

## Article

# PV-STAM: Velocity-Aware Attention for Mapless Deep Reinforcement Learning Navigation in Dynamic Environments

Anas Mahyoub Naji Saeed Alqadhi, Munef El Muhammed , Mohammed Ali M. S. Bajhaw and Aysegul Ucar \* 

RAI Laboratory, Mechatronics Engineering Department, Firat University, 23119 Elazığ, Türkiye; anas.m.qd@gmail.com (A.M.N.S.A.); munafa662@gmail.com (M.E.M.); mohammedbajhaw2020@gmail.com (M.A.M.S.B.)

\* Correspondence: agulucar@firat.edu.tr

## Abstract

Mapless deep reinforcement learning (DRL) navigation in dynamic indoor environments is difficult under single-frame 2D LiDAR, which reports where an obstacle is but not whether it is approaching. We introduce the Positional-Velocity Spatio-Temporal Attention Module (PV-STAM), a compact perception block (19,968 trainable parameters, under 3% of network capacity) that combines a per-sector scan-difference channel with a two-head self-attention mechanism and a  $384 \rightarrow 48$  compression bottleneck. Policies are trained with Soft Actor-Critic across a seven-phase progressive curriculum with bidirectional demotion, scaling from static goal-seeking to fifteen simultaneously moving obstacles at 0.18 m/s, with a Hardware-Calibrated Training Mode applied in the final phase. Seven configurations were evaluated on three zero-shot benchmark arenas over 300 episodes each across three random seeds, and key control variants were validated in 130 valid physical trials on a TurtleBot3 Waffle Pi across two matched corridor scenarios. The simulation and physical evaluations give different orderings, and this dissociation is the paper's principal result. In simulation, SAC-PV-STAM, SAC-R-PV-STAM and SAC-MLP-FS lie within 4.6 percentage points of one another on two of three benchmarks and are not separable; on hardware, they separate with large margins. SAC-R-PV-STAM reached the goal in 19 of 20 static trials and 20 of 20 trials with a moving obstacle, with no threshold violations across 40 trials, against SAC-MLP-FS (stratified  $p = 0.00068$ ) and SAC-PV-STAM (stratified  $p = 1.1 \times 10^{-7}$ ). A non-recurrent variant stalled with a clear floor ahead—SAC-PV-STAM commanded no forward velocity above 0.005 m/s in any of the 2746 samples of the final 91.3 s of a 122.3 s stall while the LiDAR reported 3.50 m directly ahead—and a prediction stated before the experiment, that introducing a moving obstacle would remove the stall, was confirmed (timeouts 9/10 to 0/20,  $p = 7 \times 10^{-7}$ ). We further report a scan-difference contamination analysis showing up to  $12.0\times$  more spurious spikes during self-rotation on hardware, a width-matched comparison separating the effect of Huber critic loss from that of critic width, and a recurrent LSTM baseline that remains significantly inferior across all three benchmarks and unstable across seeds.



Academic Editor: Alessandro Di Nuovo

Received: 11 July 2026

Revised: 6 September 2026

Accepted: 8 September 2026

Published: 13 September 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

**Keywords:** mapless navigation; deep reinforcement learning; spatio-temporal attention; LiDAR; dynamic obstacle avoidance; sim-to-real transfer

## 1. Introduction

An indoor robot rarely operates in a world that holds still. People walk past it, furniture is shifted, and other robots cross its path, so the layout in front of it is not the

layout it observed a moment earlier. SLAM-based planners perform well while the map remains fixed, but when obstacles move faster than the mapping pipeline refreshes, the map becomes stale and navigation degrades [1]. Mapless DRL removes the map: the robot learns to act directly from a raw LiDAR scan and a goal signal, which transfers to layouts not seen during training [2,3].

A single LiDAR scan captures distance but not motion. It reports where an obstacle is, not whether it is approaching. When several obstacles move at once, the policy must allocate avoidance effort on the basis of proximity alone [4]. Frame stacking provides the network with a short history, from which motion must be inferred indirectly. Recurrent networks such as GRU and LSTM [5,6] capture longer temporal dependencies at the cost of sequence-based replay and burn-in [7].

PV-STAM supplies the per-sector change in range between consecutive scans as an input channel so that motion information does not have to be recovered from a sequence of past scans. The quantity requires no scan registration, no optical flow, and no maintained recurrent state. In a scene containing several obstacles, the sectors are not equally urgent, which motivates attention: multi-head self-attention [8] over 24 LiDAR sectors weights proximity and closing rate jointly at low computational cost.

The evaluation has two parts. In simulation, every configuration is evaluated for 100 episodes per seed on each of three benchmark arenas, pooled over three seeds. On hardware, the key variants are evaluated in 130 valid trials across two matched corridor scenarios, with the validity of each trial verified from its recording. The two evaluations do not agree, and the disagreement is the principal result of this paper.

This paper makes five contributions.

1. The PV-STAM module, a sector-attention mechanism adding 19,968 trainable parameters to the policy network, is specified layer by layer in Section 3.5.
2. A seven-phase progressive curriculum with success-rate advancement thresholds and bidirectional demotion, supporting stable training under increasing obstacle density and speed [9].
3. A dissociation between simulation and physical evaluation. On two of the three benchmark arenas, the three temporally informed variants are not separable; across 130 physical trials, they separate with large margins.
4. Identification and controlled testing of an observation-stasis failure mode in non-recurrent policies, with the prediction stated before the confirming experiment was run.
5. A width-matched analysis of Huber critic regularization, showing that its effect cannot be separated from critic width: at matched width, success rises on one benchmark and falls on the other two.

Section 2 reviews related work. Section 3 describes the system architecture. Section 4 presents the curriculum. Section 5 describes the experimental setup. Section 6 reports simulation results. Section 7 reports the physical evaluation. Section 8 discusses the findings, states limitations and concludes.

## 2. Related Work

### 2.1. DRL for Mapless Navigation

Mapless DRL navigation began with Tai et al. [2], who demonstrated asynchronous DQN with sparse laser readings. Later work adopted continuous-action methods: DDPG for multi-robot collision avoidance [10], curiosity-driven exploration [3], and SAC with entropy regularization [11,12]. Beyond navigation, deep reinforcement learning has been applied to continuous control in other physical domains; in active flow control, Liu et al. [13]

compare on-policy and off-policy formulations and report that the sample budget demanded by off-policy replay becomes the binding constraint when each environment interaction is expensive, a consideration that also bears on the replay-buffer sizing adopted here (Section 5.3). Recent work has applied sensor fusion to autonomous navigation [14] and adaptive crowd navigation under emergency conditions [15]. A 2025 review of DRL navigation in dynamic environments [16,17] identifies sim-to-real transfer as the principal open problem in the area. The present work provides the per-sector range difference as an explicit input channel to an attention module rather than requiring the network to infer motion from history.

## 2.2. Attention Mechanisms in Robot Navigation

The Transformer [8] introduced scaled dot-product self-attention, later adapted to reinforcement learning by Parisotto et al. [18] and Chen et al. [19]. Social-attention models for crowd-aware navigation were introduced by SARL [20] under the assumption that pedestrian velocities are known. Yang et al. [21] applied a spatial-temporal state transformer to crowd-aware navigation. Spatio-temporal graph attention has been applied to group-aware crowd navigation [22], self-attention to navigation under a restricted field of view [23], and hybrid self-attention to multi-agent formation control [24].

The most closely related prior work is de Heuvel et al. [4], who apply temporal attention over ICP-registered consecutive LiDAR frames using a Temporal Accumulation Group Descriptor (TAGD). They report an 86.2% success rate in PyBullet corridor, intersection and office environments containing two dynamic and one static pedestrian, modeled as cuboids following A\* paths at speeds sampled from 0.5 to 1.0 m/s, over 1000 evaluation episodes, using DDPG on a Kobuki TurtleBot 2 with an RPLidar A3. PV-STAM is a registration-free alternative that avoids both the ICP overhead and the assumption that pedestrian velocities are known.

## 2.3. Curriculum Learning and Sim-to-Real Transfer

Multi-stage curricula progressing from static to dynamic obstacles have been used by Narvekar et al. [25] and Bengio et al. [26]. Chaffre et al. [9] combine incremental environment complexity with sim-to-real transfer for depth-based navigation. For sim-to-real transfer, prior work has applied domain randomization [27–29], real-world loop closure [30], monocular DRL-based obstacle avoidance [31], transfer methods for soft growing robots [32], and minimal sensor suites on differential-drive platforms [33]. Rather than randomizing physics across all training phases, we define a Hardware-Calibrated Training Mode (HCTM) that applies a fixed actuation delay and velocity noise in the final curriculum phase only.

## 3. System Architecture

Figure 1 shows the system pipeline. Stage 1 performs LiDAR preprocessing. The raw 360° scan is reduced to 24 sectors and enters two branches, one producing the proximity-amplified distance vector  $\hat{s}$  and the other the scan-difference channel  $\Delta s$ . Stage 2 supplies the sector tokens to the PV-STAM attention block, which produces a 48-dimensional sector embedding; this embedding is concatenated with the 6-dimensional navigation state  $o_{\text{nav}}$  and passed to the SAC actor and twin-critic networks, which output continuous velocity commands  $(v_{\text{lin}}, \omega)$ . Stage 3 shows the two validation domains: a Gazebo Classic 11 simulation environment and a physical corridor.

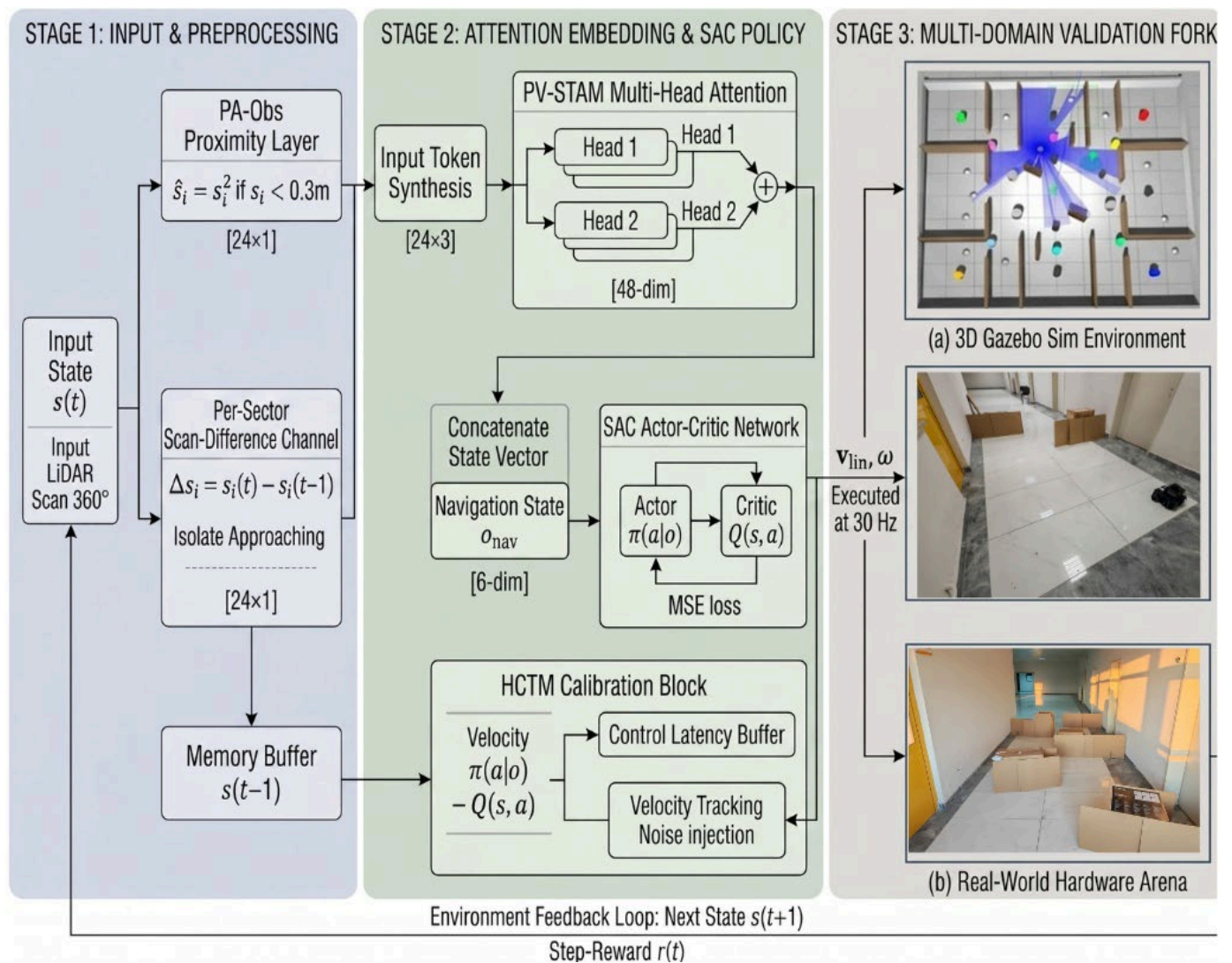


**Figure 1.** System pipeline overview. Stage 1 shows LiDAR preprocessing via proximity amplification and the scan-difference channel  $\Delta s$ , each producing a 24-dimensional output. Stage 2 shows PV-STAM sector attention, followed by state concatenation and the SAC actor–critic network outputting velocity commands. Stage 3 shows the two validation conditions: (a) the Gazebo simulation environment and (b) the real-world hardware arena. Symbols are defined in Section 3.

### 3.1. Observation Space

The environment publishes a 54-element state vector composed of 24 proximity-amplified LiDAR distances  $\hat{s}_i$ , 24 scan-difference values  $\Delta s_i$ , and the 6-element navigation state  $o_{\text{nav}}$ . Each variant selects a subset of this vector. Table 1 states the selection for every variant evaluated in this paper.

Two consequences of Table 1 should be stated directly. First, the per-sector scan-difference channel is supplied to SAC-MLP and SAC-R-PV-STAM only. In SAC-PV-STAM and SAC-PV-STAM-H, the temporal information reaching the attention module consists of three consecutive distance scans, from which the network must derive any motion estimate. Second, the frame-stacked variants are not without approach-rate information: the fifth and sixth elements of  $o_{\text{nav}}$  are the minimum LiDAR reading across all sectors and the rate of change in that minimum, so a scalar summary of  $\Delta s$  is available to every variant. What distinguishes the two groups is the availability of the per-sector signal.

**Table 1.** Observation composition by variant. Frame-stacked variants receive three consecutive distance scans and do not receive the per-sector scan-difference channel.

| Variant       | Observation Contents                                     | Dimension | Per-Sector $\Delta s$ |
|---------------|----------------------------------------------------------|-----------|-----------------------|
| SAC-MLP       | $\hat{s}(24), \Delta s(24), o_{\text{nav}}(6)$           | 54        | Yes                   |
| SAC-MLP-FS    | $\hat{s}(24) \times 3 \text{ frames}, o_{\text{nav}}(6)$ | 78        | No                    |
| SAC-PV-STAM   | $\hat{s}(24) \times 3 \text{ frames}, o_{\text{nav}}(6)$ | 78        | No                    |
| SAC-PV-STAM-H | $\hat{s}(24) \times 3 \text{ frames}, o_{\text{nav}}(6)$ | 78        | No                    |
| SAC-R-PV-STAM | $\hat{s}(24), \Delta s(24), o_{\text{nav}}(6)$           | 54        | Yes                   |
| SAC-LSTM      | $\hat{s}(24), o_{\text{nav}}(6)$                         | 30        | No                    |

Note: The Huber critic variant is evaluated under two critic-width configurations (384 and 256) in Section 6.4, bringing the total evaluated model configurations to seven.

The navigation state  $o_{\text{nav}} \in \mathbb{R}^6$  encodes (1) Euclidean distance to the current goal, normalized by 5.0 m, (2) heading error relative to the goal direction, (3) current linear velocity  $v$ , (4) current angular velocity  $\omega$ , (5) the minimum LiDAR reading across all 24 sectors, and (6) the rate of change in that minimum. The full  $360^\circ$  scan is partitioned into  $N_s = 24$  equally spaced sectors of  $15^\circ$ , with readings clipped to  $[0.12, 3.5]$  m and normalized to  $[0, 1]$ .

The scan-difference channel is susceptible to self-rotation contamination. When the robot rotates, static environmental features sweep across sectors and produce large negative values ( $\Delta s_i < -0.1$ ) resembling the signature of an approaching obstacle. Figure 2 quantifies this effect in both simulation and hardware using identical thresholds. In simulation, the mean number of spurious spikes per timestep is 1.99 during low-rotation intervals ( $|\omega| \leq 0.1$  rad/s,  $n = 20,723$  transitions) and 4.51 during high-rotation intervals ( $|\omega| > 1.0$  rad/s,  $n = 41,590$ ), a ratio of 2.27. On the physical platform the corresponding means are 0.149 ( $n = 27,461$ ) and 1.791 ( $n = 3672$ ), a ratio of 12.0. The contamination is therefore present in both settings and is approximately five times more pronounced on hardware.

Two design choices are intended to limit the effect of this contamination: both linear velocity  $v$  and angular velocity  $\omega$  are included in  $o_{\text{nav}}$ , giving the policy access to its own rotation rate, and attention weights over sectors can in principle down-weight sectors whose signature is inconsistent with a single approaching obstacle. The first of these is supported by the ablation in Section 6.5, in which removing  $\omega$  from  $o_{\text{nav}}$  prevented the policy from progressing past Phase 4. The second is a design argument and is not tested here; attention weights were recorded at deployment for representative scenarios but were not logged jointly with angular velocity, so the specific test—whether sectors are down-weighted during self-rotation—could not be performed.

### 3.2. Scan-Difference Channel

The scan-difference channel is computed by element-wise subtraction of the previous 24-sector scan from the current scan:

$$\Delta s_i = s_i(t) - s_i(t - 1) \quad (1)$$

A negative value indicates that the reading in sector  $i$  decreased between consecutive scans. The computation requires no sensor registration and no calibration and runs in time linear in the number of sectors.

This quantity is a frame-to-frame range difference, not an obstacle velocity. It is not divided by the sampling interval, and it is affected by the robot's own translation and

rotation, by sector discretization, by occlusion, and by changes in the reflecting surface. Where the physical quantity of interest is the rate at which an obstacle closes on the robot along the sensor ray, we use the term “radial closing rate”. Where the quantity is the motion of an obstacle in the world frame, we use obstacle velocity, which this channel does not measure. Division by the measured scan interval converts  $\Delta s$  to a rate. The interval differs between the two settings: 150 ms in simulation and 207 ms on the physical platform (Section 7.5).

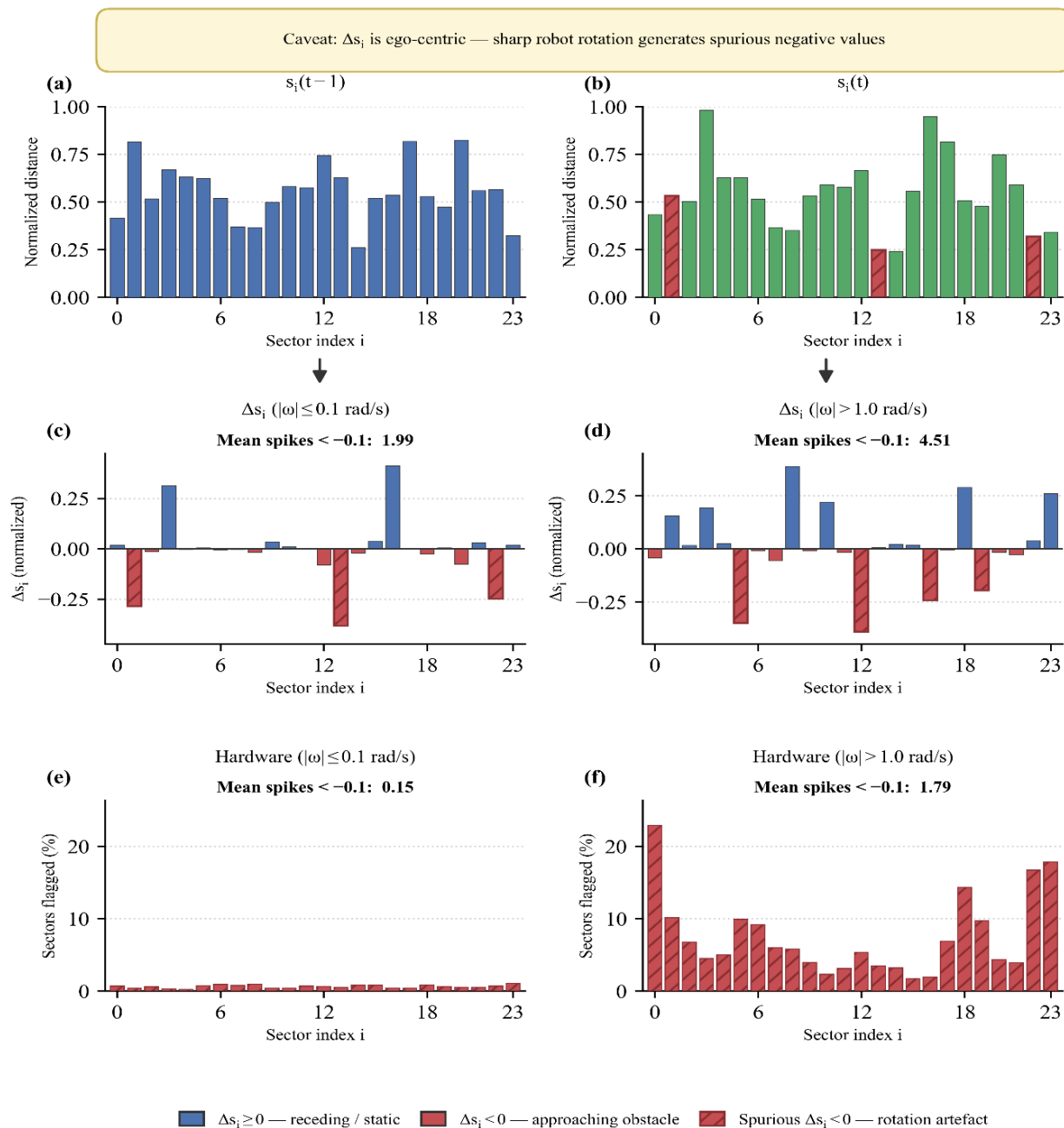


**Figure 2.** Scan-difference channel computation and rotation-contamination analysis. Panels (a,b) show consecutive scans. Panels (c,d) show  $\Delta s$  during low-rotation and high-rotation intervals in simulation: 1.99 against 4.51 spurious spikes per timestep, a ratio of 2.27. Panels (e,f) show the same analysis computed from the 130 physical trials under identical thresholds: 0.149 against 1.791 spurious spikes per control cycle, a ratio of 12.0.

On the physical platform, the previous scan is latched in the sensor callback rather than in the control loop. Between scan arrivals, both the current and previous scans are

unchanged, so  $\Delta s$  retains its last computed value rather than falling to zero. At a measured LiDAR rate of 4.85 Hz against a 30 Hz control loop, 83.9% of control decisions act on a range difference computed from a scan pair already used at least once.

### 3.3. Proximity-Amplified Observation

Before the distance channel is passed to the network, a non-linear amplification is applied within the danger zone (sectors where  $s_i < s_{DZ} = 0.3$  m):

$$\dot{s}_i = s_i^2 \quad \text{if } s_i < 0.3 \text{ m}; \quad \dot{s}_i = s_i \quad \text{otherwise} \quad (2)$$

The squaring compresses near-obstacle readings toward zero, producing a steeper gradient in the region where the distinction between a 15 cm and a 25 cm clearance is behaviorally relevant. The scan-difference channel is passed unmodified.

### 3.4. Action Space and Reward Function

The agent outputs the action  $a_t = (v_{lin}, \omega)$  via a tanh-squashed Gaussian policy, with  $v_{lin} \in [0, 0.26]$  m/s and  $\omega \in [-1.82, 1.82]$  rad/s. Reverse motion is prohibited. The reward at each timestep is

$$r_t = r_{goal} \cdot \Delta d_g - r_{col} \cdot \varphi_{col} - r_{time} \quad (3)$$

where  $\Delta d_g = d_g(t-1) - d_g(t)$  is the goal-distance reduction,  $r_{goal} = 1.5$ ,  $r_{col} = 10.0$  and  $r_{time} = 0.01$ . The proximity term  $\varphi_{col} \in \{0, 1\}$  equals 1 when the minimum LiDAR reading falls below the danger-zone threshold of 0.3 m. Terminal rewards are +100 on reaching the goal and −100 on a threshold violation.

Throughout this paper, “threshold violation” denotes a LiDAR reading below the applicable threshold, and “safety-zone intrusion” denotes entry into the danger zone without episode termination. Neither term implies measured physical contact, which was not instrumented in either setting.

### 3.5. PV-STAM Architecture

PV-STAM applies multi-head self-attention along the sector dimension. Figure 3 shows the module. Each of the 24 sector tokens carries  $C$  input channels, where  $C = 3$  for the frame-stacked variants (three consecutive distance readings for that sector) and  $C = 2$  for the recurrent variant (distance and scan difference). A shared linear projection maps each token from  $C$  to  $d_{model} = 16$  dimensions, and a learned positional encoding of shape (1, 24, 16) is added, giving the module its representation of sector position. A single linear projection without bias maps each 16-dimensional token to 48 dimensions, partitioned into query, key and value matrices of 16 dimensions each. These are divided across two attention heads of 8 dimensions per head, which independently apply scaled dot-product attention:

$$head_k = \text{softmax}\left(\frac{Q_k K_k^T}{\sqrt{d_k}}\right) V_k \quad (4)$$

The two head outputs are concatenated back to 16 dimensions, passed through a linear layer of shape (16, 16) followed by a ReLU activation, then flattened across sectors to a 384-dimensional vector and compressed to the 48-dimensional sector embedding by a single linear layer. The embedding is concatenated with  $o_{nav}$  before being passed to the SAC actor and critic networks.

Table 2 gives the parameter count layer by layer. The module totals 19,968 trainable parameters for the frame-stacked variants and 19,952 for the recurrent variant, the difference arising solely from the input projection.

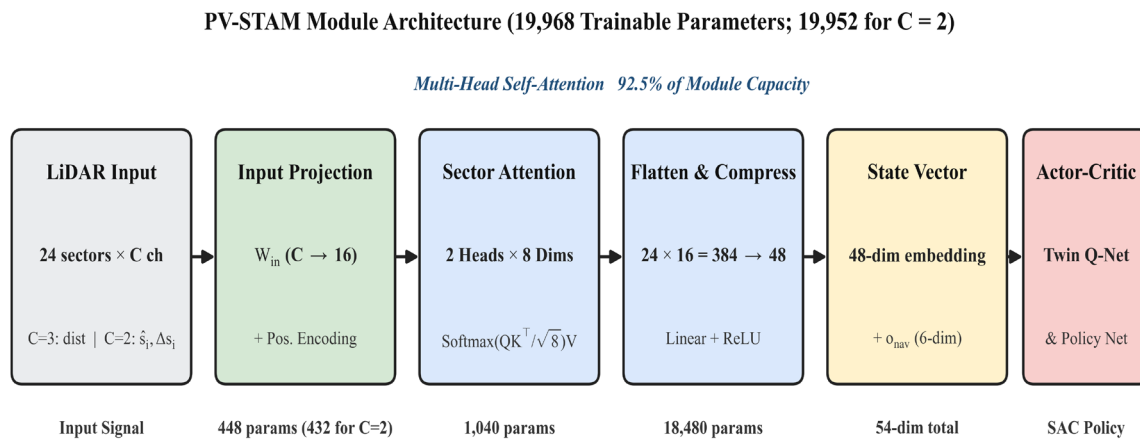


**Figure 3.** PV-STAM module architecture. Each of the 24 sector tokens carries  $C$  channels ( $C = 3$  for frame-stacked variants,  $C = 2$  for the recurrent variant) and is projected to  $d_{\text{model}} = 16$  by a shared linear layer, to which a learned positional encoding is added. A single projection produces  $Q$ ,  $K$  and  $V$  of 16 dimensions each, divided across two heads of 8 dimensions. Head outputs are concatenated to 16 dimensions, passed through a linear layer with ReLU, flattened to 384 dimensions and compressed to the 48-dimensional sector embedding. The compression layer holds 18,480 of the module's 19,968 parameters.

**Table 2.** PV-STAM parameter count by layer, frame-stacked configuration ( $C = 3$ ).

| Layer/Component               | Symbol            | Tensor Shape    | Parameters | Share (%) |
|-------------------------------|-------------------|-----------------|------------|-----------|
| Positional Encoding           | $E_{\text{pos}}$  | $24 \times 16$  | 384        | 1.9%      |
| Input Projection (weight)     | $W_{\text{in}}$   | $16 \times 3$   | 48         | 0.2%      |
| Input Projection (bias)       | $b_{\text{in}}$   | 16              | 16         | 0.1%      |
| QKV Self-Attention Projection | $W_{\text{QKV}}$  | $48 \times 16$  | 768        | 3.8%      |
| Attention Output (weight)     | $W_{\text{out}}$  | $16 \times 16$  | 256        | 1.3%      |
| Attention Output (bias)       | $b_{\text{out}}$  | 16              | 16         | 0.1%      |
| Linear Compression (weight)   | $W_{\text{comp}}$ | $48 \times 384$ | 18,432     | 92.3%     |
| Linear Compression (bias)     | $b_{\text{comp}}$ | 48              | 48         | 0.2%      |
| Total PV-STAM Module          |                   |                 | 19,968     | 100.0%    |

The final compression layer accounts for 92.3% of the module's parameters. A count derived from the attention equations and the projection dimensions alone gives approximately 3.3 K and understates the total by a factor of six. The flatten-and-compress reduction is not equivalent to mean pooling across sectors: mean pooling is permutation-invariant and would discard which sector received attention, whereas the compression layer learns a distinct weight for each sector-feature pair and is permutation-sensitive.

### 3.6. Network Variants

Table 3 specifies the network configurations. SAC-MLP is a two-layer feedforward actor and twin-critic network with no attention and no frame stacking. SAC-MLP-FS adds frame stacking with depth  $k = 3$ . SAC-PV-STAM adds the PV-STAM attention block to the frame-stacked observation. SAC-PV-STAM-H differs from SAC-PV-STAM in two respects: the critic loss is Huber ( $\delta = 10$ ) rather than MSE, and the critic width is 384 rather than 256. A width-matched configuration, differing from SAC-PV-STAM only in the critic loss, is reported in Section 6.4. SAC-R-PV-STAM replaces the frame stack with a GRU encoder of hidden size 64 and an 8-step burn-in period and receives the per-sector scan-difference

channel. SAC-LSTM is a recurrent baseline with 64 hidden units operating on distance and navigation state only.

**Table 3.** Network variant specifications and trainable parameter counts.

| Acronym             | Descriptive Name                    | Temporal Mechanism          | Perception Filter                | Critic Loss & Width            | Total Params | Critic Params | Total     |
|---------------------|-------------------------------------|-----------------------------|----------------------------------|--------------------------------|--------------|---------------|-----------|
| SAC-MLP             | Single-Frame Baseline               | None (1 frame)              | None (2-layer MLP)               | MSE, 256-d                     | 242,290      | 161,282       | 242,290   |
| SAC-MLP-FS          | Frame-Stacked Baseline              | Frame Stack ( $k = 3$ )     | None (2-layer MLP)               | MSE, 256-d                     | 655,522      | 436,738       | 655,522   |
| SAC-PV-STAM         | Positional-Velocity STAM (Proposed) | Frame Stack ( $k = 3$ )     | 2-Head STAM (48-d Bottleneck)    | MSE, 256-d                     | 696,946      | 464,386       | 696,946   |
| SAC-PV-STAM-H (384) | Uncompressed Huber Variant          | Frame Stack ( $k = 3$ )     | 2-Head STAM (384-d Uncompressed) | Huber ( $\delta = 10$ ), 384-d | 1,204,082    | 971,522       | 1,204,082 |
| SAC-PV-STAM-H (256) | Width-Matched Huber Variant         | Frame Stack ( $k = 3$ )     | 2-Head STAM (256-d Bottleneck)   | Huber ( $\delta = 10$ ), 256-d | 696,946      | 464,386       | 696,946   |
| SAC-R-PV-STAM       | Recurrent STAM (Hardware)           | GRU ( $h = 64$ , burn-in 8) | 2-Head STAM + Scan-Diff          | MSE, 256-d                     | 2,288,450    | 1,858,530     | 2,288,450 |
| SAC-LSTM            | Recurrent Baseline                  | LSTM ( $h = 64$ )           | None (Distance & Nav State)      | MSE, 256-d                     | 106,626      | 71,042        | 106,626   |

The variants are not parameter-matched. SAC-R-PV-STAM has 3.3 times the total parameter count of SAC-PV-STAM, arising from the GRU encoder and the associated critic width. This should be taken into account when interpreting the comparison between them in Sections 6 and 7. The PV-STAM module itself contributes 19,968 parameters, less than 3% of the total in either case.

#### 4. Curriculum Learning

Training follows the seven-phase schedule in Table 4, illustrated in Figure 4. Phases 1 to 4 contain no obstacles and progressively widen the goal distribution from a maximum radius of 2.0 m to the full 23-goal layout spanning the  $8 \times 8$  m training arena. Phases 5 and 6 introduce dynamic cylindrical obstacles: 6 obstacles at 0.08 m/s in Phase 5 and 9 at 0.12 m/s in Phase 6. Phase 7 uses 15 simultaneously moving obstacles at 0.18 m/s.

Advancement is governed by the rolling episode success rate (ESR) over the last  $n$  episodes. The curriculum also steps backward: when the rolling ESR falls more than 10 percentage points below the current phase threshold, the agent is demoted to the previous phase. Phase 7 additionally activates the Hardware-Calibrated Training Mode described in Section 5.2.

The SAC-LSTM baseline reported in Section 6.3 was advanced by a fixed environment-step budget per phase rather than by the ESR threshold. This modification was made because under ESR-gated advancement the baseline did not progress beyond Phase 4, which contains no obstacles, and could therefore not be evaluated on the capability under test.

**Table 4.** Curriculum phase specifications. ESR is the rolling episode success rate over  $n$  episodes.

| Phase | Goals          | Obstacles | Speed (m/s) | Notes                        | ESR Threshold | $n$ |
|-------|----------------|-----------|-------------|------------------------------|---------------|-----|
| 1     | $d \leq 2.0$ m | 0         | —           | static, goal-seeking only    | $\geq 70\%$   | 50  |
| 2     | $d \leq 2.5$ m | 0         | —           | wider goal radius            | $\geq 70\%$   | 50  |
| 3     | $d \leq 3.0$ m | 0         | —           | full goal radius             | $\geq 65\%$   | 50  |
| 4     | Full (23)      | 0         | —           | full 23-goal layout          | $\geq 60\%$   | 120 |
| 5     | Full (23)      | 6         | 0.08        | dynamic obstacles introduced | $\geq 60\%$   | 50  |
| 6     | Full (23)      | 9         | 0.12        | higher density               | $\geq 65\%$   | 100 |
| 7     | Full (23)      | 15        | 0.18        | terminal phase, HCTM active  | terminal      | —   |

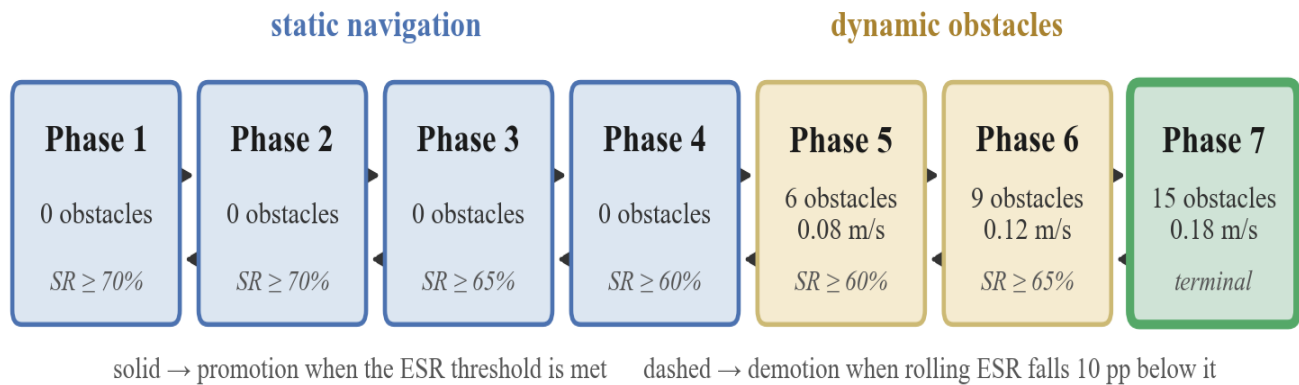


**Figure 4.** Seven-phase progressive curriculum. Phases 1–4 are obstacle-free with expanding goal radii. Phases 5–6 introduce dynamic obstacles. Phase 7 is the terminal phase. ESR thresholds are shown above each transition. Upper arrows (→) indicate promotion when the ESR threshold is met; lower arrows (←) indicate demotion when the rolling ESR falls 10 percentage points below it.

## 5. Experimental Setup

### 5.1. Simulation Environment

Training and simulated evaluation were performed in Gazebo Classic 11 under ROS 2 Humble Hawksbill on Ubuntu 22.04 LTS, with all networks implemented in PyTorch ( $\geq 2.0$ ) on Python 3.10. The simulated robot is a TurtleBot3 Waffle Pi (ROBOTIS Co., Ltd., Seoul, Republic of Korea) with differential-drive kinematics. Dynamic obstacles are rigid cylinders of radius 0.2 m following sinusoidal trajectories with randomized phase offsets. The training arena measures approximately  $8 \times 8$  m and contains rectangular wall partitions dividing the space into corridors and open zones, with 23 candidate goal positions. The partitions require the robot to pass through apertures as narrow as 0.6 m while avoiding up to 15 moving cylinders.

Three benchmark arenas sharing no geometry with the training environment were built to test zero-shot generalization (Figure 5). Benchmark A is a bare  $5 \times 5$  m space with four static cylinders placed symmetrically. Benchmark B retains the  $5 \times 5$  m footprint, adds internal wall partitions, and sets the same four cylinders on oscillating paths. Benchmark C is the TurtleBot3 World map, whose bilateral corridor geometry does not appear in training.

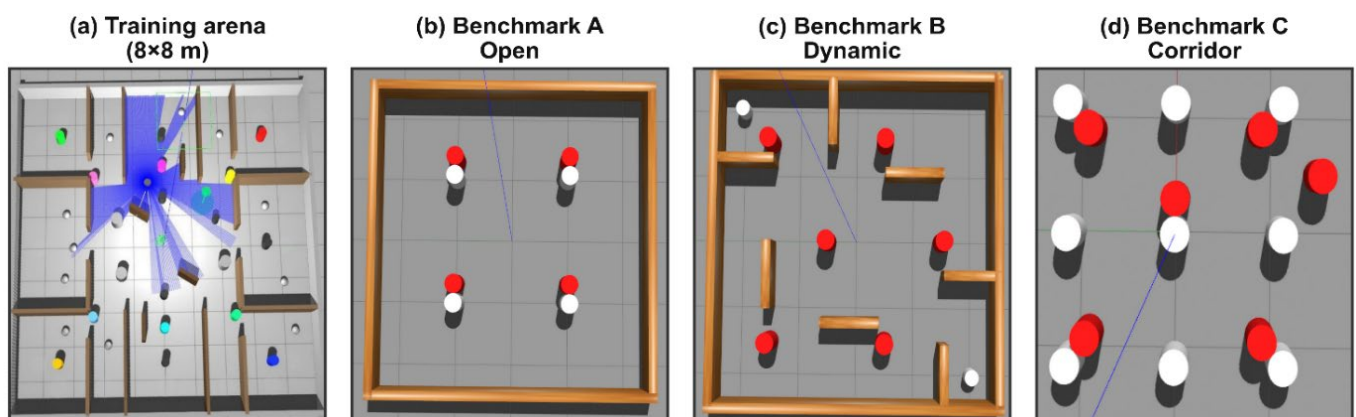


**Figure 5.** Simulation environments. (a) The  $8 \times 8$  m Gazebo Classic 11 training arena with internal wall partitions, dynamic obstacles and the 23 candidate goal positions. (b–d) The three zero-shot evaluation benchmarks, none seen during training: (b) Benchmark A, open arena with static obstacles; (c) Benchmark B, same layout with dynamic obstacles; (d) Benchmark C, TurtleBot3 World corridor geometry.

The simulation control period is 150 ms, corresponding to a policy rate of 6.67 Hz. This differs from the rate used during physical deployment and is addressed in Section 7.5.

### 5.2. Hardware-Calibrated Training Mode

HCTM is active in Phase 7 only. Figure 6 shows the pipeline. The policy emits an action  $a_t = (v_{lin}, \omega)$  at each step, which passes through a calibration block before reaching the simulator. A 65 ms FIFO latency buffer delays the command, and a state-noise model adds zero-mean Gaussian velocity noise with standard deviation  $\sigma = 0.02$  m/s to the simulator state. Restricting both perturbations to the final phase leaves the earlier phases unperturbed.

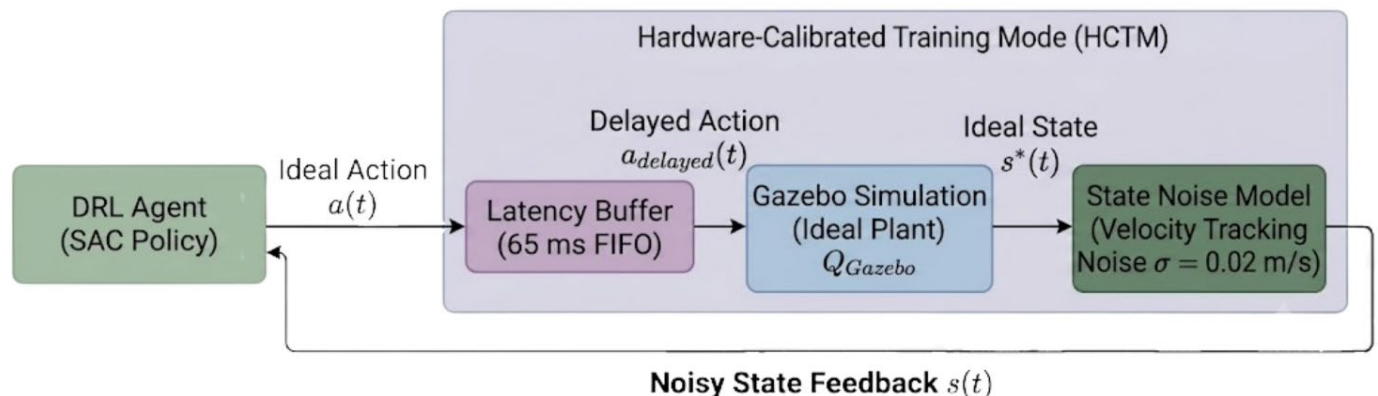


**Figure 6.** HCTM pipeline. The agent outputs action  $a_t$  which passes through a 65 ms FIFO latency buffer before being applied to the Gazebo simulator. The simulator state is corrupted by Gaussian velocity noise with  $\sigma = 0.02$  m/s. This pipeline is active only during Phase 7.  $s^*(t)$  denotes the ideal, noise-free simulator state before velocity-tracking noise is applied.

The two parameter values used in HCTM were set during earlier development of this platform. Their derivation is not documented. An independent measurement of commanded-to-observed velocity lag was made during the present study by cross-correlating the commanded velocity against the odometry-reported velocity across 70 physical trials. The mean lag was 152.8 ms (median 155 ms, SD 8.7,  $n = 40$  trials) for SAC-R-PV-STAM and 151.3 ms (median 150 ms, SD 7.5,  $n = 30$ ) for SAC-PV-STAM. The value is policy-independent and is therefore a property of the platform rather than of any controller. This quantity aggregates actuator response, odometry reporting delay and inertial ramp-up, and is not directly comparable to the actuation delay modeled in HCTM, but it indicates that the 65 ms figure understates the total commanded-to-observed lag of this platform by a factor of approximately 2.3. Replay buffer sampling follows uniform experience storage; advanced prioritized experience replay [34] is reserved for future work.

**Observation convention for the deployed policies.** The seed-777 policies deployed in Section 7 use the previous commanded action in observation slots 50 and 51 rather than the odometry-reported velocity. A deployment flag reproduces this observation layout at inference time; without it, the policy receives an observation inconsistent with its training. This is an observation-format convention. The deployed checkpoints, trained between 23 and 26 April 2026, predate the introduction of the sim-to-real adaptation code into the SAC training environment on 16 June 2026 and were therefore not trained with additional sensor-noise augmentation, modified reward shaping or a modified control period.

### 5.3. Training Hyperparameters

Table 5 summarizes the training hyperparameters. All variants share the replay buffer size (200 K transitions), actor learning rate ( $3 \times 10^{-4}$ ) and discount factor ( $\gamma = 0.99$ ). The

recurrent variants use a batch size of 64 sequences to accommodate the 16-step sequence length plus 8-step burn-in.

**Table 5.** SAC hyperparameters for each network variant.

| Hyperparameter    | SAC-MLP            | SAC-MLP-FS         | SAC-PV-STAM        | SAC-PV-STAM-H      | SAC-R-PV-STAM      | SAC-LSTM           |
|-------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Replay buffer     | 200 K              | 200 K              | 200 K              | 200 K              | 200 K              | 200 K              |
| Batch size        | 256                | 256                | 256                | 256                | 64 (seq)           | 64 (seq)           |
| Sequence length   | —                  | —                  | —                  | —                  | 16 + 8 burn-in     | 16 + 8 burn-in     |
| N-step returns    | 1                  | 3                  | 3                  | 3                  | 1                  | 1                  |
| Actor LR          | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| Discount $\gamma$ | 0.99               | 0.99               | 0.99               | 0.99               | 0.99               | 0.99               |
| Target entropy    | −2                 | −2                 | −2 (see note)      | −2                 | −2                 | −2                 |

Note on the target entropy. In SAC-PV-STAM, the SAC entropy target is explicitly reset to −2.0 on entry to Phase 7. Neither SAC-PV-STAM-H configuration (critic width 384 or 256) contains this reset. The comparison between SAC-PV-STAM and SAC-PV-STAM-H in Section 6.4 therefore differs in this respect in addition to the critic loss, and this is stated as a limitation in Section 8.3.

#### 5.4. Evaluation Protocol

**Curriculum-phase evaluation.** After curriculum completion, each variant was evaluated in deterministic mode ( $\alpha = 0$ , mean action output) for 200 episodes per dynamic phase (5, 6 and 7) in a fixed-seed environment.

**Canonical benchmark evaluation.** Each variant was evaluated on all three benchmark arenas for 100 episodes per seed across three seeds, giving 300 episodes per variant per benchmark. Three random seeds were used: 42, 777 and 123, following the multi-seed evaluation recommendations of Agarwal et al. [35]. All cells in the canonical evaluation contain the full three seeds at 100 episodes each.

**Statistical methods.** Success rates are reported with Wilson score 95% confidence intervals. For the physical evaluation, the two scenarios were not sampled in equal proportion for every variant, so a pooled rate is a weighted average with variant-dependent weights, and pooled comparisons are confounded by scenario composition. Comparisons are therefore made within scenarios using Fisher’s exact test and across scenarios using the Cochran–Mantel–Haenszel test stratified by scenario, with Mantel–Haenszel common odds ratios and Robins–Breslow–Greenland confidence intervals. Where a single summary rate is quoted, it is directly standardized with equal weight on the two scenarios. All tests are two-sided.

## 6. Simulation Results

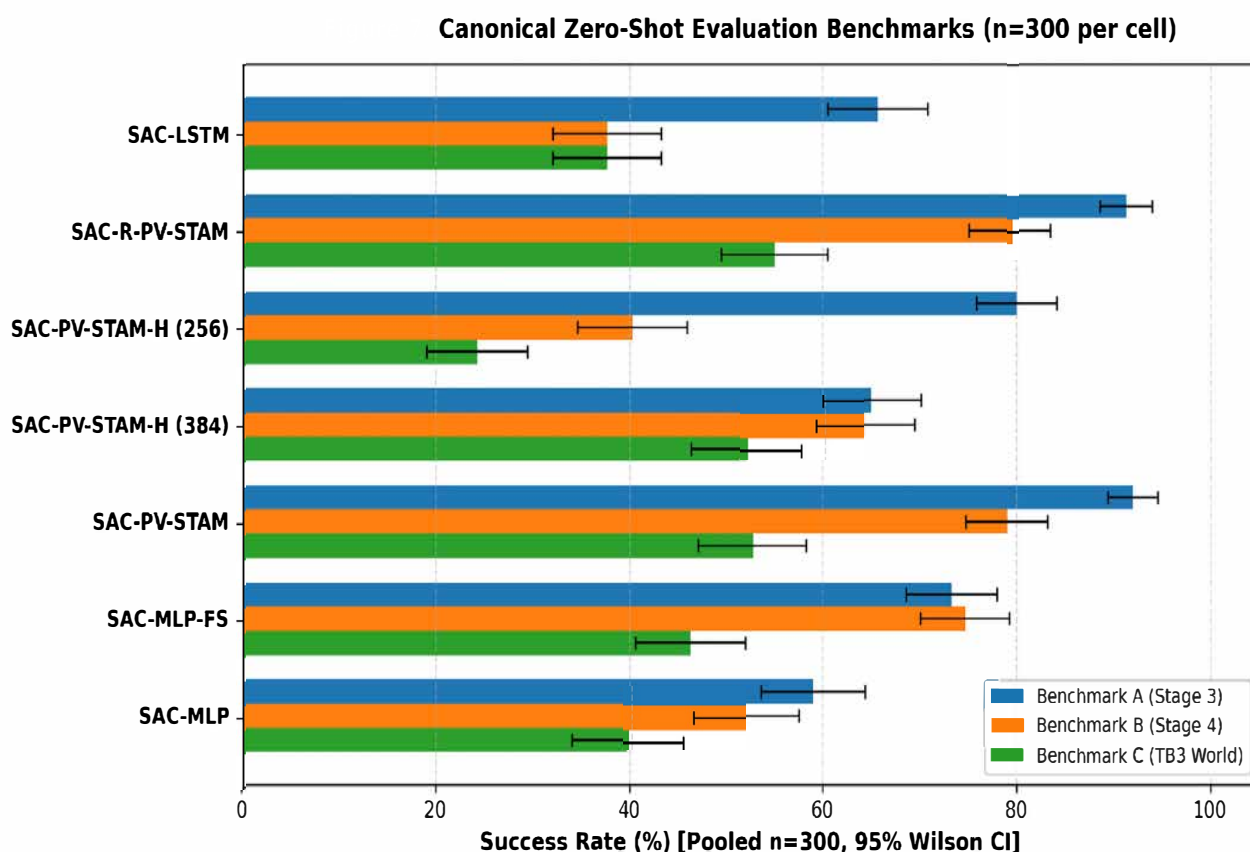
### 6.1. Canonical Benchmark Evaluation

Table 6 reports the canonical evaluation. Every cell pools three seeds at 100 episodes each. Figure 7 shows the same results with Wilson score 95% confidence intervals.

SAC-PV-STAM, SAC-R-PV-STAM and SAC-MLP-FS lie within 4.6 percentage points of one another on Benchmarks A and B, and SAC-MLP-FS leads on Benchmark C by 6.0 points over SAC-R-PV-STAM. No ordering among these three variants is supported by these data. The benchmarks do separate the remaining variants: SAC-MLP, SAC-LSTM and both SAC-PV-STAM-H configurations are distinguishable from the leading group in all three arenas.

**Table 6.** Zero-shot benchmark success rates, 300 episodes per variant per benchmark, three seeds pooled.

| Variant             | Benchmark A (Open) | Benchmark B (Dynamic) | Benchmark C (Corridor) | Seeds |
|---------------------|--------------------|-----------------------|------------------------|-------|
| SAC-PV-STAM         | 92.0%              | 79.0%                 | 52.7%                  | 3     |
| SAC-R-PV-STAM       | 91.3%              | 79.3%                 | 55.0%                  | 3     |
| SAC-MLP-FS          | 88.0%              | 74.7%                 | 61.0%                  | 3     |
| SAC-PV-STAM-H (384) | 65.0%              | 64.3%                 | 52.0%                  | 3     |
| SAC-PV-STAM-H (256) | 80.0%              | 40.3%                 | 24.3%                  | 3     |
| SAC-LSTM            | 65.7%              | 37.7%                 | 37.7%                  | 3     |
| SAC-MLP             | 59.0%              | 52.0%                 | 39.7%                  | 3     |

**Figure 7.** Canonical evaluation across the three benchmark arenas, 100 episodes per seed and three seeds per variant. Error bars are Wilson score 95% confidence intervals.

## 6.2. Curriculum-Phase Evaluation

Table 7 reports Phase 5 and Phase 6 performance, and Table 8 reports the Phase 7 results across all three seeds. Standard deviations are population standard deviations. Figure 8 shows the seed-42 Phase 7 episode outcomes and navigation efficiency.

**Table 7.** Intermediate curriculum performance, Phases 5 and 6, seed 42,  $n = 200$  episodes per phase. TVR denotes threshold-violation rate.

| Variant       | Ph5 SR | Ph5 TVR | Ph6 SR | Ph6 TVR |
|---------------|--------|---------|--------|---------|
| SAC-MLP       | 77.0%  | 22.0%   | 73.5%  | 25.5%   |
| SAC-MLP-FS    | 85.0%  | 15.0%   | 84.0%  | 16.0%   |
| SAC-PV-STAM   | 79.0%  | 11.5%   | 77.5%  | 15.0%   |
| SAC-PV-STAM-H | 59.0%  | 4.0%    | 58.5%  | 3.5%    |
| SAC-R-PV-STAM | 83.0%  | 17.0%   | 79.5%  | 20.5%   |

**Table 8.** Phase 7 performance. A dash indicates that the seed did not converge to Phase 7. The double dagger indicates a seed that reached Phase 7 with severely degraded performance.

| Variant       | SR s42 | TVR s42 | SR s777 | SR s123 | Mean SR     | Mean TVR     |
|---------------|--------|---------|---------|---------|-------------|--------------|
| SAC-MLP       | 37.5%  | 62.5%   | 21.5% ‡ | —       | —           | —            |
| SAC-MLP-FS    | 53.0%  | 47.0%   | 60.5%   | 63.5%   | 59.0 ± 4.4% | 41.0 ± 4.4%  |
| SAC-PV-STAM   | 66.0%  | 27.5%   | 62.5%   | 48.0%   | 58.8 ± 7.8% | 38.8 ± 9.8%  |
| SAC-PV-STAM-H | 56.5%  | 21.0%   | 51.5%   | —       | 54.0 ± 2.5% | 34.8 ± 13.8% |
| SAC-R-PV-STAM | 68.5%  | 31.5%   | 69.5%   | 54.0%   | 64.0 ± 7.1% | 35.8 ± 6.8%  |

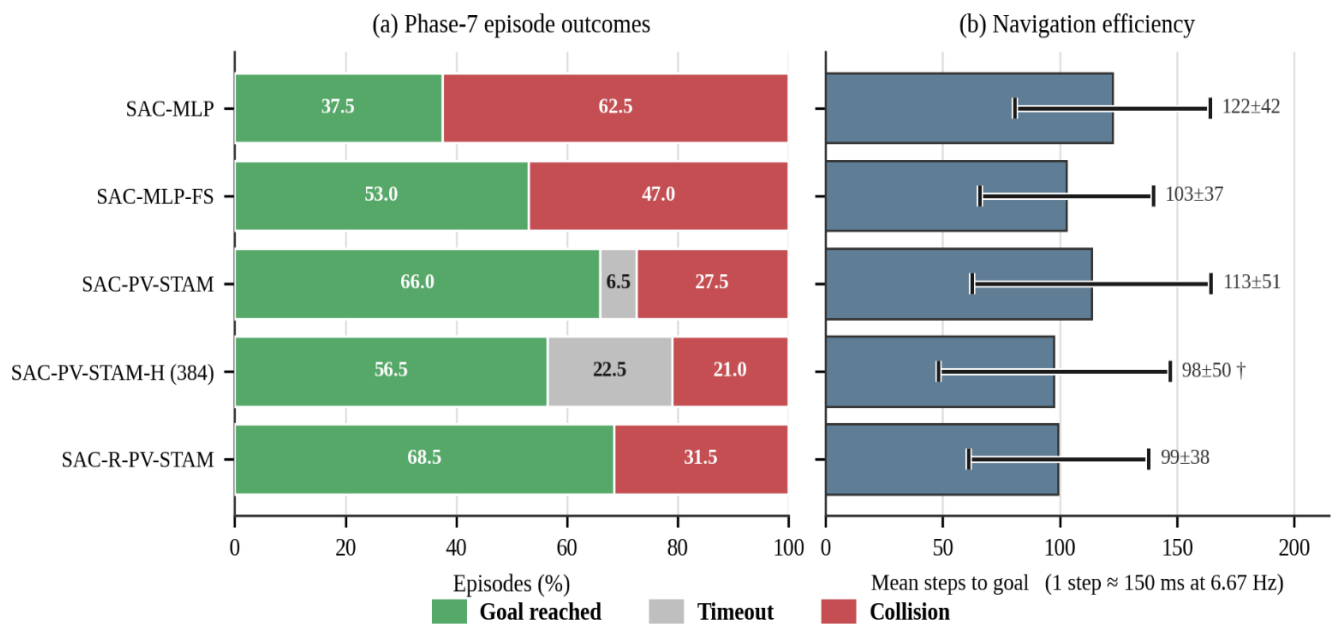


**Figure 8.** Phase 7 navigation quality. (a) Episode outcomes at seed 42,  $n = 200$  episodes per variant; collision denotes a threshold violation. (b) Mean steps to goal over successful episodes, converged seeds pooled; error bars  $\pm 1$  standard deviation. † The lower mean for SAC-PV-STAM-H reflects selection bias toward shorter successful episodes, not greater efficiency.

SAC-R-PV-STAM obtained the highest mean Phase 7 success rate at  $64.0 \pm 7.1\%$ , ahead of SAC-PV-STAM at  $58.8 \pm 7.8\%$  and SAC-MLP-FS at  $59.0 \pm 4.4\%$ . The differences between these three means are smaller than their standard deviations across seeds and no ordering is claimed. SAC-MLP-FS and SAC-PV-STAM reached statistically indistinguishable mean success rates, with a difference of 2.2 percentage points in mean threshold-violation rate.

Two protocol issues in these curriculum-phase evaluations should be recorded.

First, the seed-42 Phase 7 evaluations were not run over a common goal set. Four of the five variants were evaluated over the same 23 goals; SAC-MLP-FS was evaluated over 22, including five goals that no other variant received and omitting six that all others received. Restricted to the 17 goals common to all five variants, SAC-PV-STAM obtained a 70.7% success rate and a 27.3% threshold-violation rate against 56.1% and 43.9% for SAC-MLP-FS, a difference of 16.6 percentage points in violation rate rather than the 19.5 points obtained over the unmatched sets. Mean goal distance was similar across the two sets (2.97 m against 3.06 m). Both figures are reported here; the matched figure is the one that supports a comparison.

Second, two variants did not converge to Phase 7 under all seeds. SAC-MLP reached Phase 7 under seed 777 with 21.5% success and did not converge beyond Phase 1 under seed 123; SAC-PV-STAM-H did not converge under seed 123. These outcomes are reported in Table 8 rather than omitted, and the means for those rows are computed over converged

seeds only, which is stated in the table. Because this aggregation excludes non-converged seeds post hoc, the reported means for these two rows should be interpreted as descriptive.

### 6.3. Recurrent Baseline

To separate the contribution of the scan-difference channel from that of recurrent temporal encoding, a SAC-LSTM baseline with 64 hidden units was trained on distance and navigation state only, without the per-sector scan-difference channel and without attention.

Under ESR-gated advancement, this baseline did not progress beyond Phase 4, which contains no obstacles, and therefore never trained on the capability under evaluation. The results reported here use fixed step-budget advancement per phase, guaranteeing exposure to Phases 5 to 7, across three seeds, each trained for 3000 episodes. Under this protocol, the baseline reached 65.7%, 37.7% and 37.7% on Benchmarks A, B and C, respectively. SAC-R-PV-STAM outperformed it significantly on all three benchmarks (Fisher exact,  $p = 9.1 \times 10^{-15}$ ,  $1.4 \times 10^{-25}$  and  $2.9 \times 10^{-5}$ ).

Two properties of this baseline are reported beyond the pooled means.

The baseline is unstable across seeds. Final training success rates across the three seeds were 11%, 18% and 26%, and overall goal rates were 13.7%, 14.8% and 28.7%. For comparison, the width-matched SAC-PV-STAM-H configuration trained under identical conditions produced three seeds within five points of one another (43%, 44% and 48%). A pooled mean conceals this spread, and single-seed comparisons against this baseline would not be reliable.

The baseline times out rather than colliding. On Benchmark A, it timed out on 19.3% of episodes, the highest rate among all variants evaluated, whereas the remaining variants are near zero. The behavior is the same class as that documented on the physical platform in Section 7.4 for the non-recurrent variants.

We conclude that recurrent memory added to a distance-only observation was not sufficient in these experiments. We do not claim that recurrence is insufficient in principle or independently of architecture; one recurrent baseline was tested under one training protocol.

### 6.4. Huber Critic Loss

The SAC-PV-STAM-H configuration differs from SAC-PV-STAM in two variables: the critic loss function and the critic width, which is 384 rather than 256. A width-matched configuration, trained across three seeds, separates them. Table 9 gives the result.

**Table 9.** Effect of matching critic width to the SAC-PV-STAM baseline. Both configurations use Huber loss with  $\delta = 10$ .

| Benchmark    | Critic Width 384 | Critic Width 256 | Difference | $p$                 |
|--------------|------------------|------------------|------------|---------------------|
| A (open)     | 65.0%            | 80.0%            | +15.0      | $5 \times 10^{-5}$  |
| B (dynamic)  | 64.3%            | 40.3%            | −24.0      | $7 \times 10^{-9}$  |
| C (corridor) | 52.0%            | 24.3%            | −27.7      | $3 \times 10^{-12}$ |

Reducing the critic to the baseline width improved success on the simplest benchmark by 15.0 points and degraded it by 24.0 and 27.7 points on the two harder benchmarks. The additional critic capacity was therefore environment-dependent in effect rather than generally advantageous. The negative result for Huber regularization holds in the sense that both configurations remain below SAC-PV-STAM and SAC-R-PV-STAM on all three benchmarks, but its magnitude cannot be attributed to the loss function alone.

We did not collect TD-error distributions, Q-value distributions, action entropies or stopping statistics, so no mechanistic account of the Huber effect is offered here. The description in this section is behavioral.

### 6.5. Angular Velocity Ablation

The navigation state includes angular velocity  $\omega$ . Because the LiDAR scan is rotationally symmetric about the robot frame,  $\omega$  might be redundant with the scan pattern. A variant with  $\omega$  removed from  $o_{\text{nav}}$  was trained with all other hyperparameters and the curriculum identical to SAC-PV-STAM.

Without  $\omega$  the ablated agent required 1440 episodes to exit Phase 1 against 175 for the full model and did not enter Phase 5 within 4000 training episodes (Figure 9). On Benchmark B at seed 42 over 218 episodes, the ablated agent obtained 13.8% success with an 86.2% threshold-violation rate against 74.0% and 26.0% for SAC-PV-STAM. Restricted to the first 200 episodes for consistency with the other benchmark evaluations, the figures are 12.5% and 87.5%.

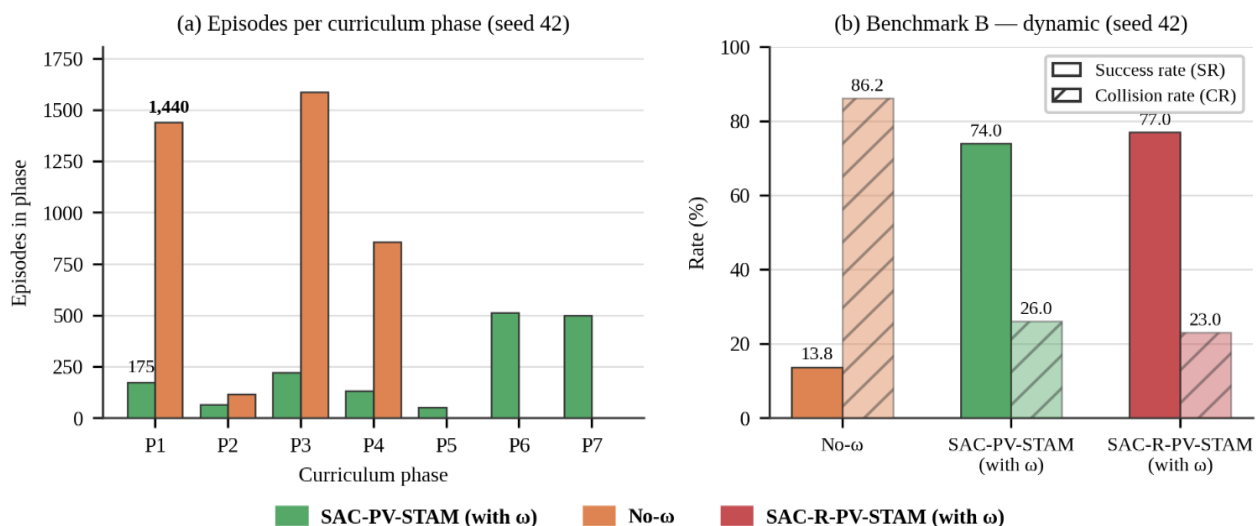


**Figure 9.** Angular-velocity ablation, seed 42. (a) Episodes spent in each curriculum phase for the ablated agent against the full model: without  $\omega$ , Phase 1 requires 1440 episodes against 175, and the agent never reaches Phase 5. (b) Benchmark B success rate and threshold-violation rate.

Angular velocity is therefore not redundant with the rotationally symmetric scan. Without ego-rotation feedback, the agent did not correct accumulated heading error when navigating to far-field or corner-positioned goals.

### 6.6. Comparison with Prior Work

SAC-R-PV-STAM was evaluated in a  $12 \times 8$  m open arena containing 15 Social Force Model pedestrians over 200 deterministic episodes, obtaining 91.5% success, a 1.5% threshold-violation rate and 7.0% timeouts.

This is an indicative cross-study comparison and not a matched benchmark. de Heuvel et al. [4] report 86.2% success under conditions that differ from ours in several respects, summarized in Table 10. We make no claim of a performance advantage over their method. The algorithmic distinction is independent of the performance figures: their method applies ICP point-cloud registration at every control cycle, whereas the scan-difference channel used here is a single element-wise subtraction with no registration.

**Table 10.** Evaluation conditions in this work and in de Heuvel et al. [4].

| Dimension           | This Work                      | de Heuvel et al. [4]                                        |
|---------------------|--------------------------------|-------------------------------------------------------------|
| Arena               | 12 × 8 m open                  | PyBullet corridor (6–8 m × 2.0–2.5 m), intersection, office |
| Obstacle count      | 15 simultaneous                | 2 dynamic, 1 static                                         |
| Obstacle model      | Social Force Model pedestrians | A*-following cuboids, no inter-agent avoidance              |
| Obstacle speed      | 0.18 m/s training ceiling      | 0.5–1.0 m/s sampled                                         |
| Base algorithm      | SAC                            | DDPG                                                        |
| Platform            | TurtleBot3 Waffle Pi, LDS-01   | Kobuki TurtleBot 2, RPlidar A3                              |
| Registration        | none                           | ICP at every control cycle                                  |
| Evaluation episodes | 200 deterministic, 3 seeds     | 1000                                                        |

### 6.7. Corridor Fine-Tuning

To test whether Benchmark C performance can be improved by fine-tuning, the SAC-R-PV-STAM seed-42 checkpoint was fine-tuned for 50,000 additional training steps in the corridor environment, starting from Phase 1 of a corridor-specific curriculum. Performance decreased: success fell from 48.0% to 36.5%, and the threshold-violation rate rose from 50.0% to 63.5%. Throughout the 50,000 steps, the agent remained in Phase 1 with a rolling success rate of approximately 18%.

This establishes a lower bound on the fine-tuning budget required for structural domain transfer from an open arena to bilateral corridor geometry. It also indicates that sequential fine-tuning requires the policy to overwrite a converged behavior before acquiring the new one. Interleaving corridor geometry into the curriculum from the start would not impose this requirement. That alternative was not tested and remains open.

## 7. Real-Robot Evaluation

### 7.1. Hardware Platform

The platform is a TurtleBot3 Waffle Pi differential-drive robot (ROBOTIS Co., Ltd., Seoul, Republic of Korea) (Figure 10). Its LDS-01 2D LiDAR (ROBOTIS) returns a 360° planar scan over a 0.12–3.5 m range, with the range normalized in simulation. An NVIDIA Jetson AGX Orin (NVIDIA Corporation, Santa Clara, CA, USA) runs the trained SAC actor under ROS 2 Humble. An OpenCR 1.0 board (ROBOTIS) drives the wheels and streams the IMU at 200 Hz, and an onboard power bank supplies the stack.

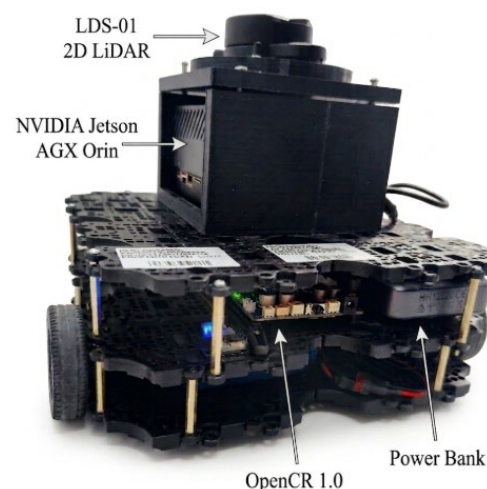


**Figure 10.** TurtleBot3 Waffle Pi hardware platform with component annotations: LDS-01 2D LiDAR, NVIDIA Jetson AGX Orin for onboard inference, OpenCR 1.0 embedded controller, and onboard power bank.

Actor inference latency was measured over 1000 timed forward passes per variant after 100 warm-up passes, with device synchronization around each timing. Mean latency ranged from 0.201 ms for SAC-MLP to 0.725 ms for SAC-R-PV-STAM, with a maximum 99th percentile of 1.416 ms. The largest of these values occupies 4.3% of the 33.3 ms control period used at deployment. Inference latency was therefore not a limiting factor for any variant. These measurements were taken on a laptop GPU rather than on the Jetson, so they establish that the networks are small relative to the control period rather than characterizing the deployed device.

## 7.2. Experimental Design

Two scenarios were run in a 3.3 m × 7.6 m indoor corridor with a marble floor (Figure 11). Scenario 2 placed two cardboard obstacles between the start pose and the goal. Scenario 3 retained those two obstacles in the same positions and added a third cardboard obstacle drawn across the robot's path. Holding the static layout fixed makes Scenario 2 a control condition for Scenario 3 so that any difference between them is attributable to obstacle motion. Integrating full 3D visual-LiDAR odometry [36] represents a promising extension for complex unmapped terrain.



**Figure 11.** Physical evaluation corridor, 3.3 × 7.6 m. (a) Corridor overview with the TurtleBot3 Waffle Pi at the start mark. (b) Cardboard obstacle configuration. The Scenario 2 and Scenario 3 layouts are specified in Section 7.2. Obstacles are cardboard throughout; no human participants took part in any trial.

Odometry was zeroed at the start mark before every trial, so each trial uses a coordinate frame whose origin is the robot's starting pose, with the x-axis directed along the corridor toward the goal and the y-axis lateral. The goal was a single fixed target placed 4.5 m along x from this origin; 4.5 m is therefore the goal distance, not a range limit or a maximum traversable distance. A trial was counted as successful when the robot entered a 0.40 m acceptance radius about that target and terminated on goal acceptance, on a LiDAR reading below 0.20 m, or after 4500 control steps (150 s).

Goal distances were capped at 5.0 m by design. The network receives goal distance normalized by 5.0 m, and training goals did not exceed 4.4 m, so a longer goal produces an input value outside the training range. This was verified on the platform: with a 9 m goal, corresponding to a normalized input of 1.80, every variant rotated in place without translating.

The moving obstacle was at least 0.30 m tall so as to intersect the LiDAR plane at its mounting height of 0.18 m. It was drawn perpendicular to the robot's path at approximately 2.5 m from the start mark and traveled approximately 1.5 m at approximately 0.2 m/s.

It was pulled by a line from beside the corridor rather than pushed by hand, so that the operator did not enter the sensor field at a varying distance. The obstacle speed was chosen to lie within the 0.08–0.18 m/s range used in the curriculum; a pedestrian walking at 0.5–0.8 m/s would have tested out-of-distribution generalization rather than the trained capability.

All variants were evaluated at seed 777 using the observation convention described in Section 5.2.

### 7.3. Data Validity

Trial validity was verified from the recordings after collection. Three checks were applied to every trial.

First, the command publication rate was measured in each recording. A single policy node publishes at 30 Hz; two concurrent nodes produce approximately 60 Hz, and the robot executes interleaved commands from two policies. This identified five affected trials at 59–60 Hz against 30.0–30.2 Hz for all others.

Second, a 24-sector LiDAR summary was extracted from the first scan of every trial and compared within each collection block to confirm that obstacle layout and start pose had not drifted. This detected an unrecorded change to the physical arena during the campaign: success fell from 16 of 16 to 0 of 4 at a specific time boundary, and a different variant subsequently failed at the same location ( $x \approx 2.6$ – $3.0$  m). Concurrent controllers, battery state (12.1–12.2 V), stale device handles, sensor health (4.98 Hz, no dropouts), and start geometry were each excluded by measurement before the arena was identified as the cause. The layout was restored and verified with three control trials before collection resumed.

Third, for Scenario 3, the crossing event was located in each recording to confirm that the encounter occurred while the robot was approaching.

Of 143 recorded trials, 130 were analyzed and 13 excluded: five for the concurrent-node fault, four following the arena change, one in which the operator contacted the robot after it had passed the obstacles, and three in which the intended crossing did not occur. The exclusion criteria are properties of the experimental conditions rather than of the outcomes and were applied strictly prior to outcome analysis. All 13 excluded trials resulted in failures. Exclusion categories are recorded per trial, and all raw recordings are retained.

### 7.4. Results

Table 11 and Figure 12 give the outcomes by variant and scenario.

**Table 11.** Physical trial outcomes with Wilson score 95% confidence intervals. A threshold violation is a LiDAR reading below 0.20 m that terminated the trial. Non-terminating approaches below this distance are reported as safety-zone intrusions and are not counted here; SAC-R-PV-STAM recorded two (0.160 m and 0.120 m), both in trials that reached the goal.

| Variant       | Scenario 2 (Static) | 95% CI | Scenario 3 (Moving) | 95% CI  | Threshold Violations | Timeouts |
|---------------|---------------------|--------|---------------------|---------|----------------------|----------|
| SAC-R-PV-STAM | 19/20 = 95.0%       | 76–99% | 20/20 = 100%        | 84–100% | 0/40                 | 1        |
| SAC-MLP-FS    | 14/20 = 70.0%       | 48–85% | 12/20 = 60.0%       | 39–78%  | 3/40                 | 11       |
| SAC-PV-STAM   | 1/10 = 10.0%        | 2–40%  | 11/20 = 55.0%       | 34–74%  | 9/30                 | 9        |
| SAC-PV-STAM-H | 0/10 = 0.0%         | 0–28%  | not run             | —       | 0/10                 | 10       |
| SAC-MLP       | 0/10 = 0.0%         | 0–28%  | not run             | —       | 0/10                 | 10       |

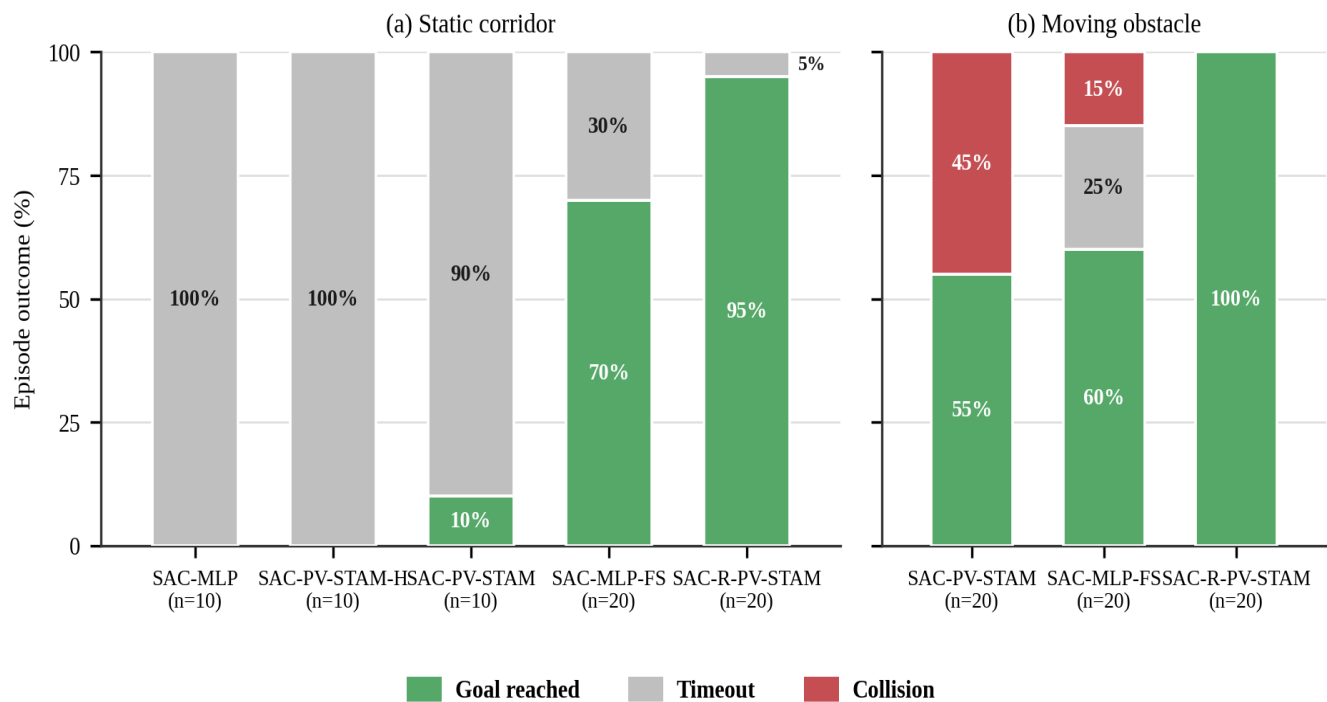


**Figure 12.** Physical trial outcomes by variant and scenario across 130 valid trials. (a) Static corridor (Scenario 2). (b) Moving obstacle (Scenario 3). Bars show the proportion of trials ending in goal reached, timeout or threshold violation; n is given under each bar. Wilson score 95% confidence intervals are given in Table 11. Results are not pooled across scenarios.

Results are reported per scenario and are not pooled. The scenario proportions differ by variant: SAC-R-PV-STAM was evaluated 50% in the moving-obstacle scenario and SAC-PV-STAM 67%. Where a single summary figure is required, the directly standardized rate, weighting both scenarios equally, is 97.5% for SAC-R-PV-STAM, 65.0% for SAC-MLP-FS, and 32.5% for SAC-PV-STAM. The unweighted pooled rate for SAC-PV-STAM is 40.0%, which exceeds its standardized rate by 7.5 points because it ran twice as many trials in the scenario in which it performs better.

SAC-PV-STAM-H and SAC-MLP were evaluated in the static scenario only, so no statement about their behavior under obstacle motion is made.

Table 12 gives pairwise comparisons.

**Table 12.** Pairwise comparisons. Within-scenario tests are Fisher exact; stratified tests are Cochran–Mantel–Haenszel stratified by scenario. All tests two-sided.

| Comparison                      | Static               | Moving | Stratified           |
|---------------------------------|----------------------|--------|----------------------|
| SAC-R-PV-STAM vs. SAC-MLP-FS    | 0.091                | 0.0033 | 0.00068              |
| SAC-R-PV-STAM vs. SAC-PV-STAM   | $6.7 \times 10^{-6}$ | 0.0012 | $1.1 \times 10^{-7}$ |
| SAC-MLP-FS vs. SAC-PV-STAM      | 0.0052               | 1.00   | 0.052                |
| SAC-R-PV-STAM vs. SAC-PV-STAM-H | $3.7 \times 10^{-7}$ | —      | —                    |
| SAC-R-PV-STAM vs. SAC-MLP       | $3.7 \times 10^{-7}$ | —      | —                    |
| SAC-MLP-FS vs. SAC-PV-STAM-H    | 0.0003               | —      | —                    |

Two points of interpretation. For SAC-MLP-FS against SAC-PV-STAM, the Mantel–Haenszel odds ratio is 2.87 with a 95% confidence interval of 1.08–7.68, which excludes 1, while the stratified *p*-value is 0.052. The two disagree because the stratified test applies a continuity correction and the interval does not. We report this comparison as not significant at the 0.05 level. The advantage of SAC-MLP-FS over SAC-PV-STAM is confined to the

static scenario ( $p = 0.0052$ ); under obstacle motion, the two are indistinguishable ( $p = 1.00$ ). For SAC-R-PV-STAM against SAC-PV-STAM, the common odds ratio is 306 with a 95% interval of 7.96–11,759; this estimate is inflated by near-zero cells and is not a usable effect size, so the  $p$ -value is reported and the point estimate omitted.

Table 13 gives behavioral measurements extracted from the commanded velocity and odometry topics.

**Table 13.** Behavioral signatures over valid physical trials in Scenario 2 (static corridor). Forward command active is the proportion of control steps commanding forward velocity above 0.005 m/s. Each quantity is computed per trial and then averaged across trials, with equal weight on each trial.

| Metric                            | SAC-MLP | SAC-PV-STAM-H | SAC-PV-STAM (Static) | SAC-MLP-FS | SAC-R-PV-STAM |
|-----------------------------------|---------|---------------|----------------------|------------|---------------|
| Forward command active            | 1.4%    | 19.9%         | 16.3%                | 52.0%      | 89.6%         |
| Mean forward speed (m/s)          | 0.0005  | 0.016         | 0.023                | 0.062      | 0.193         |
| Mean turn rate $ \omega $ (rad/s) | 0.481   | 1.006         | 0.101                | 0.405      | 0.897         |
| Furthest point reached (m)        | 0.03    | 1.12          | 2.75                 | 3.25       | 4.27          |
| Maximum lateral deviation (m)     | 0.40    | 0.40          | 0.78                 | 0.56       | 0.90          |
| Maximum trajectory progress (m)   | 0.22    | 1.35          | 4.22                 | 4.20       | 6.63          |
| Median trajectory progress (m)    | 0.02    | 1.25          | 2.83                 | 4.14       | 4.20          |

The proportion of steps commanding forward motion orders the variants in the same sequence as the success rate. SAC-R-PV-STAM and SAC-PV-STAM-H commanded comparable turn rates (0.897 against 1.006 rad/s); they differ in whether the policy translated while turning.

#### 7.4.1. Failure Modes

Four distinct failure modes were observed, each reproducible within its variant.

SAC-MLP did not depart. The mean distance traveled was 0.03 m in 150 s across all ten trials. Forward motion was commanded in 1.4% of steps, and turn commands were approximately balanced in sign (22,958 positive against 22,116 negative) and therefore self-canceling.

SAC-PV-STAM-H rotated without translating. The turn rate saturated at the 1.80 rad/s actuator limit while forward speed remained at 0.016 m/s. The robot advanced 1.12 m and rotated in place for the remainder of each trial. Failure locations had a standard deviation of 0.18 m, indicating a converged policy failing reproducibly. Actuator saturation alone does not account for this: SAC-R-PV-STAM was also saturated, achieving 0.099 m/s of an available 0.122 m/s under its commanded turn rate, and succeeded.

SAC-PV-STAM stalled. The robot drove to approximately 2.75 m and stopped. In a representative trial, the robot last gained ground at  $t = 29.6$  s at  $x = 2.74$  m and remained there for the final 122.3 s of the trial. Over that window, 12 of 3691 commands (0.3%) requested forward motion above 0.005 m/s, and over the final 91.3 s, the commanded forward velocity was exactly zero in all 2746 samples. The LiDAR reported 3.50 m, the sensor maximum, directly ahead. The robot was flanked, with an obstacle 0.75 m to one side and a wall 0.59 m to the other. Failure locations clustered at  $2.59 \pm 0.30$  m in nine of ten trials.

SAC-MLP-FS terminated bimodally. Trials ended either close to the start (0.44–0.47 m, timing out after the full 150 s) or at the goal, with few intermediate outcomes; failure locations had standard deviations of 1.11–1.16 m, an order of magnitude broader than the deterministic failures above. Three of its five timeouts in the moving-obstacle scenario occurred before the robot reached the crossing point and are therefore static failures rather than failures of dynamic avoidance.

SAC-R-PV-STAM exhibited no stall mode. Across 40 trials, it did not at any point stop commanding forward motion. Its single failure was a timeout at 3.12 m with forward motion commanded in 96.8% of steps.

#### 7.4.2. Controlled Test of the Stall Mechanism

The stall admits a testable account. In the frame-stacked variants, the temporal information available to the policy consists of three consecutive distance scans. In a static scene with the robot stationary, those three scans are near-identical, so the temporal component of the observation carries no information distinguishing a stopped robot from a scene in which nothing is approaching. The policy is not acting on a frozen input: the navigation state updates independently at 20 Hz, and 96.3% of issued commands differ from their predecessor. The temporal component specifically is uninformative. A moving obstacle changes successive scans whether or not the robot moves, so under obstacle motion, the condition sustaining the stall should not arise.

This prediction was stated before Scenario 3 was run. Table 14 gives the results.

**Table 14.** SAC-PV-STAM under static and moving obstacles.

|                          | Scenario 2, Static (n = 10) | Scenario 3, Moving (n = 20) |
|--------------------------|-----------------------------|-----------------------------|
| Success                  | 10.0%                       | 55.0%                       |
| Timeouts                 | 9/10                        | 0/20                        |
| Threshold violations     | 0/10                        | 9/20 (45%)                  |
| Forward command active   | 16.3%                       | 79.6%                       |
| Mean forward speed (m/s) | 0.023                       | 0.129                       |

Timeouts fell from 9 of 10 to zero (Fisher exact,  $p = 7 \times 10^{-7}$ ) and the proportion of steps commanding forward motion rose approximately five-fold. The stall was removed under the predicted condition.

The policy did not become safe. It recorded threshold violations in 45% of trials in the moving-obstacle scenario, the highest rate measured in this study, terminating at a mean lateral displacement of 1.23 m. With a corridor half-width of 1.65 m and a robot half-width of 0.22 m, this places the robot approximately 0.2 m from the wall. The trajectories show the robot avoiding the moving obstacle and continuing into the corridor wall. SAC-R-PV-STAM's avoidance was bounded by comparison, with a maximum lateral deviation of 0.90 m, no threshold violations in 40 trials, and a minimum recorded clearance of 0.120 m.

On successful trials, SAC-PV-STAM was faster than SAC-R-PV-STAM (43.6 s against 62.1 s to goal). The recurrent variant is not a faster driver; it stops at a larger margin.

These two results separate the roles of the two components. Temporal information in the observation, whether from stacked frames or from the explicit scan-difference channel, determines whether the policy stalls. The recurrent state determines whether the resulting motion stays within the corridor. Neither alone was sufficient in these trials.

#### 7.5. Measured Deployment Conditions

Table 15 gives the conditions under which all variants were evaluated. Three distinct rates are reported separately.

The policy executes at 30 Hz while the LiDAR delivers scans at 4.85 Hz, so 83.9% of control decisions are taken on a scan already used at least once. Because the previous scan is latched in the sensor callback, the scan-difference channel retains its last value between scan arrivals rather than falling to zero, and the policy acts on a range difference up to 200 ms old for the majority of its decisions. The navigation state updates independently at 20 Hz, so the observation continues to change between scans and 96.3% of issued commands

differ from their predecessor. Consecutive scans were never byte-identical across 1603 pairs examined, confirming that the sensor delivers new data at its measured rate.

The simulation control period is 150 ms and the deployment period is 33.3 ms, a ratio of 4.5, and no compensation for this difference is applied at deployment. A policy trained to hold an action for 150 ms holds it for 33 ms on the platform. This condition is applied identically to all variants. It cannot account for differences between them: SAC-R-PV-STAM obtained 95.0% and 100% success under this mismatch, and the controlled manipulation in Section 7.4.2 varies only obstacle motion while holding the rate constant. It is reported here as a measured deployment condition and is listed as a limitation in Section 8.3.

**Table 15.** Measured deployment conditions, identical across variants.

| Quantity                                      | Measured Value                                     |
|-----------------------------------------------|----------------------------------------------------|
| Policy inference rate                         | 30 Hz, fixed-rate timer                            |
| Command publication rate                      | 30.07 Hz (mean over six trials, range 29.92–30.14) |
| Odometry update rate                          | 20.02 Hz                                           |
| LiDAR update rate                             | 4.85 Hz (207 ms mean interval, 3.1% dropped scans) |
| Scan-difference update rate                   | 4.85 Hz, held between updates                      |
| Control cycles acting on an already-used scan | 83.9%                                              |
| Commands differing from predecessor           | 96.3%                                              |
| Commanded-to-observed velocity lag            | 152 ms (Section 5.2)                               |
| Actuator coupling                             | $v_{\max} = 0.26 -  \omega  \cdot 0.1435$          |
| Goal-distance normalization                   | $d/5.0$                                            |

## 8. Discussion and Conclusions

### 8.1. Simulation and Physical Evaluation Give Different Orderings

The canonical evaluation and the physical trials disagree. In simulation SAC-PV-STAM obtained 92.0% on Benchmark A against 91.3% for SAC-R-PV-STAM, and SAC-MLP-FS led on Benchmark C at 61.0% against 55.0%. On the physical platform the standardized rates were 97.5% for SAC-R-PV-STAM, 65.0% for SAC-MLP-FS and 32.5% for SAC-PV-STAM, with stratified  $p$ -values of 0.00068 and  $1.1 \times 10^{-7}$ .

The conditions in Table 15 provide an account of this. In simulation the LiDAR is sampled at the control rate and the observation is complete at every step, so the temporal component of the observation is always current. On the platform, the sensor delivers 4.85 Hz against a 30 Hz control loop, the corridor is narrower than the training arena, and forward and angular velocity compete for the same actuators. Under those conditions SAC-PV-STAM, which matched SAC-R-PV-STAM in simulation, held zero forward velocity for 91.3 s with 3.5 m of clear floor ahead.

The scope of this claim requires care. Simulation separated architectures differing in gross capability: SAC-MLP, SAC-LSTM and both SAC-PV-STAM-H configurations are distinguishable from the leading group on all three benchmarks in Table 6, and the LSTM baseline's 19.3% timeout rate on Benchmark A shows that stalling behavior is detectable in simulation when its cause is architectural. What the simulation did not separate were policies differing in how they handle sparse and asynchronous observations. Architectures that are equivalent under complete synchronous observation diverged under a sensor delivering one scan per six control decisions.

Two methodological consequences follow. Physical trial counts and confidence intervals should be reported alongside simulation success rates, since the two need not agree [37]. Simulation benchmarks intended to discriminate between temporal architec-

tures should model sensor rate and latency explicitly rather than sampling observations at the control rate.

### 8.2. Advantages and Limitations of the Approach

The scan-difference channel supplies motion information without scan registration, optical flow or a maintained recurrent state, at a cost of 19,968 parameters, less than 3% of the smallest complete variant in Table 3. It requires no calibration and runs in time linear in the number of sectors. Combined with a recurrent encoder, it produced the only variant in this study that navigated the physical corridor reliably, with no threshold violations across 40 trials in both scenarios.

The channel has three limitations established by these experiments. It is ego-centric and contaminated by the robot's own rotation, by a factor of 2.27 in simulation and 12.0 on hardware (Section 3.1); the mitigation employed here supplies  $\omega$  to the policy rather than compensating for ego-motion explicitly. It carries no information when nothing in the scene moves, which is the condition under which the non-recurrent variants stalled. On a platform whose sensor rate is a sixth of the control rate, the value acted on is stale for the majority of decisions. Section 7.4.2 further shows that the channel alone does not produce safe avoidance: without a recurrent state, the policy stopped stalling and began violating the threshold in 45% of trials.

### 8.3. Limitations

The physical results use a single seed (seed 777). The other trained seeds used a different observation convention and cannot serve as a seed-robustness check without confounding seed with observation format. Simulation results pool three seeds throughout.

Sample sizes in the physical evaluation are 10 to 20 trials per variant per scenario, giving wide confidence intervals, and the two scenarios were not sampled in equal proportion across variants, so comparisons are stratified by scenario. SAC-R-PV-STAM and SAC-MLP-FS differ significantly under obstacle motion ( $p = 0.0033$ ) and when stratified ( $p = 0.00068$ ) but not on static obstacles alone ( $p = 0.091$ ). SAC-PV-STAM-H and SAC-MLP were evaluated in the static scenario only. The corridor is out of distribution relative to the open training arena. The moving obstacle was drawn by hand, and its crossing point varied by approximately  $\pm 0.7$  m across trials; the between-variant means of this quantity are smaller than the within-variant spread, so this contributes noise rather than bias. Of 143 recorded trials, 130 were analyzed and 13 were excluded with documented cause.

The variants are not parameter-matched. SAC-R-PV-STAM has 3.3 times the parameters of SAC-PV-STAM (Table 3), so the comparison between them confounds the observation channel, the recurrent encoder and network capacity.

The SAC-PV-STAM-H comparison in Section 6.4 is not fully isolated. In addition to the critic loss, SAC-PV-STAM resets the SAC entropy target on entry to Phase 7, while neither SAC-PV-STAM-H configuration does. The width-matched comparison removes the critic-width confound but not this one.

The HCTM parameter values in Section 5.2 could not be traced to a documented calibration. The independently measured commanded-to-observed lag of 152 ms is approximately 2.3 times the 65 ms modeled, though the two quantities are not directly comparable.

The attention-based mitigation of rotation contamination described in Section 3.1 is a design argument and is not tested. Attention weights were recorded at deployment for representative scenarios but not jointly with angular velocity, so the pairing required to test the down-weighting mechanism was not captured.

The deployment control rate is 4.5 times the training rate (Section 7.5). The SAC-LSTM baseline is unstable across seeds. All simulated obstacles were cylinders on sinusoidal

paths. No variant was trained to remain stationary at the goal, an artifact of the training termination condition.

#### 8.4. Conclusions

- Across three zero-shot benchmark arenas at 300 episodes each, SAC-PV-STAM, SAC-R-PV-STAM and SAC-MLP-FS lie within 4.6 percentage points on two of three benchmarks, and SAC-MLP-FS leads on the third. Across 130 physical trials, the same three variants separate with stratified  $p$ -values of 0.00068 and  $1.1 \times 10^{-7}$ . Simulation benchmarks of this kind did not have the resolution to rank policies differing in their handling of sparse observations.
- A stall failure mode was identified in the non-recurrent variants and tested under a controlled manipulation. SAC-PV-STAM commanded no forward velocity above 0.005 m/s in any of the 2746 samples of the final 91.3 s of a 122.3 s stall, with 3.50 m of clear floor ahead. Introducing a moving obstacle removed the stall as predicted (timeouts 9/10 to 0/20,  $p = 7 \times 10^{-7}$ ).
- Temporal information and recurrence contribute differently. Temporal information determines whether the policy stalls; the recurrent state determines whether the resulting avoidance stays within the corridor. Without recurrence, the policy stopped stalling and recorded threshold violations in 45% of trials.
- The width-matched comparison shows the effect was environment-dependent: at matched critic width, success rose 15.0 points on Benchmark A and fell 24.0 and 27.7 points on Benchmarks B and C. The negative result holds; its magnitude is not attributable to the loss alone.
- A recurrent LSTM baseline trained under a fixed step budget per phase remains significantly inferior on all three benchmarks and is unstable across seeds, with Benchmark C performance spanning approximately 36 points between seeds.
- On this platform and under this single-seed evaluation, SAC-R-PV-STAM was the only variant that navigated reliably, on the basis of 19 of 20 successes in the static scenario, 20 of 20 under obstacle motion, and no threshold violations across 40 trials.

#### 8.5. Future Work

Four directions follow from these results.

**Ego-motion-compensated range difference.** Subtracting the predicted range change due to the robot's own motion, using wheel odometry or the IMU, would separate obstacle motion from ego-motion in the scan-difference channel. The retained trial recordings permit this to be evaluated offline before any retraining, which would establish whether the reported effects derive from obstacle-motion information specifically or from a more general temporal-change signal.

**Multi-arena curricula.** Section 6.7 shows that transferring a converged open-arena policy to corridor geometry by fine-tuning degraded it. Interleaving corridor geometry into the curriculum from the start would not require the policy to overwrite converged behavior.

**Rate-matched deployment and sensor-rate simulation.** Deploying at the training control period and modeling sensor rate and latency explicitly in simulation rather than sampling observations at the control rate would test directly whether the discrepancy reported in Section 8.1 narrows.

**Broader physical evaluation.** Multiple seeds trained under a single observation convention, matched scenario counts for every variant, additional physical environments, and 30 to 50 trials per cell would be required to convert the present result into a deployment reliability claim. Instrumenting the policy to record attention weights alongside angular velocity would make the mitigation argument in Section 3.1 testable.

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

**Funding:** This research was funded by The Scientific and Technological Research Council of Türkiye (TÜBİTAK), grant number 123E406, and by Firat University Scientific Research Projects Unit (FÜBAP), grant numbers MF.24.80, MF.25.154, and MF.25.155. Part of this work was supported within the scope of the TÜBİTAK 2209-A project, grant number 1919B012439942, conducted by Anas M. N. S. Alqadhi.

**Institutional Review Board Statement:** Not applicable. The real-robot evaluation reported in Section 7 used cardboard obstacles only. No human participants took part as obstacles or in any other role, and no identifiable person appears in any figure.

**Informed Consent Statement:** Not applicable. No human participants were involved.

**Data Availability Statement:** The 130 real-robot trial recordings, the trial index recording per-trial exclusion reasons, the simulation training and evaluation logs, and the figure-generation scripts are available from the corresponding author on reasonable request. Training code and evaluation scripts will be made publicly available upon acceptance at <https://github.com/railab-firat/PV-STAM-mapless-DRL-navigation> (accessed on 6 September 2026).

**Acknowledgments:** The authors thank the RAI Laboratory technical staff at Firat University for hardware support during real-robot experiments. This work was conducted within the RAI Laboratory, Mechatronics Engineering Department, Firat University, Elazığ, Türkiye. During the preparation of this manuscript, the authors used Claude Opus 5 (Anthropic, San Francisco, CA, USA) to verify the reported numerical results against the raw experimental logs, to check internal consistency between the text, tables and figures, and for language editing. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
2. Tai, L.; Paolo, G.; Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 31–36.
3. Zhelo, O.; Zhang, J.; Tai, L.; Liu, M.; Burgard, W. Curiosity-driven exploration for mapless navigation with deep reinforcement learning. *arXiv* **2018**, arXiv:1804.00456.
4. de Heuvel, J.; Zeng, X.; Shi, W.; Sethuraman, T.; Bennewitz, M. Spatiotemporal attention enhances lidar-based robot navigation in dynamic environments. *IEEE Robot. Autom. Lett.* **2024**, *9*, 4202–4209. [[CrossRef](#)]
5. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
6. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
7. Kapturowski, S.; Ostrovski, G.; Quan, J.; Munos, R.; Dabney, W. Recurrent experience replay in distributed reinforcement learning. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.
9. Chaffre, T.; Moras, J.; Chan-Hon-Tong, A.; Marzat, J. Sim-to-real transfer with incremental environment complexity for reinforcement learning of depth-based robot navigation. In *Proceedings of the ICINCO, 2020*; Scitepress: Setúbal, Portugal, 2020; pp. 314–323. [[CrossRef](#)]
10. Fan, T.; Long, P.; Liu, W.; Pan, J. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *Int. J. Robot. Res.* **2020**, *39*, 856–892. [[CrossRef](#)]

11. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
12. Zhao, W.; Queralta, J.P.; Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 1–4 December 2020; pp. 737–744. [\[CrossRef\]](#)
13. Liu, T.; Zhou, Y.; Zhao, Z.; Luo, Z.; Wang, H.; Deng, X.; Liu, Q. Lift enhancement optimization for an airfoil with distributed dual synthetic jets based on deep reinforcement learning. *Chin. J. Aeronaut.* **2026**, *53*, 104128. [\[CrossRef\]](#)
14. Ou, Y.; Cai, Y.; Sun, Y.; Qin, T. Autonomous navigation by mobile robot with sensor fusion based on deep reinforcement learning. *Sensors* **2024**, *24*, 3895. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Alexander, A.; Vangaveeti, V.N.S.; Venkatesan, K.; Mounsef, J.; Ramanujam, K. Adaptive emergency response and dynamic crowd navigation for mobile robot using deep reinforcement learning. *Front. Robot. AI* **2025**, *12*, 1612392. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Zhu, Y.; Wan Hasan, W.Z.; Harun Ramli, H.R.; Norsahperi, N.M.H.; Mohd Kassim, M.S.; Yao, Y. Deep reinforcement learning of mobile robot navigation in dynamic environment: A review. *Sensors* **2025**, *25*, 3394. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Wang, P.; Li, X.; Zhu, M.; Ma, J. Editorial: Reinforcement learning for real-world robot navigation. *Front. Robot. AI* **2026**, *13*, 1861947. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Parisotto, E.; Song, H.F.; Rae, J.W.; Pascanu, R.; Gulcehre, C.; Jayakumar, S.M.; Jaderberg, M.; Kaufman, R.L.; Clark, A.; Noury, S.; et al. Stabilizing transformers for reinforcement learning. In Proceedings of the 37th International Conference on Machine Learning (ICML), Online, 13–18 July 2020; pp. 7487–7498.
19. Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems 34*; ACM Digital Library: New York, NY, USA, 2021; pp. 15084–15097.
20. Chen, C.; Liu, Y.; Kreiss, S.; Alahi, A. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6015–6022.
21. Yang, Y.; Jiang, J.; Zhang, J.; Huang, J.; Gao, M. ST2: Spatial-temporal state transformer for crowd-aware autonomous navigation. *IEEE Robot. Autom. Lett.* **2023**, *8*, 912–919. [\[CrossRef\]](#)
22. Lu, X.; Faragasso, A.; Wang, Y.; Yamashita, A.; Asama, H. Group-aware robot navigation in crowds using spatio-temporal graph attention network with deep reinforcement learning. *IEEE Robot. Autom. Lett.* **2025**, *10*, 4140–4147. [\[CrossRef\]](#)
23. Li, Y.; Lyu, Q.; Yang, J.; Salam, Y.; Wang, B. Visual target-driven robot crowd navigation with limited FOV using self-attention enhanced deep reinforcement learning. *Sensors* **2025**, *25*, 639. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Cui, Z.; Guan, W.; Zhang, X. USV formation navigation decision-making through hybrid deep reinforcement learning using self-attention mechanism. *Expert Syst. Appl.* **2024**, *256*, 124906. [\[CrossRef\]](#)
25. Narvekar, S.; Peng, B.; Leonetti, M.; Sinapov, J.; Taylor, M.E.; Stone, P. Curriculum learning for reinforcement learning domains: A framework and survey. *J. Mach. Learn. Res.* **2020**, *21*, 1–50. [\[CrossRef\]](#)
26. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum learning. In Proceedings of the International Conference on Machine Learning (ICML), Montreal, QC, Canada, 14–18 June 2009; pp. 41–48.
27. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30.
28. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1–8.
29. Muratore, F.; Ramos, F.; Turk, G.; Yu, W.; Gienger, M.; Peters, J. Robot learning from randomized simulations: A review. *Front. Robot. AI* **2022**, *9*, 799893. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Mirowski, P.; Pascanu, R.; Viola, F.; Soyer, H.; Ballard, A.J.; Banino, A.; Denil, M.; Goroshin, R.; Sifre, L.; Kavukcuoglu, K.; et al. Learning to navigate in complex environments. *arXiv* **2016**, arXiv:1611.03673.
31. Xie, L.; Wang, S.; Markham, A.; Trigoni, N. Towards monocular vision based obstacle avoidance through deep reinforcement learning. *arXiv* **2017**, arXiv:1706.09829.
32. Wu, H.; Sun, F.; Kang, Z.; Tang, L.; Chu, Z. A novel sim-to-real reinforcement learning algorithm for soft growing robot navigation. *Sci. China Inf. Sci.* **2025**, *68*, 214201. [\[CrossRef\]](#)
33. Ampuero, G.C.; Hermosilla, G.; Varas, G.; Toribio Clark, M. Deep reinforcement learning for sim-to-real robot navigation with a minimal sensor suite for beach-cleaning applications. *Appl. Sci.* **2025**, *15*, 10719. [\[CrossRef\]](#)
34. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.

35. Agarwal, R.; Schwarzer, M.; Castro, P.S.; Courville, A.C.; Bellemare, M.G. Deep reinforcement learning at the edge of the statistical precipice. In *Proceedings of the NIPS'21: Proceedings of the 35th International Conference on Neural Information Processing Systems*; ACM Digital Library: New York, NY, USA, 2021; pp. 29304–29320.
36. Balazadegan Sarvrood, Y.; Hosseinyalamdary, S.; Gao, Y. Visual-LiDAR odometry aided by reduced IMU. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 3. [[CrossRef](#)]
37. Hanna, J.P. Toward the confident deployment of real-world reinforcement learning agents. *AI Mag.* **2024**, *45*, 396–403. [[CrossRef](#)]

**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.