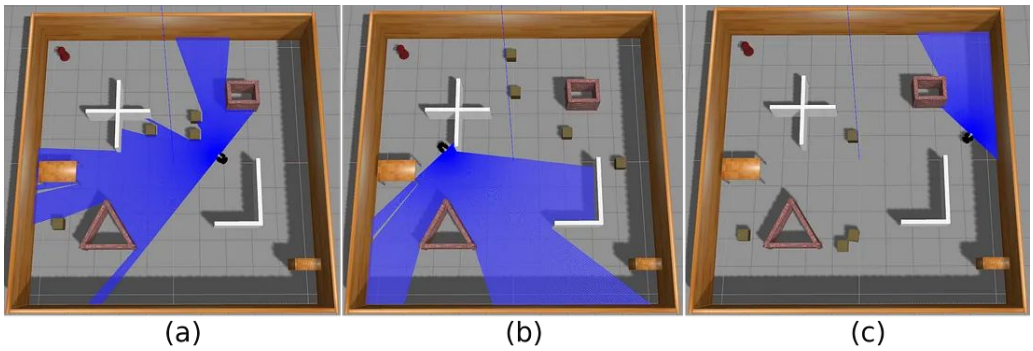


Figure 3. ROS-Gazebo simulation environment for the Pioneer 3-DX and TurtleBot3 Waffle. (**a**–**c**) show three randomized obstacle configurations during environment resets. Obstacles remain static within each episode.

Table 2 summarizes the environmental scope of the reported simulation experiments and the perturbations that are not included in the present evaluation.

Table 2. Environmental scope of the reported ROS-Gazebo experiments.

Scenario Component	Status and Interpretation
Robot start pose, goal position, and four obstacle locations	Randomized at every episode reset, yielding multiple stationary navigation layouts.
Obstacle motion	Not included; obstacles remain fixed within an episode. No dynamic-obstacle evaluation is claimed.
Fog, rain, dust, and snow	Not modeled. No adverse-weather robustness is claimed.
Illumination or camera-exposure changes	Not modeled; standard simulator lighting is used. No illumination-robustness evaluation is claimed.
Sensor failure	No naturally occurring failure injection; the LiDAR-to-vision change is externally scheduled in the handover test. This demonstrates controlled continuation, not automatic failure detection.

2.2.1. State Representation and Action Space

The teacher state combines a 20-element LiDAR summary, the relative position of the goal, and the preceding control command, as shown in Equation (4):

$$s_t = (l_t, g_t, a_{t-1}), \tag{4}$$

Here, $l_t \in \mathbb{R}^{20}$ is the sectorized LiDAR representation, $g_t = (d_t, \theta_t)$ contains goal distance and relative heading, and $a_{t-1} = (v_{t-1}, \omega_{t-1})$ records the previous linear and angular velocity. The policy action is $a_t = (v_t, \omega_t)$. This compact state supplies the teacher with obstacle-range information and goal-related variables required for continuous local control.

2.2.2. Training Process

A teacher episode ends when the robot reaches the goal, collides, or reaches the configured time limit. Figure 4 shows the actor-critic structure used during TD3 training. The actor maps the current state to a continuous pair of linear and angular velocity commands. Its hidden layers use ReLU activations and its output layer uses `Tanh`, which is subsequently mapped to the command range used by the simulator [17].

The archived final teacher actors use the 24-dimensional state described above (20 LiDAR sectors plus four robot-state values), followed by fully connected layers of 800 and 600 units and a two-dimensional action output. The associated critic checkpoints contain the two TD3 critic branches used to evaluate state-action values during teacher optimization.

To improve training stability, TD3 employs two critic networks instead of one. Each critic estimates the Q-value of a state-action pair, and the minimum of the two estimates is used during target computation to mitigate overestimation bias. In addition, the architecture maintains target versions of both the actor and critic networks, which are updated through soft updates to avoid abrupt parameter changes during optimization. During training, noise is added to the target actions for policy smoothing, and the actor network is updated less frequently than the critics according to the delayed-policy-update principle. Together, the twin critics, target policy smoothing, and soft target updates make TD3 well suited for learning robust teacher policies that can later provide high-quality demonstrations for the student model [17].

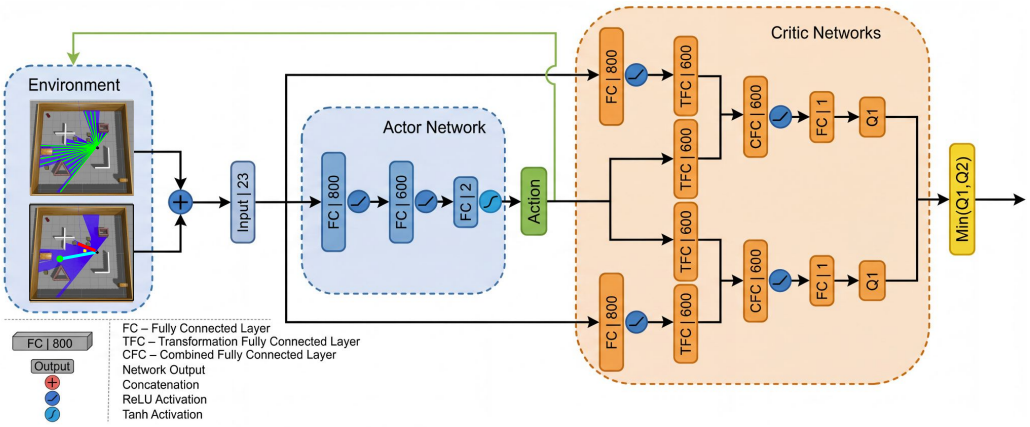


Figure 4. Structure of the TD3 network used for training the teacher models. The actor outputs actions (linear and angular velocities), and two critic networks evaluate the action quality.

2.2.3. Reward Function Design

The two platform-specific teacher policies use distinct reward designs:

- **Pioneer 3-DX teacher**: Encourages forward linear velocity and penalizes turning magnitude and obstacle proximity.
- **TurtleBot3 Waffle teacher**: Incentivizes non-increasing goal distance and forward linear velocity, while penalizing movement away from the goal, turning magnitude, and obstacle proximity.

The implemented reward functions are stated explicitly here. Let $d_t^{\min}$ denote the minimum LiDAR range at time t, and define the obstacle-proximity term $p_t = \max(0, 1 - d_t^{\min})$. For both teachers, reaching the goal yields $+100$, and collision yields -100; these terminal rewards override the non-terminal expression. For the Pioneer teacher, the non-terminal reward is given in Equation (5):

$$r_t^{\mathrm{P}} = \frac{1}{2} v_t - \frac{1}{2} |\omega_t| - \frac{1}{2} p_t. \tag{5}$$

For the Waffle teacher, let d_t be the current goal distance and $\bar{d}_{t-1}$ the previous goal distance used for reward evaluation. Its non-terminal reward is given in Equation (6):

$$r_t^{\mathrm{W}} = b_t + 0.75 v_t - \frac{1}{2}|\omega_t| - \frac{1}{2} p_t, \tag{6}$$

The distance-shaping term b_t is defined in Equation (7):

$$b_t = \begin{cases} 0.3 \exp[-0.1(d_t - 8)], & d_t \le \bar{d}_{t-1}, \\ -0.3 \exp[0.1(d_t - 2)], & d_t > \bar{d}_{t-1}. \end{cases} \tag{7}$$

At reset, $\bar{d}_{t-1}$ is unset. On the first step it is initialized to the current distance before reward evaluation; after each step, it is updated to the current distance for the next evaluation. Consequently, equality $d_t = \bar{d}_{t-1}$ is deliberately handled by the non-increasing-distance branch. All non-terminal Waffle numerical constants are active: $A = 0.3$, $\alpha = 0.1$, the distance offsets 8 and 2, the linear-velocity coefficient 0.75, and the obstacle-proximity and turning penalty coefficients 0.5. The goal orientation θ_t remains part of the state representation but is not an explicit Waffle reward term.

Table 3 summarizes the terminal and non-terminal reward terms used by the two teacher policies.

Table 3. Teacher reward terms and implemented coefficients. Terminal goal and collision rewards override all non-terminal terms.

Teacher	Terminal Rewards	Non-Terminal Terms
Pioneer 3-DX	$+100$ goal; -100 collision	$+0.5 v_t$; $-0.5|\omega_t|$; $-0.5 p_t$
TurtleBot3 Waffle	$+100$ goal; -100 collision	$+0.75 v_t$; $-0.5|\omega_t|$; $-0.5 p_t$; distance-shaping term b_t (Equation (7))

The final saved teacher policies were evaluated over 1000 ROS-Gazebo episodes using the corresponding platform configuration. The Pioneer teacher achieved 941 successes (94.1%), and the Waffle teacher achieved 911 successes (91.1%). These final-policy results are reported again in Table 4 to make the quality of the action-label generators explicit.

Table 4. Final evaluation of the saved LiDAR TD3 teacher policies.

Teacher Policy	Evaluation Episodes	Successes	Success Rate
Pioneer 3-DX LiDAR TD3 teacher	1000	941	94.1%
TurtleBot3 Waffle LiDAR TD3 teacher	1000	911	91.1%

2.3. Data Collection

Each teacher is trained separately and records its own trajectories. The implemented replay-buffer entry at time step t is given in Equation (8):

$$e_t = (x_t, u_t, r_t, \tau_t, x_{t+1}, c_t, \ell_t), \tag{8}$$

where x_t and x_{t+1} are the current and next TD3 environment states, $u_t = (v_t, \omega_t)$ is the teacher action, r_t is the TD3 reward, τ_t is the terminal flag, c_t is the stored grayscale camera frame, and ℓ_t is the stored laser-scan observation. For action imitation, the student uses the associated robot state $q_t = (d_t, \theta_t, v_{t-1}, \omega_{t-1})$, one selected sensor observation, and u_t as its supervision target. The RL-only fields r_t, τ_t, and x_{t+1} are retained in the buffer but are not prediction targets for the student.

This record format is also the reason the camera and LiDAR supervision can be described without generic sensor-fusion language. Both observations are stored with the same robot state and teacher command, but only one observation branch is active in a given student update or evaluation step.

Teacher provenance is preserved by buffer membership rather than by an additional behavior label inside every entry: $\mathcal{B}_{\mathrm{P}}$ stores Pioneer-teacher trajectories, and $\mathcal{B}_{\mathrm{W}}$ stores Waffle-teacher trajectories. The two teachers are platform-specific action generators, not two alternative behavior classes for one robot. During generalized-student training, each update draws equal-sized subsets from $\mathcal{B}_{\mathrm{P}}$ and $\mathcal{B}_{\mathrm{W}}$, concatenates them into one batch, and attaches the corresponding scalar Robot-ID token only in the configurations that use Robot-ID conditioning.

For an active modality $m \in \{\mathrm{LiDAR}, \mathrm{Vision}\}$, the action-imitation objective is defined in Equation (9):

$$\mathcal{L}_m = \frac{1}{N} \sum_{i=1}^{N} \left\| \pi_\phi(o_i^m, q_i, \rho_i) - u_i \right\|_2^2, \tag{9}$$

where o_i^m is the selected LiDAR or visual observation and ρ_i is the scalar Robot-ID when that conditioning is enabled. Camera and LiDAR updates are alternated across optimization steps; therefore, the training schedule optimizes the balanced expected objective $\frac{1}{2}\mathcal{L}_{\mathrm{LiDAR}} + \frac{1}{2}\mathcal{L}_{\mathrm{Vision}}$, with each modality loss computed over a batch that contains equal numbers of Pioneer and Waffle samples.

The supervision is therefore action-level distillation rather than sensor-level translation. The LiDAR teacher action u_t is not used as a LiDAR reconstruction target and the camera branch is not trained to reproduce ℓ_t. Instead, ℓ_t is privileged information used by the TD3 teacher to choose a navigation command during data generation, while the paired camera frame c_t and robot state q_t are used as the student input for the vision update. Because the teacher and student share the same differential-drive action space, $u_t = (v_t, \omega_t)$ is a valid modality-independent target: both the LiDAR and vision branches are trained to predict the same continuous velocity command for the same recorded control step.

This formulation does not assume that visual observations contain the same information as LiDAR or that a grayscale image can recover an exact range scan. It assumes only that, for many navigation states in the simulated environment, the image and robot-state variables contain enough cues for the student to approximate the teacher's control decision. The lower vision success rates compared with LiDAR in the results are therefore interpreted as an expected consequence of incomplete visual depth information and closed-loop imitation error, not as evidence of perfect cross-modal equivalence.

2.4. Training the Student Model

Student training is evaluated in two stages. The single-robot stage establishes LiDAR and camera execution for each platform separately. The generalized stage uses data from both platform-associated replay buffers and evaluates the effect of the Robot-ID token.

The student objective is kept deliberately simple: predict the teacher's continuous action from the selected student observation and the robot-state variables. The network is therefore evaluated as an action-imitation policy, not as a sensor reconstruction model, semantic perception system, or clinically deployed assistive robot.

Figure 5 summarizes the teacher-data collection process and the separate replay buffers used for the two robot platforms.

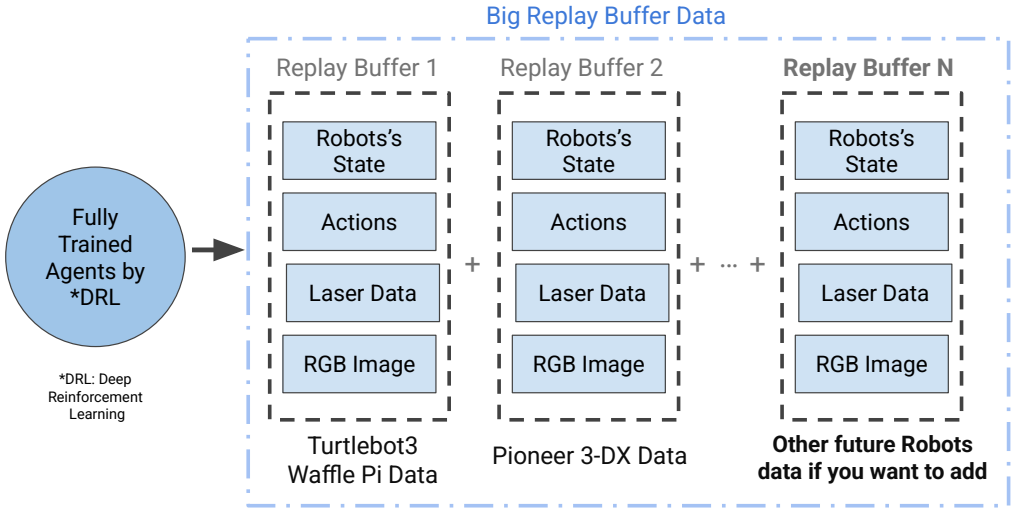


Figure 5. Data collection process from teacher models in simulation (see Figure 3). Robot states, LiDAR, and camera frames are recorded and stored in separate replay buffers.

2.4.1. Phase 1: Single-Robot Training

The student network contains two modality-specific convolutional branches. The one-dimensional LiDAR CNN processes the 20-sector range vector, whereas the two-dimensional camera CNN processes the grayscale image. Each branch outputs a 32-dimensional feature vector. The same shared encoder transforms this vector into a five-dimensional context representation. The actor receives both the direct 32-dimensional modality features and the encoded context, together with the four robot-state variables and, in the generalized baseline, the scalar robot token. It regresses the desired linear and angular velocities. Training alternates camera mini-batches on even optimization steps and LiDAR mini-batches on odd optimization steps to balance updates across modalities; this schedule does not itself imply complete feature decoupling. Figure 6 illustrates the architecture and training process for the single-robot model.

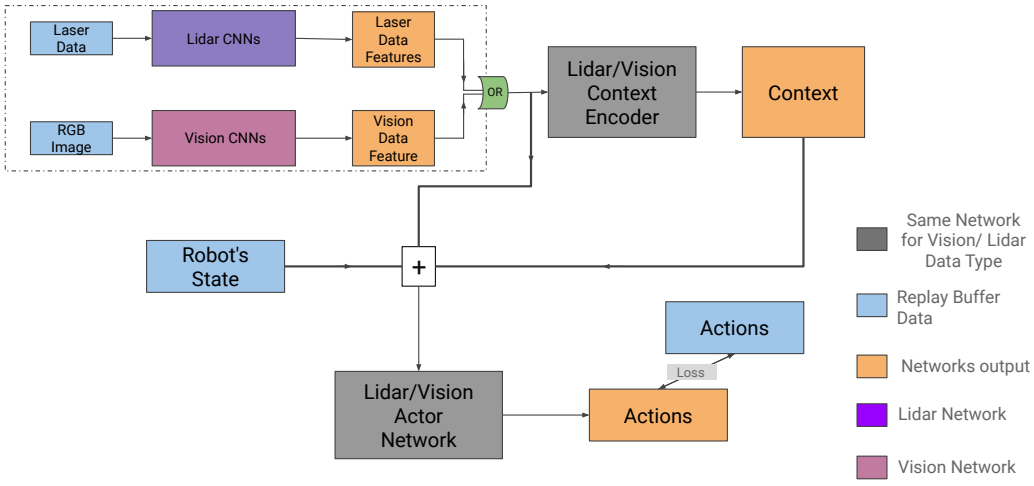


Figure 6. Phase 1: Architecture and training process for the single-robot model, leveraging alternating LiDAR and RGB sensor modalities. The dashed box denotes the modality-specific sensor-processing branches; only the branch corresponding to the selected active modality is used for a given input.

In a camera-training step, the student receives the camera-derived feature vector, the shared encoded context, the robot-state vector, and the applicable conditioning token; the LiDAR observation is not concatenated to the actor input. The loss is nevertheless com-

puted against the teacher action stored in the same replay-buffer transition. This is matched because the target is a control command in the robot action space, not a sensor measurement. The same reasoning applies to LiDAR-training steps, where the scan branch predicts the stored teacher action from the LiDAR observation and the same robot-state variables.

2.4.2. Phase 2: Generalization Across Robots

For cross-platform training, the Pioneer and Waffle replay buffers are retained separately. Each update draws an equal-sized mini-batch from each buffer and concatenates the samples, thereby preventing one platform from dominating the training batch. In the generalized baseline, a scalar Robot-ID token is appended to the actor input so that the shared policy can condition its action on the known platform. This conditioning substantially improves LiDAR-based generalization because similar 1D range patterns alone do not reliably identify the robot embodiment. Figure 7 illustrates the generalized student architecture and training pipeline.

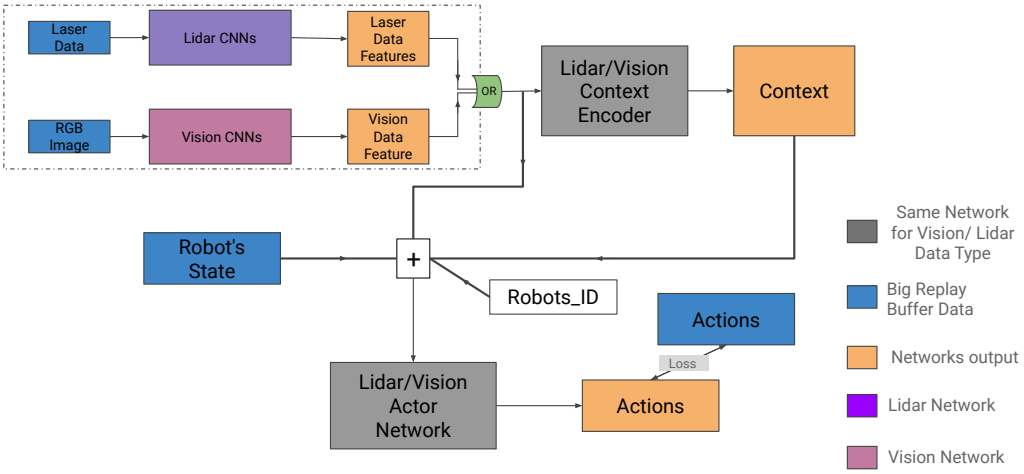


Figure 7. Phase 2: Architecture and training process for generalizing the student model for multiple robots, using a robot ID feature to align LiDAR inputs with specific robot embodiments while maintaining camera-based navigation. The dashed box denotes the modality-specific sensor-processing branches; only the branch corresponding to the selected active modality is used for a given input.

2.4.3. Training Hyper-Parameters

The student is optimized with Adam by minimizing mean-squared error between predicted actions and the teacher commands stored in the replay buffer. During generalized training, equal-sized requested subsets from the Pioneer and Waffle buffers are concatenated for each update. The camera pipeline uses grayscale inputs and a compact 32×64 student tensor, as detailed below.

The native Gazebo camera stream is a 640×480 color frame (width $\times$ height). The environment first resizes this frame to 128×64 pixels, converts it to grayscale, and stores the result in the replay buffer. During student training and evaluation, the stored frame is resized again to a 32×64 grayscale tensor (height $\times$ width) and normalized before entering the camera CNN. No cropping, padding, or aspect-ratio-preservation step is applied; therefore, the implemented pipeline intentionally trades spatial detail and geometric fidelity for a compact input representation.

The 32×64 configuration was selected as a compact input for the present student architecture. The reported vision-navigation results show that this representation can support the controlled ROS-Gazebo task used in this study. However, these results do not show that this resolution is optimal or suitable for all visual-navigation settings. A formal resolution-sensitivity comparison was not conducted in this study.

The camera branch is not required to estimate the goal position from image pixels. The goal distance and relative goal heading are provided separately through the robot-state input, together with the previous linear and angular velocities. The grayscale image therefore mainly provides information about the scene and nearby obstacles. Compact visual inputs have also been used in related navigation studies. Cimurs et al. used 80×64 single-channel depth images while providing the goal position separately in polar coordinates [31]. Niu et al. used 32×32 monocular RGB images to produce coarse depth information for obstacle-aware steering [32]. These studies do not show that 32×64 is optimal for the present model, but they support the use of compact visual inputs when fine-grained visual target recognition is not the main task. A controlled resolution-sensitivity study would require retraining the visual models under the same training and evaluation protocol and remains future work.

When the training set combines multiple robots, the generalized baseline appends a scalar Robot-ID token to the actor input. This explicit conditioning improves LiDAR-based performance in the multi-robot setting.

2.4.4. Final Model Performance

Sensor Modality Flexibility: The model can execute from either LiDAR or visual observations. Controlled LiDAR-to-vision continuation is evaluated later in Section 3.4; automatic sensor-failure detection and autonomous modality selection are outside the present scope.

The generalized evaluation assesses three specific capabilities. The student can execute from a selected LiDAR or visual input; the corresponding controlled handover study is reported in Section 3.4. It is evaluated on the two known Pioneer 3-DX and TurtleBot3 Waffle configurations, using the scalar Robot-ID token where specified. Its LiDAR outcomes are compared with the platform-specific teachers, and its camera outcomes are reported separately. These results are bounded by the simulated task definition and do not demonstrate automatic fault detection, physical deployment, or transfer to unseen platforms.

3. Results

This section reports a controlled simulation-based evaluation in ROS-Gazebo using Pioneer 3-DX and TurtleBot3 Waffle configurations. At each episode reset, the robot start pose, goal position, and the positions of four obstacles are randomized; obstacles remain static during an individual episode. This setting serves as an initial feasibility-validation stage because it enables repeatable comparisons of teacher–student transfer, Robot-ID conditioning, and within-episode modality handover while holding the task definition and control interface fixed. It does not model fog, rain, dust, snow, abrupt illumination changes, moving obstacles, or naturally occurring sensor failures. It also does not replicate all physical-robot effects, including sensor calibration errors, camera exposure changes, actuator delay, wheel slip, or unmodeled interactions. The following experiments therefore establish controlled evidence for the proposed architecture rather than a claim of completed real-world validation.

The result paragraphs below separate reported counts from interpretation. Success rates are stated directly from the logged episode outcomes, and causal explanations are limited to what the implemented architecture and ablation comparisons can support.

3.1. Single-Student Model Performance

Table 5 reports the single-student baselines. For each platform, the LiDAR teacher, LiDAR student, and vision student are evaluated over 1000 ROS-Gazebo episodes. The com-

parison measures episode completion in the same controlled environment; it is not a cross-study benchmark.

Table 5. Success rates of the single-student models on Pioneer 3-DX and TurtleBot3 Waffle Pi, compared to their LiDAR-based teacher models, over 1000 test runs.

Scenario	Pioneer 3-DX [1]			TurtleBot3 Waffle Pi [2]		
	Successes	Failures	Success Rate	Successes	Failures	Success Rate
Teacher-LiDAR	941	59	94.1%	911	89	91.1%
Student-LiDAR	916	84	91.6%	922	78	92.2%
Student-Vision	847	153	84.7%	842	158	84.2%

Note: Success rates reflect performance over 1000 test runs, comparing single-student models to their respective teacher models. [1] Pioneer 3-DX uses RGB image data for its vision-based student model. [2] TurtleBot3 Waffle Pi uses RGB image data for its vision-based student model.

3.2. Generalized Student Model Performance

Tables 6 and 7 compare the generalized student without and with scalar Robot-ID conditioning. Both tables retain the same teacher references and report 1000 ROS-Gazebo episodes per platform and active modality. This comparison isolates the observed effect of the conditioning token within the present two-platform training design.

Table 6. Success rates of the generalized student model **without** robot ID on Pioneer 3-DX and TurtleBot3 Waffle Pi, compared to their LiDAR-based teacher models, over 1000 test runs.

Scenario	Pioneer 3-DX [1]			TurtleBot3 Waffle Pi [2]		
	Successes	Failures	Success Rate	Successes	Failures	Success Rate
Teacher-LiDAR	941	59	94.1%	911	89	91.1%
Student-LiDAR	718	282	71.8%	795	205	79.5%
Student-Vision	869	131	86.9%	842	158	84.2%

Note: Success rates reflect performance of the generalized student model (without robot ID) over 1000 test runs. [1] Pioneer 3-DX uses RGB image data for its vision-based student model. [2] TurtleBot3 Waffle Pi uses RGB image data for its vision-based student model.

Table 7. Success rates of the generalized student model **with** robot ID on Pioneer 3-DX and TurtleBot3 Waffle Pi using LiDAR and RGB image data, compared to their LiDAR-based teacher models, over 1000 test runs.

Scenario	Pioneer 3-DX [1]			TurtleBot3 Waffle Pi [2]		
	Successes	Failures	Success Rate	Successes	Failures	Success Rate
Teacher-LiDAR	941	59	94.1%	911	89	91.1%
Student-LiDAR	904	96	90.4%	912	88	91.2%
Student-Vision	873	127	87.3%	840	160	84.0%

Note: Success rates reflect performance over 1000 test runs using LiDAR and RGB image data for both robots. [1] Pioneer 3-DX uses RGB image data for its vision-based student model. [2] TurtleBot3 Waffle Pi uses RGB image data for its vision-based student model.

3.3. Visualization of Simulation Evaluation Metrics

To complement the tables, Figures 8–10 visualize the aggregate ROS-Gazebo evaluation metrics. Figure 8 summarizes the 1000-episode success rates across the teacher, single-student, and generalized-student settings. Figure 9 isolates the Robot-ID ablation, and Figure 10 summarizes continuation success after the scheduled LiDAR-to-vision handover. These are simulation-evaluation results, not physical measurement data.

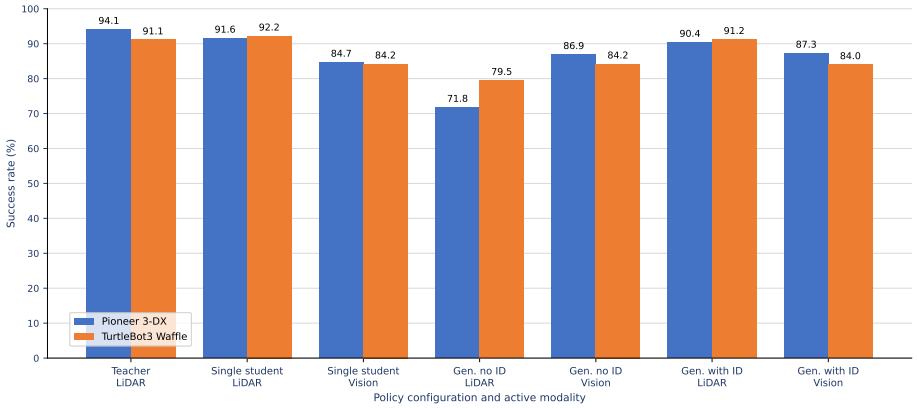


Figure 8. Success-rate summary across teacher, single-student, and generalized-student configurations. Each bar is calculated from 1000 ROS-Gazebo test episodes for the stated robot platform and active modality.

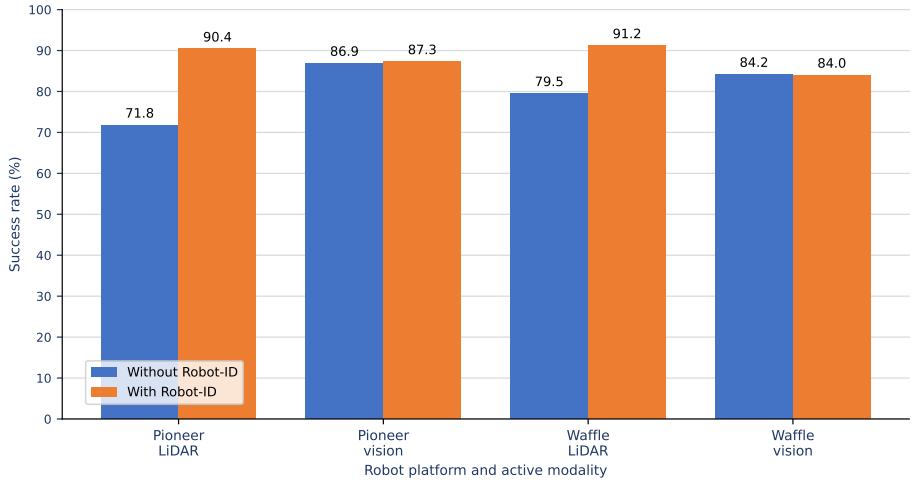


Figure 9. Robot-ID ablation for the generalized student. The scalar Robot-ID token substantially improves LiDAR success on both platforms, whereas the vision rates remain similar under the tested configurations. Each bar is calculated from 1000 ROS-Gazebo test episodes.

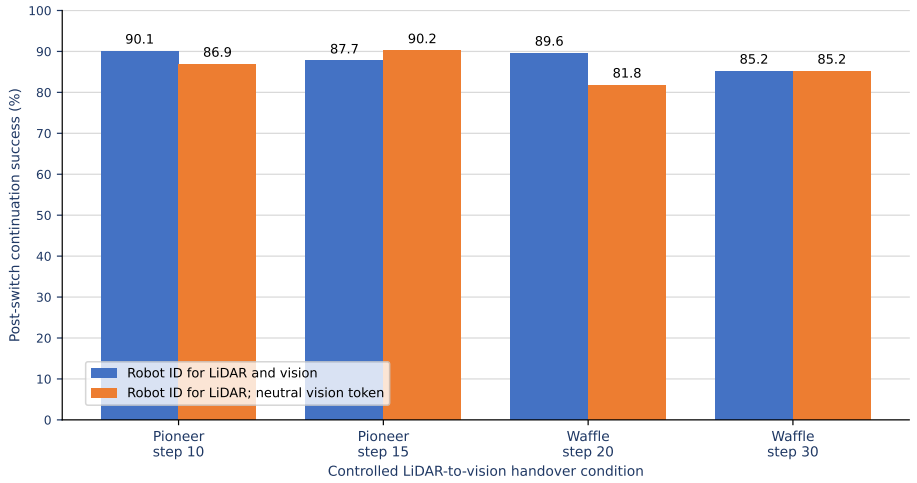


Figure 10. Post-switch continuation success for controlled LiDAR-to-vision handovers. Continuation success is calculated only over episodes that remained active at the scheduled switch step. Bars correspond to separate evaluation configurations; the full Robot-ID and asymmetric conditions are displayed separately because their schedules and other evaluation settings are not fully aligned for direct head-to-head comparison.

3.4. Controlled LiDAR-to-Vision Handover Evaluation

The static single-modality results demonstrate that the student can navigate with either LiDAR or vision. To test whether this capability persists during an episode, we conducted controlled LiDAR-to-vision handover evaluations without resetting or retraining the student policy. Each episode began with LiDAR-based action prediction. At a predefined switch step, only the sensor-specific feature-extraction branch was changed to the vision branch; the shared encoder, actor, and robot state remained in use. The handover was externally scheduled for evaluation and does not represent automatic sensor-failure detection or autonomous modality selection.

For transparency, the post-switch continuation success rate is calculated only over episodes that were still active at the scheduled switch step. Episodes that had already reached the goal or failed before the scheduled handover are excluded from this denominator.

When the platform token was supplied to both branches, post-switch continuation success ranged from 85.2% to 90.1% across the four controlled handover conditions in Table 8. These outcomes show that the shared action policy can continue navigation after the active perceptual branch changes from LiDAR to vision.

Table 8. Controlled LiDAR-to-vision handover results when the platform token is supplied to both the LiDAR and vision branches. Each condition contains 500 episodes.

Robot	Switch Step	Runs	Active at Switch	Overall Success	Post-Switch Goals	Post-Switch Failures	Timeouts	Continuation Success
Pioneer 3-DX	10	500	445	90.8%	401	38	6	90.1%
Pioneer 3-DX	15	500	351	89.4%	308	37	6	87.7%
TurtleBot3 Waffle	20	500	414	90.2%	371	38	5	89.6%
TurtleBot3 Waffle	30	500	325	88.8%	277	44	4	85.2%

We additionally evaluated an asymmetric conditioning configuration in which the LiDAR branch receives the platform token while the vision branch receives a neutral token of zero. This condition is reported as a conditioning-sensitivity experiment, rather than as a direct head-to-head ablation against Table 8, because the reported episode horizons and switch schedules are not identical across all conditions.

The asymmetric configuration also completed a large proportion of episodes after the scheduled handover, with continuation success ranging from 81.8% to 90.2% (Table 9). It therefore provides supporting evidence that the vision branch can sustain teacher-guided navigation without an explicit platform token. However, this result should be interpreted together with the different evaluation settings shown in the table and not as proof that the asymmetric configuration is uniformly preferable.

Table 9. Controlled LiDAR-to-vision handover results for the asymmetric conditioning configuration: LiDAR receives the platform token, whereas vision receives a neutral token. Each condition contains 500 episodes.

Robot	Switch Step	Runs	Active at Switch	Overall Success	Post-Switch Goals	Post-Switch Failures	Timeouts	Continuation Success
Pioneer 3-DX	10	500	442	87.6%	384	54	4	86.9%
Pioneer 3-DX	15	500	357	90.8%	322	33	2	90.2%
TurtleBot3 Waffle	30	500	332	89.4%	283	49	0	85.2%
TurtleBot3 Waffle	20	500	413	84.2%	338	74	1	81.8%

Supplementary Video S1 provides representative ROS-Gazebo rollouts for the controlled LiDAR-to-vision handovers summarized in Tables 8 and 9. It visually shows the student policy continuing navigation after the active perception branch changes from LiDAR to vision for the full Robot-ID and asymmetric conditioning configurations. The

video is qualitative support for the reported simulation trials; the quantitative conclusions are based on the aggregate results in the tables.

4. Discussion

The discussion interprets the simulation evidence in terms of action imitation, sensor-specific information, and platform conditioning. It avoids extending the conclusions to adverse weather, dynamic crowds, physical robots, or rehabilitation use cases that were not experimentally tested.

The discussion interprets the reported success rates in relation to the action-imitation objective, Robot-ID conditioning, image representation, and the controlled handover protocol. It distinguishes the evidence provided by the simulations from claims that would require physical-robot or human-centered validation.

4.1. Comparison with Related Navigation Approaches

The state-of-the-art scope comparison is provided in the Introduction (Table 1) to clarify the relation of this work to navigation studies using fixed LiDAR inputs, fixed visual inputs, or simultaneous LiDAR-camera fusion. Because those studies use different robots, tasks, environments, sensors, and protocols, this article does not treat their reported success rates as directly comparable with the present results.

The numerical evidence for the proposed design is therefore based on controlled comparisons within the same Gazebo environment. Relative to the LiDAR TD3 teachers, the student retains comparable LiDAR success while adding a vision-only execution path. Relative to the generalized student without platform conditioning, the scalar Robot-ID token improves LiDAR success from 71.8% to 90.4% on Pioneer 3-DX and from 79.5% to 91.2% on TurtleBot3 Waffle. The controlled handover results in Tables 8 and 9 further show that the policy can continue an episode after an externally scheduled LiDAR-to-vision change without policy reset or retraining. These are targeted advantages of the proposed framework; they are not presented as universal numerical superiority over the cited navigation studies.

4.2. Single-Student Model Analysis

The single-student baselines indicate how closely action imitation can reproduce the LiDAR teachers on each platform before the two replay-buffer sources are combined.

4.2.1. Comparison Between Teacher and Student Models Using LiDAR

For the Pioneer 3-DX, the student model's LiDAR-based navigation success-rate difference is given in Equation (10):

$$\text{Performance Difference (LiDAR)} = 94.1\% - 91.6\% = 2.5\% \tag{10}$$

For the TurtleBot3 Waffle Pi, the corresponding difference is given in Equation (11):

$$\text{Performance Difference (LiDAR)} = 91.1\% - 92.2\% = -1.1\% \tag{11}$$

On Pioneer 3-DX, the LiDAR student reaches 91.6%, which is 2.5 percentage points below the 94.1% teacher result. On TurtleBot3 Waffle, the LiDAR student reaches 92.2%, compared with 91.1% for its teacher. The close values are consistent with successful action imitation under the reported protocol. They do not, however, establish a statistically significant advantage of the student over the teacher, because the study reports episode counts and success rates rather than a formal significance analysis.

4.2.2. Comparison Between Teacher and Student Models Using RGB Images

For the Pioneer 3-DX, the vision-based difference is given in Equation (12):

$$\text{Performance Difference (Vision)} = 94.1\% - 84.7\% = 9.4\% \quad (12)$$

For the TurtleBot3 Waffle Pi, the corresponding vision-based difference is given in Equation (13):

$$\text{Performance Difference (Vision)} = 91.1\% - 84.2\% = 6.9\% \quad (13)$$

The camera branch reaches 84.7% on Pioneer 3-DX and 84.2% on TurtleBot3 Waffle. Those values are 9.4 and 6.9 percentage points below the corresponding LiDAR teachers. The gap is compatible with the information available to the two policies: the teacher receives range measurements, whereas the camera branch must infer obstacle-related cues from a compact grayscale image. The result supports vision-based execution in the tested simulation setting, but it does not establish equivalent sensing quality, robustness to illumination changes, or performance in more complex scenes.

These vision results should be interpreted in the context of action-level supervision. The teacher action is generated by a LiDAR-based policy, but the student is evaluated by whether the camera branch can issue a useful velocity command in the same closed-loop navigation task. The student is not expected to reproduce LiDAR readings or to observe the environment through the same sensor geometry as the teacher. The remaining performance gap between teacher LiDAR and student vision is therefore consistent with the fact that the camera input provides indirect appearance cues rather than explicit range measurements.

Under the reported protocol, the 32×64 camera input achieved 84.7% success for the Pioneer 3-DX single-student model and 84.2% for the TurtleBot3 Waffle single-student model. These outcomes show that the selected compact representation supports the present simulated navigation task; they do not show that fine visual detail is preserved or that 32×64 is the best resolution. In particular, the goal distance and relative goal heading are supplied separately in the robot-state vector, so the camera branch is not required to locate a distant goal solely from pixels. Its visual input primarily contributes scene and obstacle-appearance cues. The current study therefore does not claim resolution robustness, and the effect of larger images on distant-obstacle perception remains an open empirical question.

4.3. Generalized Student Model Analysis

The generalized experiments test whether one shared student can use demonstrations from both platform-associated buffers and still produce appropriate actions for each known robot configuration.

4.3.1. Performance Without Robot-ID

For the Pioneer 3-DX, the LiDAR difference without Robot-ID is given in Equation (14):

$$\text{Performance Difference (LiDAR)} = 94.1\% - 71.8\% = 22.3\% \quad (14)$$

For the TurtleBot3 Waffle Pi, the corresponding difference without Robot-ID is given in Equation (15):

$$\text{Performance Difference (LiDAR)} = 91.1\% - 79.5\% = 11.6\% \quad (15)$$

Without Robot-ID conditioning, generalized LiDAR success falls to 71.8% on Pioneer 3-DX and 79.5% on TurtleBot3 Waffle, while the camera results are 86.9% and 84.2%, respectively. The contrast indicates that the shared LiDAR control mapping benefits from explicit

platform context in this two-robot setting. It does not identify a single-cause property of the sensor streams. In particular, the reported outcomes do not directly prove that camera images encode robot identity, that LiDAR observations lack every embodiment cue, or that the same pattern would hold for other robots or environments. Those interpretations would require targeted representation analyses and additional controlled experiments.

4.3.2. Performance with Robot-ID

For the Pioneer 3-DX, the LiDAR difference with Robot-ID is given in Equation (16):

$$\text{Performance Difference (LiDAR)} = 94.1\% - 90.4\% = 3.7\% \tag{16}$$

For the TurtleBot3 Waffle Pi, the corresponding difference with Robot-ID is given in Equation (17):

$$\text{Performance Difference (LiDAR)} = 91.1\% - 91.2\% = -0.1\% \tag{17}$$

With scalar Robot-ID conditioning, LiDAR success increases to 90.4% on Pioneer 3-DX and 91.2% on TurtleBot3 Waffle. The ablation therefore supports explicit platform conditioning for the shared LiDAR policy under the reported training and evaluation protocol. It does not reveal which internal features changed or establish that the token would remain sufficient for a larger and more heterogeneous robot set.

4.4. Data Distribution, Modality-Specific Representation, and Closed-Loop Interpretation

The reported success rates are interpreted here together with the training-data distribution, policy architecture, and closed-loop evaluation protocol. Success rate is an aggregate outcome measure under the present ROS-Gazebo setup; by itself, it does not directly measure feature separability or establish robustness under physical-robot conditions.

4.4.1. Training-Data Distribution and Platform Balance

The generalized-student training process preserves platform provenance by storing Pioneer 3-DX and TurtleBot3 Waffle trajectories in separate replay buffers. Before training, the entries of each buffer are shuffled and divided into training and held-out replay subsets using an 85:15 split. During every generalized-student update, equal-sized subsets are requested from the Pioneer and Waffle buffers and concatenated into one batch. Therefore, the two platforms contribute the same number of sampled transitions within an update, and the LiDAR degradation observed without Robot-ID conditioning cannot be explained simply by one platform contributing a larger number of training samples. This balancing does not imply that the two data sources have identical task difficulty, trajectory geometry, or action distributions.

The replay buffers contain trajectories generated under randomized ROS-Gazebo resets. At the beginning of each episode, the robot pose, goal position, and positions of four obstacles are randomized; however, the obstacles remain static during that episode. Consequently, the demonstrations cover multiple stationary obstacle arrangements and goal configurations, but they do not represent moving-obstacle trajectories, illumination changes, severe weather, or physical sensor disturbances. The held-out replay subset measures transition-level action-prediction behavior, whereas the 1000-episode success-rate tests evaluate complete closed-loop navigation. These two evaluations answer different questions and should not be interpreted interchangeably.

4.4.2. Robot-ID Ablation and Representation Interpretation

The Robot-ID ablation provides the clearest evidence about the relation between platform conditioning and the learned control mapping. As shown in Tables 6 and 7, adding the scalar Robot-ID token increases LiDAR success from 71.8% to 90.4% for Pioneer 3-DX, an increase of 18.6 percentage points, and from 79.5% to 91.2% for TurtleBot3 Waffle, an increase of 11.7 percentage points. In contrast, the corresponding vision success rates change only from 86.9% to 87.3% for Pioneer 3-DX and from 84.2% to 84.0% for TurtleBot3 Waffle.

This pattern is consistent with the LiDAR observations alone providing insufficient platform-specific context for the shared actor to select the most appropriate control mapping across the two robot configurations. The small vision-rate changes indicate that the evaluated vision pathway remained comparatively stable when Robot-ID conditioning was introduced. However, these outcome statistics do not directly prove that the camera branch explicitly identifies robot embodiment, because the present study does not include a latent-space visualization or a dedicated feature-probing experiment. The discussion therefore treats the result as performance-based evidence rather than a direct measurement of internal feature semantics.

4.4.3. Modality-Specific Processing and Shared Control Learning

The architecture separates early sensory processing while sharing the downstream control representation. The LiDAR branch processes a 20-sector range representation, whereas the camera branch processes a normalized 32×64 grayscale image. Each modality-specific CNN outputs 32 direct features. These features are passed through the same encoder to produce a five-dimensional context vector, and the actor receives one conditioning token, 32 direct features, five context features, and four robot-state values, for a total input dimension of 42. Thus, the two modalities have separate early feature extractors but are constrained to support the same continuous linear- and angular-velocity output space.

The alternating LiDAR and camera updates balance the exposure of the shared encoder and actor to the two modalities. They do not, by themselves, guarantee complete feature decoupling or explicit cross-modal alignment, because no contrastive, adversarial, or modality-disentanglement loss is used. The appropriate interpretation is therefore modality-specific early processing followed by shared action learning: each front end must produce representations that allow the common encoder and actor to imitate the same teacher-guided navigation objective.

4.4.4. Closed-Loop Error, Visual Information, and Handover Interpretation

For each stored transition, the student receives either the paired LiDAR observation or the paired camera observation and is supervised by the LiDAR-teacher action recorded at that same control step. The mean-squared action loss therefore evaluates transition-level imitation. During closed-loop navigation, however, even a small action discrepancy can move the student into a state that differs from the teacher trajectory, causing later visual and LiDAR observations to differ from those represented in the demonstrations. This distribution shift provides a plausible explanation for why low action-imitation error does not necessarily translate into identical episode-level success.

The visual pathway must additionally infer obstacle geometry from grayscale image information without explicit depth measurements. In the implemented pipeline, the incoming camera frame is first converted to a 128×64 grayscale image and is then resized to 32×64 for the student network. This reduces computational cost but also limits spatial detail available to the vision branch. The current experiments do not isolate the separate effects of image resolution, camera viewpoint, missing depth, and accumulated action

error; therefore, the observed vision gap relative to LiDAR should not be attributed to a single cause.

The controlled LiDAR-to-vision experiments provide an additional closed-loop perspective. Their post-switch continuation success is calculated only over episodes that remained active at the scheduled handover, and it ranges from 81.8% to 90.2% across the reported conditions. These results show that the same student policy can continue navigation after an externally scheduled change in the active perception branch without reset or retraining. They do not demonstrate automatic failure detection or prove that the full Robot-ID and asymmetric conditions are directly comparable in every switch setting, because their evaluation schedules are not fully aligned.

4.5. Practical Relevance to Rehabilitation and Assistive Robotics

Rehabilitation and assistive robotics encompass therapeutic, social, and service-oriented functions that may be delivered in rehabilitation clinics, hospitals, long-term care facilities, or private homes. Existing literature has described socially assistive robots for post-stroke rehabilitation and mobile robotic assistants intended to support older adults and care staff [10,11]. Across these application domains, a mobile robot must first move safely and predictably through shared indoor spaces before any higher-level service, interaction, or therapy function can be considered. The present study addresses only this navigation layer through a ROS-Gazebo evaluation; it does not implement rehabilitation therapy, patient assessment, manipulation, clinical decision-making, or healthcare workflow support.

Within this limited scope, LiDAR-to-vision continuity could support future assistive mobile robots by providing a candidate navigation-resilience mechanism when the active sensing modality is degraded or unavailable. The current handover experiments show continuation after an externally scheduled change in the active perception branch, using the same student policy without reset or retraining. They do not demonstrate automatic sensor-failure detection or safe operation during a real sensing fault. Similarly, the shared actor and lightweight Robot-ID conditioning could reduce adaptation effort when a future service or assistive system is transferred between the two known mobile bases examined here. This result does not establish transfer to unseen robot embodiments, clinical deployment, or a reduction in deployment effort measured on physical robots.

Navigation infrastructure alone is insufficient for rehabilitation or healthcare use. Before a system of this type could be considered for those settings, it would require person-aware navigation, explicit human-robot interaction design, evaluation with moving people and dynamic obstacles, physical-robot validation, safety assessment, and user-centered studies. Human-aware navigation must account for people, their motion, and socially acceptable robot behavior [33]; hospital-oriented navigation studies also motivate assessment under progressively more realistic static and dynamic conditions [34]. Future validation should therefore examine reliability, safety, usability, workflow fit, and trust with intended users, caregivers, therapists, and clinicians. None of these rehabilitation- or healthcare-specific evaluations has been completed in the present study.

Simulation Scope, Environmental Limitations, and Future Physical Validation

The present study reports simulation-only results. This is a deliberate first validation stage: the ROS-Gazebo campaign provides a controlled and repeatable environment for testing whether the proposed action-distillation, Robot-ID conditioning, and LiDAR-to-vision handover mechanisms are feasible before introducing hardware-specific variability. It is not intended to substitute for physical-robot validation. The randomized resets provide several stationary obstacle layouts, but the experiments do not include moving

obstacles, severe weather, dust, rain, fog, snow, abrupt illumination changes, or natural sensor-failure injection. These sensing conditions are the motivation for modality-flexible navigation, not a robustness claim established by the current evaluation. Subsequent physical-robot experiments will evaluate the trained policy on a TurtleBot platform under real sensor noise, camera-exposure variation, calibration mismatch, actuation delay, wheel slip, safety constraints, dynamic obstacles, and controlled environmental perturbations. These experiments will determine the extent to which the controlled simulation findings transfer to operating robots.

Future rehabilitation and assistive-robotics validation should begin with physical testing in realistic indoor care or rehabilitation-like environments. Such studies should include moving people, variable illumination, sensor noise or controlled sensor failure, human-aware navigation, and safety-relevant behavior during interaction with staff, users, or clinicians. Collaboration with intended users and clinical stakeholders would be needed to evaluate usability, workflow fit, trust, and appropriate safety requirements; no such evaluation has been performed in the present study.

5. Conclusions

The conclusion is limited to the simulation study reported in this manuscript and does not restate the method as a deployed navigation product.

This study evaluates a conditional teacher–student policy for navigation on Pioneer 3-DX and TurtleBot3 Waffle simulations. Platform-specific LiDAR TD3 teachers provide action labels, and one student is trained to predict the same continuous command from either LiDAR or camera observations. The system does not use simultaneous LiDAR-vision fusion at execution. Instead, it shares downstream control layers across two modality-specific feature extractors and applies scalar Robot-ID conditioning in the generalized configuration.

Together with the controlled handover evaluations, these results establish that LiDAR-to-vision distillation can provide a candidate fallback mechanism under the tested simulation protocol. They do not yet demonstrate robustness to unmodeled physical sensor failures.

In the single-platform evaluations, LiDAR-student success is close to the associated teacher result, and the camera branch completes more than 84% of the simulated episodes on both platforms. These findings show that the student can map either selected modality to useful velocity commands in the stated environment.

The cross-platform experiments further show the importance of embodiment-aware conditioning. When data from both robots were merged without a robot identifier, LiDAR-based performance dropped substantially, suggesting that one-dimensional LiDAR observations alone were insufficient for the policy to infer the correct robot embodiment.

Adding a scalar Robot-ID token largely recovered this loss, increasing LiDAR-based success rates from 71.8% to 90.4% on the Pioneer 3-DX and from 79.5% to 91.2% on the TurtleBot3 Waffle Pi. At the same time, vision-based navigation achieved success rates of 87.3% and 84.0% on the two platforms. These findings show that lightweight robot-identity conditioning can substantially improve cross-platform performance on the two evaluated robots without requiring platform-specific fine-tuning.

The deeper interpretation of these outcomes is that balanced platform sampling alone is not sufficient to resolve embodiment-dependent LiDAR control. The substantial LiDAR improvement after introducing Robot-ID conditioning, together with the nearly unchanged vision success rates, is consistent with a modality-dependent need for explicit platform context in the shared policy. This interpretation is based on the observed success-rate ablation rather than direct latent-feature measurements. Moreover, the remaining vision-performance gap is consistent with the limited spatial information and lack of explicit

depth in the grayscale input, as well as with the accumulation of action-imitation errors during closed-loop execution.

The study provides controlled evidence for three design choices: action-level distillation from LiDAR teachers, modality-specific front ends with a shared control pathway, and scalar Robot-ID conditioning for the two evaluated robots. The framework may reduce the need for separate modality-specific control policies within this restricted setting, but it is not a substitute for sensor-health estimation, explicit safety supervision, or physical validation. Its relevance to service and assistive mobile robots therefore remains prospective and depends on the additional evaluations described in the discussion.

The current study is intentionally a controlled simulation-based proof-of-feasibility evaluation on two known robot configurations. It establishes the internal behavior of the proposed architecture under repeatable randomized navigation tasks, but it does not replace physical-robot validation or establish robustness to real lighting variation, weather, sensor noise, calibration drift, latency, wheel-ground interaction, or moving obstacles. Supplementary Video S1 provides representative simulation rollouts for visual inspection of the controlled handovers; it does not depict a physical-robot experiment. The framework also assumes that a valid active modality is available and does not include automatic sensor-health monitoring or autonomous modality selection. Future work will deploy the policy on physical robots, broaden the robot and environment set, and add sensor-health estimation. Richer embodiment descriptors may further improve scalability beyond the scalar Robot-ID token.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/machines14080936/s1, Supplementary Video S1: Representative ROS-Gazebo rollouts for the controlled LiDAR-to-vision handovers summarized in Tables 8 and 9.

Author Contributions: Conceptualization, A.M.A., S.A. and A.M.; methodology, A.M.A., S.A. and A.M.; software, A.M.A.; validation, A.M.A., S.A. and A.M.; formal analysis, A.M.A., S.A. and A.M.; investigation, A.M.A.; resources, A.M.A., S.A. and A.M.; data curation, A.M.A.; writing-original draft preparation, A.M.A.; writing-review and editing, S.A. and A.M.; visualization, A.M.A.; supervision, S.A. and A.M.; project administration, A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request. The authors have also made the project source code available on the GitHub profile: https://github.com/am-amani/multimodal-drl-robot-navigation (accessed on 28 May 2026).

Acknowledgments: The authors used AI-based language tools solely for refining the language, correcting grammar, and improving the readability of the manuscript. The authors carefully reviewed and edited all AI-assisted text and take full responsibility for the accuracy, integrity, and final content of the manuscript.

Conflicts of Interest: Author Sajjad Amani was employed by Seaos Co., Ltd. The company had no involvement in the research, funding, study design, data collection, analysis, manuscript preparation, or decision to publish. The remaining authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DRL	Deep reinforcement learning
KD	Knowledge distillation
LiDAR	Light detection and ranging

MSE	Mean squared error
RGB	Red-green-blue
RL	Reinforcement learning
TD3	Twin Delayed Deep Deterministic Policy Gradient

References

1. Pan, Y.; Geng, X.; Sun, X.; Zhao, H. Deep learning for 3D perception in autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3937–3954.
2. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer: London, UK, 2010. [CrossRef]
3. Chang, F.; Jafarzadeh, E.; Del Gatto, J.; Cran, G.; Sadjadi, H. Failure Mode Investigation to Enable LiDAR Health Monitoring for Automotive Application. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society*; PHM Society: State College, PA, USA, 2023; Volume 15. [CrossRef]
4. Kutila, M.; Pyykönen, P.; Holzhüter, H.; Colomb, M.; Duthon, P. Automotive LiDAR performance verification in fog and rain. In *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*; IEEE: New York, NY, USA, 2018; pp. 1695–1701. [CrossRef]
5. Dreissig, M.; Scheuble, D.; Piewak, F.; Boedecker, J. Survey on LiDAR Perception in Adverse Weather Conditions. In *Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV)*; IEEE: New York, NY, USA, 2023; pp. 1–8. [CrossRef]
6. Sattler, T.; Maddern, W.; Toft, C.; Torii, A.; Hammarstrand, L.; Stenborg, E.; Safari, D.; Okutomi, M.; Pollefeys, M.; Sivic, J.; et al. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. *arXiv* **2017**, arXiv:1707.09092. [CrossRef]
7. Wozniak, M.K.; Hansson, M.; Thiel, M.; Jensfelt, P. UADA3D: Unsupervised Adversarial Domain Adaptation for 3D Object Detection With Sparse LiDAR and Large Domain Gaps. *IEEE Robot. Autom. Lett.* **2024**, *9*, 11210–11217. [CrossRef]
8. Huang, W.; Mordatch, I.; Pathak, D. One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control. *arXiv* **2020**, arXiv:2007.04976. . [CrossRef]
9. Shah, D.; Sridhar, A.K.; Bhorkar, A.; Hirose, N.; Levine, S. GNM: A General Navigation Model to Drive Any Robot. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2023; pp. 7226–7233. [CrossRef]
10. Matarić, M.J.; Eriksson, J.; Feil-Seifer, D.J.; Winstein, C.J. Socially Assistive Robotics for Post-Stroke Rehabilitation. *J. NeuroEng. Rehabil.* **2007**, *4*, 5. [CrossRef] [PubMed]
11. Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; Thrun, S. Towards Robotic Assistants in Nursing Homes: Challenges and Results. *Robot. Auton. Syst.* **2003**, *42*, 271–281. [CrossRef]
12. Lynen, S.; Achtelik, M.W.; Weiss, S.; Chli, M.; Siegwart, R. A Robust and Modular Multi-Sensor Fusion Approach Applied to MAV Navigation. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: New York, NY, USA, 2013; pp. 3923–3929. [CrossRef]
13. Huang, X.; Deng, H.; Zhang, W.; Song, R.; Li, Y. Towards Multi-Modal Perception-Based Navigation: A Deep Reinforcement Learning Method. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4986–4993. [CrossRef]
14. Song, H.; Li, A.; Wang, T.; Wang, M. Multimodal Deep Reinforcement Learning with Auxiliary Task for Obstacle Avoidance of Indoor Mobile Robot. *Sensors* **2021**, *21*, 1363. [CrossRef] [PubMed]
15. Liu, G.H.; Siravuru, A.; Prabhakar, S.; Veloso, M.; Kantor, G. Learning End-to-end Multimodal Sensor Policies for Autonomous Navigation. In *Proceedings of the 1st Annual Conference on Robot Learning*; PMLR: Cambridge, MA, USA, 2017; Volume 78, pp. 249–261.
16. Bohez, S.; Verbelen, T.; Coninck, E.D.; Vankeirsbilck, B.; Simoens, P.; Dhoedt, B. Sensor Fusion for Robot Control through Deep Reinforcement Learning. *arXiv* **2017**, arXiv:1703.04550. [CrossRef]
17. Fujimoto, S.; van Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. *arXiv* **2018**, arXiv:1802.09477. [CrossRef]
18. Huang, B.; Xie, J.; Yan, J. Inspection Robot Navigation Based on Improved TD3 Algorithm. *Sensors* **2024**, *24*, 2525. [CrossRef] [PubMed]
19. Cheng, C.; Zhang, H.; Sun, Y.; Tao, H.; Chen, Y. A Cross-Platform Deep Reinforcement Learning Model for Autonomous Navigation without Global Information in Different Scenes. *Control Eng. Pract.* **2024**, *150*, 105991. [CrossRef]
20. Rusu, A.A.; Colmenarejo, S.G.; Gulcehre, C.; Desjardins, G.; Kirkpatrick, J.; Pascanu, R.; Mnih, V.; Kavukcuoglu, K.; Hadsell, R. Policy Distillation. In *Proceedings of the International Conference on Learning Representations (ICLR)*; ICLR: Appleton, WI, USA, 2016.
21. Schmitt, S.; Hudson, J.J.; Zidek, A.; Osindero, S.; Doersch, C.; Czarnecki, W.M.; Leibo, J.Z.; Kuttler, H.; Zisserman, A.; Simonyan, K.; et al. Kickstarting Deep Reinforcement Learning. *arXiv* **2018**, arXiv:1803.03835. [CrossRef]
22. Chen, D.; Zhou, B.; Koltun, V.; Krähenbühl, P. Learning by Cheating. *arXiv* **2019**, arXiv:1912.12294. [CrossRef]
23. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. *arXiv* **2017**, arXiv:1710.06537. [CrossRef]

24. Clavera, I.; Nagabandi, A.; Fearing, R.S.; Abbeel, P.; Levine, S.; Finn, C. Learning to Adapt: Meta-Learning for Model-Based Control. *arXiv* **2018**, arXiv:1803.11347. [CrossRef]
25. Reed, S.; Zolna, K.; Parisotto, E.; Colmenarejo, S.G.; Novikov, A.; Barth-Maron, G.; Gimenez, M.; Sulsky, Y.; Kay, J.; Springenberg, J.T.; et al. A Generalist Agent. *arXiv* **2022**, arXiv:2205.06175. [CrossRef]
26. Pfeiffer, M.; Shukla, S.; Turchetta, M.; Cadena, C.; Krause, A.; Siegwart, R.; Nieto, J. Reinforced Imitation: Sample Efficient Deep Reinforcement Learning for Map-less Navigation by Leveraging Prior Demonstrations. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4423–4430. [CrossRef]
27. Chiang, H.T.L.; Faust, A.; Fiser, M.; Francis, A. Learning Navigation Behaviors End-to-End With AutoRL. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2007–2014. [CrossRef]
28. Devo, A.; Mezzetti, G.; Costante, G.; Fravolini, M.L.; Valigi, P. Towards Generalization in Target-Driven Visual Navigation by Using Deep Reinforcement Learning. *IEEE Trans. Robot.* **2020**, *36*, 1546–1561. [CrossRef]
29. Han, Y.; Zhan, I.H.; Zhao, W.; Pan, J.; Zhang, Z.; Wang, Y.; Liu, Y.J. Deep Reinforcement Learning for Robot Collision Avoidance With Self-State-Attention and Sensor Fusion. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6886–6893. [CrossRef]
30. Shah, D.; Sridhar, A.; Dashora, N.; Stachowicz, K.; Black, K.; Hirose, N.; Levine, S. ViNT: A Foundation Model for Visual Navigation. In *Proceedings of the 7th Conference on Robot Learning*; PMLR: Cambridge, MA, USA, 2023; Volume 229, pp. 711–733.
31. Cimurs, R.; Lee, J.H.; Suh, I.H. Goal-Oriented Obstacle Avoidance with Deep Reinforcement Learning in Continuous Action Space. *Electronics* **2020**, *9*, 411. [CrossRef]
32. Niu, C.; Newlands, C.; Zauner, K.P.; Tarapore, D. An Embarrassingly Simple Approach for Visual Navigation of Forest Environments. *Front. Robot. AI* **2023**, *10*, 1086798. [CrossRef] [PubMed]
33. Kruse, T.; Pandey, A.K.; Alami, R.; Kirsch, A. Human-Aware Robot Navigation: A Survey. *Robot. Auton. Syst.* **2013**, *61*, 1726–1743. [CrossRef]
34. Rondoni, C.; Scotto di Luzio, F.; Tamantini, C.; Tagliamonte, N.L.; Chiurazzi, M.; Ciuti, G.; Zollo, L. Navigation Benchmarking for Autonomous Mobile Robots in Hospital Environment. *Sci. Rep.* **2024**, *14*, 18334. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.