

PEDS: Closed-Loop Benchmarking of Social Navigation across Pedestrian Behavior Models

Duc Tai Nguyen¹, Volodymyr Shcherbyna^{1,2}, Anh Do Duc¹, Zhengcheng Shen³, and Linh Kästner¹

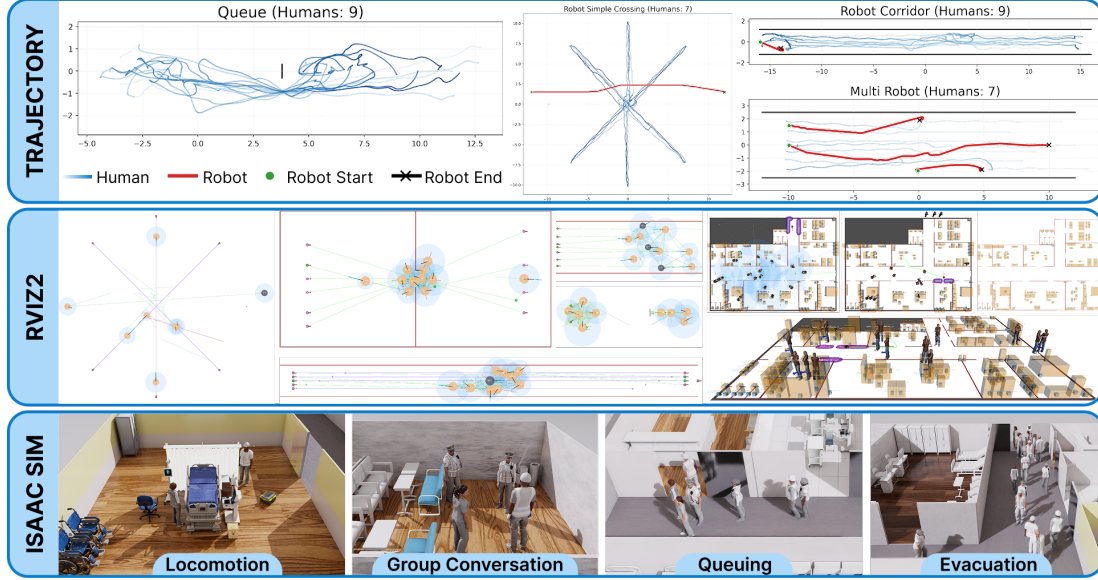


Fig. 1: *Scenario Montage of PEDS*: PEDS covers lightweight batch runs and photorealistic evaluation. **Top**: Trajectories recorded in scenarios from Section IV, with no robot (Queue), a single robot (Simple Crossing, Corridor), or multiple robots (Multi-Robot). Human paths are blue, and darker blue means later in the run. **Middle**: RViz2 integration visualizes agent goals, visual cones, and personal spaces during a trial. **Bottom**: IsaacSim scenes on that hospital map show photorealistic rendering of a locomotion scenario, in which a robot navigates through human-occupied space, and of three social-interaction scenarios: group conversation, queuing, and evacuation.

Abstract—Social navigation benchmarks evaluate robot policies but leave the pedestrian behavior model as an uncontrolled axis of evaluation, conflating claims about policies with claims about the human models used to evaluate them. We introduce PEDS (Pedestrian Evaluation Dataset & Simulator), a modular human simulator that treats the pedestrian driver as the unit of variation. A deterministic executor synchronizes the driver and the robot policy so that, at fixed seed, driver substitution is the only source of trial-to-trial variation. The release comprises six pedestrian drivers across four model classes: force-based (SFM, HSFM), velocity-based (ORCA), a no-avoidance baseline (Straight), and imitation-learned (NSP, SocialGAIL). The accompanying dataset spans 20 pedestrian-driver scenarios on a load $\times$ density grid, crossed with four robot policies in closed loop. Within-policy success varies by up to 14 percentage points

across pedestrian drivers overall, and up to 30 percentage points in dense navigation scenarios. Cross-class trajectory divergence averages $1.5\times$ in general and peaks above $3\times$ on dense scenarios. On pure-navigation scenarios, a change of seed moves trajectories more than a change of driver (ratio 0.75), on behavior-tree scenarios the driver effect is the larger one (1.43). The dataset, source code, and comprehensive evaluation results are available at <https://peds.arena-rotnav.org/>.

Index Terms—Software Tools for Benchmarking and Reproducibility, Performance Evaluation and Benchmarking, Human-Aware Motion Planning, Modeling and Simulating Humans

I. INTRODUCTION

SOcial navigation benchmarks ask whether a robot policy can move safely and efficiently through crowds [1], [2]. However, they leave a different variable uncontrolled: the pedestrian behavior model. A policy that succeeds under one pedestrian driver can fail under another, even when both produce similar spatial statistics but differ in their reactive dynamics. As a result, performance and robustness are correlated with the choice of pedestrian model, potentially introducing systematic bias in evaluation. A recent community-wide call for principled evaluation methodology articulates this concern [3], yet most evaluation environments fix the pedestrian

Manuscript received: June, 22, 2026; Revised September, 20, 2026; Accepted October, 14, 2026.

This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments.

¹Duc Tai Nguyen, Volodymyr Shcherbyna, Anh Do Duc, and Linh Kästner are with Singapore Management University, Singapore vshcherbyna@smu.edu.sg

²Volodymyr Shcherbyna is also with Technical University of Berlin, Berlin, Germany

³Zhengcheng Shen is with Max Planck Institute for Plasma Physics, Garching, Germany zhengcheng.shen@ipp.mpg.de

Digital Object Identifier (DOI): see top of this page.

TABLE I: *Pedestrian-motion paradigm and substrate coverage across prior social-navigation simulators and benchmarks.* Cross-class divergence calibration is measurable only when *Pluggable*, *Reproducible*, and ≥ 2 paradigms co-occur, which holds for PEDS alone.

	Force	Geom.	Behav.	Learned	Reprod.	Plug.
CrowdBot	✓	✗	✗	✗	✗	✗
SocNavBench	✗	✗	✗	✗	✓	✗
HuNavSim	✓	✗	✓	✗	✗	✗
SEAN 2.0	✗	✓	✓	✗	✗	✗
Arena 5.0	✓	✗	✓	✗	✗	✓
Ours	✓	✓	✓	✓	✓	✓

Geom.: Geometric; Behav.: Behavioral;
Reprod.: Reproducible; Plug.: Pluggable.

driver as a design choice and provide no mechanism to vary it.

Existing benchmarks either bake in a single pedestrian driver family or replay fixed trajectories (Section II-B). We treat the pedestrian driver as the primary unit of variation and quantify how much a reported performance depends on it, which addresses the evaluation concerns raised in [3]. We introduce PEDS, a modular human simulator built on ROS 2 [4], with live simulations shown in Figure 1. Pedestrian behaviors are composable drivers with a global-pathfinder slot and a local collision-avoidance slot, and either component can be substituted while the navigation intent stays fixed. Six reference drivers span the force-based (SFM, HSFM) and velocity-based (ORCA) families, a no-avoidance control (Straight), and learned drivers (NSP, SocialGAIL) with pre-trained weights. PEDS samples pedestrian attributes such as vision range and walking velocity per trial, and exposes human-robot (HRI), human-object (HOI), and human-human (HHI) interaction primitives for behavior-tree scenario specification (Figure 2).

Our contributions are as follows:

- PEDS, a modular human simulator whose pedestrian-driver interface is a unit of variation rather than a fixed design choice, released with six reference drivers and a correspondence test suite that measures each adapter against its public upstream reference where one exists.
- A companion trial dataset¹: 136 hours of paired-seed crowd simulation across 20 scenarios, six drivers, and four robot policies.
- A kinematic evaluation harness that isolates pedestrian-driver substitution from simulator variance, demonstrating that pedestrian models function as active, reportable determinants of robot evaluation outcomes.

II. RELATED WORK

A. Pedestrian drivers

Classical models are based on hand-designed forces and geometric constraints. The SFM [5] models pedestrians as particles under attractive and repulsive potentials. The HSFM [6] adds a forward-facing body frame to resolve rotational indeterminacy. ORCA [7] derives velocity updates from reciprocal velocity obstacles and guarantees collision-free behavior under

mutual cooperation. Moussaïd et al. [8] propose a cognitive-heuristics formulation that replaces force superposition with a small rule set. These models are analytically tractable but assume a specific reactive structure that may break in spatial configurations the model was not fit to.

Learned drivers replace the hand-designed structure with a function fit to trajectory data. This lineage begins with trajectory-prediction networks [9], [10], which are fit to datasets like ETH [11] and UCY [12] and benchmarked on Trajnet++ [13], and extends to reactive drivers usable inside a simulator. NSP [14] retains a physics-style dynamics term and learns a residual. The authors release pre-trained weights, of which PEDS runs the SDD-trained checkpoints. SocialGAIL [15] applies attention-graph-based generative adversarial imitation learning [16] to crowd trajectories and likewise comes with pre-trained weights. These models give up analytical clarity in exchange for trajectory fidelity on specific datasets, and whether that fidelity carries over to reactive, evaluation-time behavior is open.

B. Human-aware simulators and benchmarks

The Arena-Rosnav family [17], [18] provides a training and evaluation environment with SFM-family pedestrians. Its successor Arena 5.0 [19] exposes a plug-in interface for pedestrian-behavior modules but populates it only with force-based drivers. HuNavSim [20] adds richer behavioral scenarios over the same force-based core. SocNavBench [21] replays pre-recorded ETH and UCY trajectories, sacrificing reactivity for reproducibility. SEAN 2.0 [22] formalizes social situation generation around geometric avoidance and scripted social states. CrowdBot [23] instruments real-robot-in-the-loop trials with force-based crowd patterns. JRDB [24] is a real-world egocentric dataset of humans around a mobile robot for perception benchmarks, and ATC [25] provides real-world pedestrian trajectories from a shopping mall that we use for distributional comparisons. HA-VLN [26] couples pre-recorded SMPL-mesh playback with vision-and-language navigation, and is closer to a downstream task than a peer simulator. Broader embodied-AI simulators such as Habitat [27], iGibson [28], and MetaUrban [29] target indoor interaction and outdoor micromobility and emphasize photorealism and scale, but do not expose the pedestrian driver as a composable unit. Each system covers a slice of the pedestrian-motion paradigm space (Table I). PEDS populates a single driver interface across the force-based, geometric, behavioral, and learned paradigms, and pairs it with a deterministic executor so that swapping the driver is the only source of trial-to-trial variation.

a) *Evaluation guidelines*: Francis et al. [3] recommend a documented and diverse human behavior model, reproducible protocols with repeatability at scale, and social metrics beside task metrics. PEDS meets the first with the pluggable driver interface (Section III-A), the second with the deterministic executor and the paired-seed protocol (Section III-D), and the third with the PSV metric (Section III-C). Real-world validation, which the guidelines also recommend, is outside the scope of a simulation benchmark (Section V).

¹doi:10.57967/hf/11009

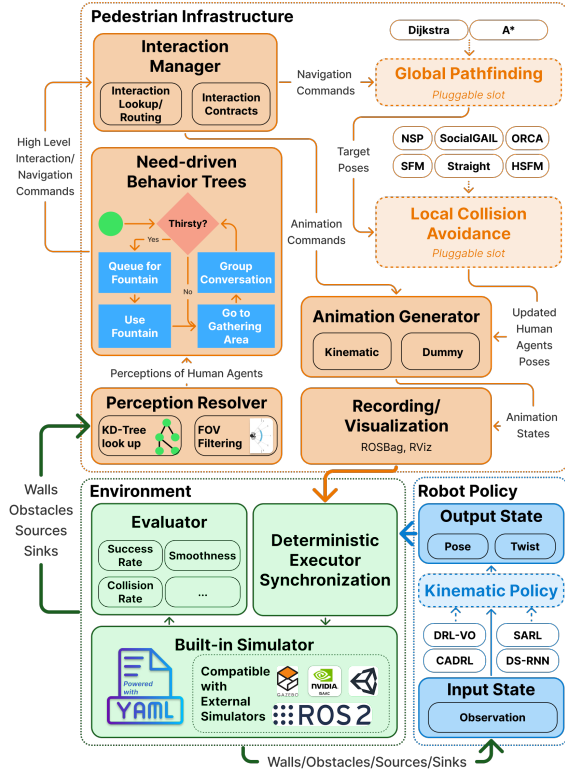


Fig. 2: *System overview.* PEDS decouples the Pedestrian Infrastructure (orange), the Environment (green), and the Robot Policy under test (blue). Dashed boxes are pluggable slots, their chips are the implementations currently available. Local Collision Avoidance is the pedestrian-driver slot, filled by one of six reference drivers per trial. The Deterministic Executor synchronizes all layers such that driver substitution becomes the distinct source of trial variation.

C. Robot navigation policies under evaluation

We catalog robot navigation policies only to clarify scope. CADRL [30] and SARL [31] are learned policies that map pedestrian-observation vectors to collision-avoiding actions. DS-RNN [32] extends the lineage with attention-structured recurrence. DRL-VO [33] combines velocity obstacles with learned tuning. Policies live in the paper-only kinematic evaluation harness and are not part of the public release, because the release contract covers pedestrian drivers only.

III. PEDS: SYSTEM DESIGN

Treating the pedestrian behavior model as a controlled variable requires (i) representations that allow systematic variation of pedestrian behavior, (ii) mechanisms to model interactive human behaviors, and (iii) an execution pipeline that isolates driver substitution from other sources of variation. PEDS is designed around these requirements.

A. Pedestrian driver interface

To enable controlled variation of pedestrian behavior, we present a pedestrian model as a composition of two modules, a *global pathfinder* and a *local collision-avoidance policy*. PEDS implements both modules as independent slots. To cover

diverse behavioral regimes, we provide six reference drivers spanning force-based (SFM, HSFM), velocity-based (ORCA), imitation-learned policies (NSP, SocialGAIL) released with pre-trained weights, and a no-avoidance baseline (Straight) that drives toward the subgoal without reacting to neighbors.

B. Modeling human behavior and interactions

The driver interface governs locomotion. Higher-level intent and interactions come from two modes of behavior specification. A *pure navigation mode* captures waypoint-following scenarios. A *need-driven behavior-tree* formulation lets human needs interrupt ongoing activities and trigger new behaviors, which prior simulators [20] with fixed action sequences cannot express. We further expose a set of interaction primitives (native social interaction nodes) covering human–robot (HRI), human–object (HOI), and human–human (HHI) interactions, which are composable with any driver. Interactions are coordinated through a handshake protocol, where agents propose, accept, or decline interaction requests, forming temporary interaction contracts without requiring manual synchronization. This separation of *social behavior* and *locomotor control* enables independent variation of both factors.

C. Stochastic human sampler

To capture population-level variability, we model pedestrians as distributions rather than fixed agents. Each trial samples pedestrian attributes (e.g., perception range, preferred velocity, and behavioral parameters) from scenario-specific distributions.

Defaults are grounded in empirical pedestrian-dynamics literature [34], [35], [5], [8], while scenario authors can override distributions as needed. The personal-space radius used by the PSV metric of Section IV, 0.5 m around each pedestrian, follows the inner bound of the personal zone in the proxemics literature [36], [37].

D. Deterministic evaluation pipeline

To prevent asynchronous execution between pedestrians and the robot from confounding driver substitution, we design a synchronized deterministic execution pipeline. The deterministic executor advances pedestrian drivers and the robot policy in a synchronized cycle with a fixed time step. Randomness is confined to the sampling of spawn events and agent parameters. Every driver-independent quantity (spawn events, initial states, radii, preferred velocities, behavior-tree choices, interaction schedules) is drawn from a random-number stream named by its purpose and seeded by a hash of the trial seed and that name. Because drivers consume no randomness, the spawn schedule is a function of the seed alone: under the concurrency cap a driver can delay a release but cannot change it. An integration test runs two drivers on one seed and requires the same released agents with identical first pose, preferred velocity, and radius. On the released sweeps, all 31,906 agents shared by driver pairs across the nav scenarios at three seeds have identical first poses. All 20,809 agents of the divergence sweep have identical radii under every driver that released

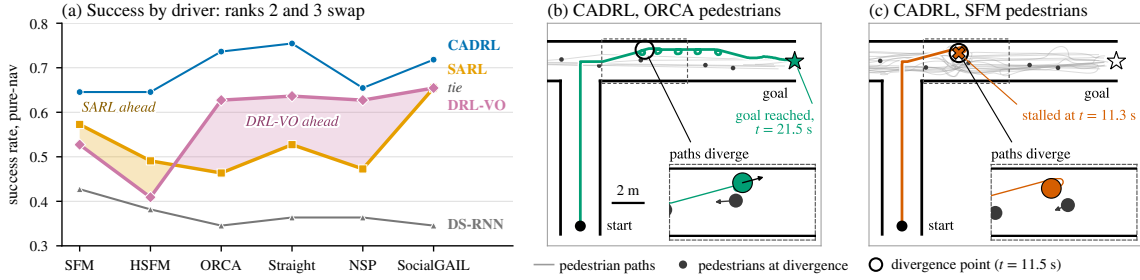


Fig. 3: PEDS treats the pedestrian driver as an axis of evaluation because the same robot policy (CADRL) stalls or reaches the goal depending on which driver controls the surrounding pedestrians (b, c). Furthermore, per-driver policy ranks reorder across the six drivers on the pure-navigation scenarios (a). Panels (b) and (c) show the same trial under ORCA and SFM pedestrians, respectively.

them. Trials that differ only in the driver therefore contain trajectory differences attributable to that substitution alone, and the executor removes scheduler jitter and message ordering as sources of variation. It does not remove floating-point nondeterminism of the GPU kernels in the learned drivers, and bitwise repeatability of whole trajectories over repeated runs was not measured (Section V).

a) Correspondence tests: Finally, to ensure that each driver faithfully implements its intended model, we validate implementations against their upstream references using correspondence tests. Each released driver with a public reference is tested against trajectory targets (median Hausdorff distance ≤ 0.25 m and mean velocity error ≤ 0.05 m/s) on held-out scenarios. These targets are significantly tighter than the cross-driver divergences observed in Section IV-B. The Straight driver meets both targets on every scenario, while ORCA meets them on three of six scenarios. SFM fails on all scenarios because it differs from its reference by specification: the released driver integrates the circular Helbing potential and pysocialforce the elliptical one. The test therefore measures the gap between two published SFM variants. Since SFM and HSFM anchor the force-based class, Section IV-B gives K with each driver removed in turn. HSFM has no public implementation. NSP and SocialGAIL ship pre-trained weights, which our adapters load, but their public code is a prediction or training pipeline that cannot be stepped alongside our driver, which leaves them without a closed-loop reference.

E. Scenario suite

The benchmark contains 20 scenarios organized along two axes: *pedestrian load* and *crowd density*. *Pedestrian load* distinguishes *pure-navigation (nav)* scenarios, which require pedestrians to execute low-level waypoint tracking only, from *behavior-tree-loaded (BT)* scenarios, which introduce high-level interactions (HRI, HOI, HHI) such as group formation or task execution. *Crowd density* distinguishes *sparse* from *dense* spatial layouts. *Sparse* scenarios range between 3 and 32 pedestrians, while *dense* scenarios contain between 12 and 100 agents. Note that the density set per scenario is also determined by the layout design rather than a strict agent count. The closed-loop evaluation uses the robot variants of these 20 scenarios and adds four robot-only scenarios. One

is a multi-robot flow scenario. The other three are *heterogeneous* scenarios whose crowds mix agent types, namely mixed walking speeds, a robot embedded in a walking group, and a wheelchair user at a bottleneck. We put the heterogeneous scenarios in a separate bucket and do not include them in Table IV.

IV. EVALUATION

Two studies evaluate the pedestrian driver as an axis of benchmarking: Section IV-B quantifies how pedestrian trajectories differ across drivers under identical conditions, and Section IV-C measures the downstream effect on robot navigation policies.

A. Experimental setup

a) Metrics: We report five metrics capturing complementary aspects of performance. Trials are recorded at $\Delta t = 0.05$ s. *Success rate* is the fraction of trials in which the robot comes within 0.5 m of its goal at any frame within the time budget. *Time-to-goal ratio* measures efficiency by normalizing the actual traversal time against an ideal, unhindered straight-line path at a nominal preferred velocity $v_{\text{pref}} = 1.1$ m/s. *Personal space violation rate (PSV)* quantifies social compliance as the number of frames in which any pedestrian center lies within 0.5 m plus the robot radius of the robot center, divided by the trial duration, and reads as violation frames per second. *Robot collisions* count the frames in which the robot disc overlaps any pedestrian disc, and a contact held for 5 s counts 100. Per-event counts are a separate field of the release. Finally, to evaluate driver divergence, *trajectory Hausdorff distance* is used to measure how much an agent's path shifts when only its driver model is modified. While the first four metrics evaluate the robot, the trajectory distance serves as the core metric for our driver divergence analysis.

b) Protocol: A trial denotes one simulation run of a fixed scenario. Each trial is initialized with a given seed that determines pedestrians' initial states, stochastic parameters, and behavior. To isolate the effect of the pedestrian driver, we execute paired runs that share the same seed and scenario but differ only in the pedestrian driver. The driver divergence study evaluates 20 seeds per (scenario, driver) pair across all 20 scenarios. The closed-loop evaluation uses 10 paired-seed trials per (policy, scenario, driver) combination, chosen

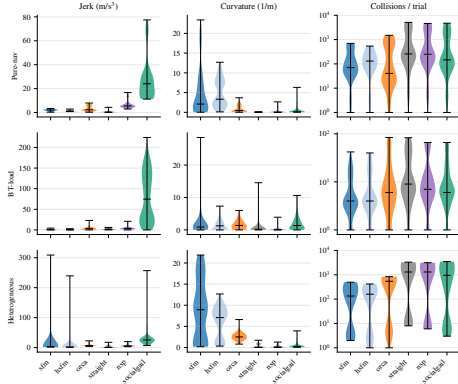


Fig. 4: Pedestrian jerk, path curvature, and collisions between agents per trial, for each driver. The heterogeneous row comes from the closed-loop runs with a robot, the other two rows from runs without one.

to reveal ranking differences at the scenario level. We report cell means with 95% percentile bootstrap CIs (10^4 resamples over seeds), propagated to Figure 5. The divergence analysis instead uses a 90% *scenario-clustered* bootstrap on the ratio metric, capturing across-scenario variability. More seeds do not tighten the nav interval, which is $[1.24, 2.07]$ at 1 seed against $[1.23, 2.02]$ at 20 seeds. Its width reflects the 9 resampled scenarios, which makes the scenario count the limiting factor. The estimate of K also stays within 0.03 of its 20-seed value from 5 seeds on in both buckets, and single-scenario bootstraps over the 20 trials give 3.03 $[2.91, 3.17]$ for the dense bottleneck and 0.90 $[0.88, 0.93]$ for the sitting scenario. For the closed-loop sweep, all 252 subsets of 5 of its 10 seeds move the per-(policy, bucket) success rate by 0.009 on average and the collision count by 4.7%. Because these subsets share half their trials with the full sample, this is a stability check only. With 9 and 11 scenario clusters the percentile bootstrap can understate interval width, which qualifies every class-level interval below.

Ranking flips are tested per driver with a paired sign-flip test over trials and Holm–Bonferroni correction across the six drivers at $\alpha = 0.05$.

B. Driver-class behavioral divergence

To evaluate the effect of different drivers on pedestrian behavior, we analyze both the trajectory distances between paired trials that differ only in the driver and the kinematic distributions of drivers without robot interference. We quantify trajectory differences using the Hausdorff distance and summarize the effect at the class level via the divergence ratio $K = \bar{d}_{\text{across-class}} / \bar{d}_{\text{within-class}}$, where $\bar{d}_{\text{across-class}}$ and $\bar{d}_{\text{within-class}}$ denote distances between drivers of different and same classes, respectively. Distances are computed between the raw recorded (x, y) point sets of the same agent under two drivers, without time alignment, truncation, or resampling. Under the four driver classes of Section III, the within-class pairs are (SFM, HSFM) and (NSP, SocialGAIL). The remaining 13 driver pairs are across-class.

For a reproducible definition of K , we formalize the two mean distances as follows. For

$c \in \{\text{within-class}, \text{across-class}\}$, let $\mathcal{R}_s^c$ be the set of rows $(g, i, \{p, q\})$ in scenario s whose driver pair $\{p, q\}$ belongs to c and whose agent i was recorded under both drivers at seed g . Then

$$\bar{d}_c = \frac{1}{|S|} \sum_{s \in S} \frac{1}{|\mathcal{R}_s^c|} \sum_{(g, i, \{p, q\}) \in \mathcal{R}_s^c} d_H(X_{s, g, i}^p, X_{s, g, i}^q), \quad (1)$$

where $X_{s, g, i}^p$ is the point set of agent i under driver p , d_H is the symmetric Hausdorff distance, and S is the scenario set of the bucket. The mean weights scenarios and intra-scenario rows equally, which means that a driver pair with more matched agents also carries more weight. K is the ratio of the two means and not a mean of per-scenario ratios.

a) *Driver divergence depends on scenario locomotion demand*: K follows the kinematic demand of the scenario. In dense navigation scenarios with continuous motion and collision avoidance it reaches 3.03, in scenarios dominated by stationary behaviors such as sitting or queuing it stays near 1: driver class perturbs trajectories in proportion to how much locomotion the scenario requires.

b) *Bucket means are a robustness floor*: When averaged across all scenarios, $K \geq 1$ consistently, with $K_{\text{nav}} = 1.63$ (90% bootstrap CI $[1.23, 2.02]$), and $K_{\text{BT}} = 1.24$ (90% bootstrap CI $[1.10, 1.46]$), indicating that driver choice has a significant influence on crowd trajectories. The underlying mean distances are 2.86 m within and 4.66 m across classes on nav, against 1.91 m and 2.38 m on BT. Removing one driver at a time keeps K_{nav} between 1.40 and 1.82, and the interval excludes 1 in five of six cases. The exception is the removal of SocialGAIL (1.40, $[0.88, 1.99]$), the driver the nav ratio is most sensitive to. On BT the ratio depends on which within-class pair remains: without either force-based driver it drops to 0.86, because the within mean then rests on the learned pair alone, and without either learned driver it rises above 2.2. Of the 13 across-class pairs, 8 contain SFM or HSFM. On nav every such pair lies between 5.18 m and 7.44 m while the other five lie between 1.47 m and 4.42 m. The nav numerator is thus set by the distance of the force-based drivers to every other driver. Detailed per-pair distances are reported in Table II.

c) *Within-class divergence*: Membership in a shared model class does not guarantee behavioral similarity. Pooled over the release, the learned pair (NSP, SocialGAIL) diverges more (2.80 m, CI $[2.24, 3.37]$) than the force-based pair (HSFM, SFM) at 1.98 m, CI $[1.34, 2.68]$, and on BT the gap is 2.84 m against 0.99 m. On nav the order reverses (2.75 m against 3.20 m), and the intervals overlap in every bucket. The within-class evidence rests on these two pairs and does not rank the classes. It shows that the force-based pair is homogeneous on BT where the learned pair is not, and that a learned driver cannot stand in for its class.

d) *Kinematic signatures explain divergence structure*: Figure 4 shows per-driver kinematic distributions. SocialGAIL exhibits a median jerk nearly an order of magnitude higher than all other drivers ($\sim 30 \text{ m/s}^3$ vs. $\leq 5 \text{ m/s}^3$), indicating highly reactive, rapidly changing motion. The real datasets provide no jerk reference, since ETH is annotated at 2.5 Hz and ATC is sensor-tracked, which leaves third differences of position dominated by sampling and tracking noise. We

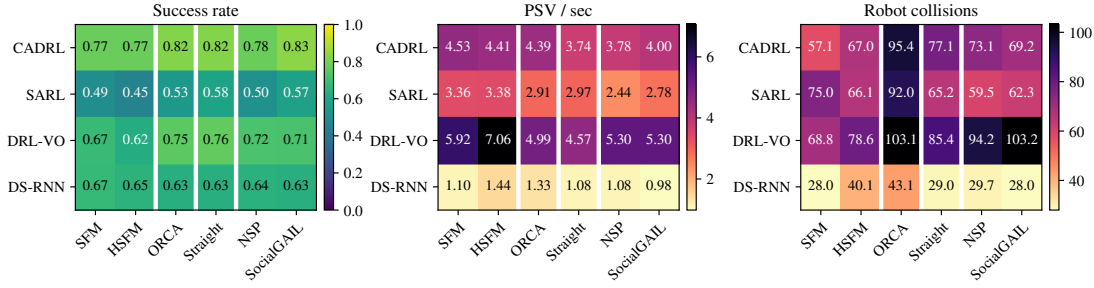


Fig. 5: Sensitivity of robot policies to pedestrian driver class. Per-(policy, driver) cells on three closed-loop metrics (success rate, PSV/sec (personal space violation rate), robot collisions/trial). Results are aggregated across the closed-loop sweep weighted by trial count, columns are grouped by driver class. Success rate and collisions do not rank the policies equally under each driver.

TABLE II: Within-class trajectory Hausdorff per pair and bucket. Mean and scenario-clustered 90% bootstrap CI over n scen. scenarios. ORCA and Straight are singleton classes without a within-class pair. Pairs are matched by agent id where both trials released the agent within 60 s. Drivers whose agents exit later release fewer agents under the concurrency cap, hence the differing counts.

Bucket	Class	Pair	Mean (m)	CI lo	CI hi	n pairs	n scen.
nav	force-based	HSFM SFM	3.20	2.07	4.45	11831	9
nav	learned	NSP SocialGAIL	2.75	1.85	3.82	17733	9
bt	force-based	HSFM SFM	0.99	0.65	1.34	1860	11
bt	learned	NSP SocialGAIL	2.84	2.23	3.52	1860	11
pooled	force-based	HSFM SFM	1.98	1.34	2.68	13691	20
pooled	learned	NSP SocialGAIL	2.80	2.24	3.37	19593	20

treat the SocialGAIL jerk as a property of the driver, the largest single-driver kinematic outlier of the release, and not as a human-like value. Force-based drivers (SFM, HSFM) show wider curvature tails from late repulsive corrections during close interactions, whereas velocity-obstacle and learned drivers follow smoother paths. Non-avoidance and learned drivers (Straight, NSP, SocialGAIL) exhibit collision counts 3 to 7 times higher than avoidance-capable drivers.

We further compare per-driver speed, clearance, and turn-rate distributions against the ETH [11] and ATC [25] real-world pedestrian datasets (Table III). SFM, HSFM, ORCA, and SocialGAIL overlap the interquartile range of both datasets on all three statistics. Straight and NSP fall outside both datasets on their measured turn rate. NSP runs SDD-trained checkpoints, and both datasets are held out for it. Straight overlaps the real clearance band because clearance in these scenarios is set by spawn density and layout. Its missing avoidance shows in its contact rate instead. Every driver's median clearance lies below both real medians, which reflects the deliberately denser scenario design. No driver matches the real data on all three axes, and the six drivers span the real-data support.

e) *Driver effect relative to seed variation:* Finally, we compare the effect of changing the driver to the effect of changing the initialization seed. We define $K_{\text{seed}} := \bar{d}_{\text{across-class}} / \bar{d}_{\text{self}}$, where $\bar{d}_{\text{self}}$ is the distance between trajectories of the same agent under the same driver but different seeds. Under pure-navigation settings $K_{\text{seed}} = 0.75$ (90% CI [0.52, 0.99]), because a new seed redraws the whole crowd and therefore moves trajectories more than a change of drivers.

TABLE III: Real-data comparison: median (interquartile range) per source. Clearance is the nearest-neighbor distance per agent and frame. Turn rate is the heading change between consecutive 0.8 s chords divided by 0.8 s. Sim rows pool the 400 divergence trials of each driver.

Source	Speed (m/s)	Clearance (m)	Turn rate (rad/s)
ETH	1.39 (0.79, 1.68)	1.12 (0.79, 2.18)	0.09 (0.04, 0.18)
ATC	1.10 (0.66, 1.38)	1.71 (0.91, 3.23)	0.20 (0.09, 0.45)
SFM	0.58 (0.07, 1.00)	0.84 (0.60, 1.16)	0.11 (0.02, 0.47)
HSFM	0.32 (0.06, 0.85)	0.71 (0.56, 0.98)	0.11 (0.02, 0.29)
ORCA	1.03 (0.11, 1.24)	0.68 (0.51, 1.06)	0.02 (0.00, 0.13)
Straight	1.10 (0.93, 1.26)	0.46 (0.24, 0.88)	0.00 (0.00, 0.00)
NSP	1.04 (0.64, 1.21)	0.49 (0.26, 0.94)	0.00 (0.00, 0.00)
SocialGAIL	1.34 (0.75, 1.59)	0.61 (0.31, 1.12)	0.05 (0.02, 0.14)

TABLE IV: Policy success rate by scenario regime, marginalized over the six pedestrian drivers. Cells show mean $\pm$ 95% bootstrap CI over 10 paired-seed trials per (policy, scenario, driver), aggregated across the scenarios in each (load $\times$ density) cell of the 21-scenario grid. Bold entries are per-column bests across the four reactive policies.

Policy	Pure-navigation		Behavior-tree-loaded	
	Sparse	Dense	Sparse	Dense
CADRL	.74 $\pm$.03	.61 $\pm$.05	1.00 $\pm$.00	1.00 $\pm$.00
SARL	.53 $\pm$.02	.53 $\pm$.02	.61 $\pm$.02	.33 $\pm$.03
DS-RNN	.42 $\pm$.04	.28 $\pm$.03	1.00 $\pm$.00	.98 $\pm$.02
DRL-VO	.67 $\pm$.01	.43 $\pm$.02	.98 $\pm$.01	.85 $\pm$.04

Under BT settings the driver effect is the larger one ($K_{\text{seed}} = 1.43$, CI [1.02, 1.96]). Seeds change who is in the crowd, the paired-seed protocol holds the crowd fixed, and K measures the driver effect at that fixed crowd. Hence $K_{\text{seed}} < 1$ on nav bounds the driver effect against seed variation.

C. Robot policies benchmark

a) *Success rate:* Table IV reports the success rates for each policy across different scenario regimes, aggregated over all pedestrian drivers. CADRL consistently maintains the highest performance across all benchmarks. While DS-RNN exhibits competitive efficacy in structured behavior-tree scenarios, its performance degrades noticeably in pure-navigation tasks. SARL demonstrates moderate proficiency but suffers from high performance variance. DRL-VO is competitive and ranks second in sparse nav, with dense nav being its weakest

TABLE V: *Success rate per driver in dense navigation.* Mean over the four dense nav scenarios and 10 paired seeds (40 trials per cell). Spread is the difference between the best and the worst driver.

Policy	SFM	HSFM	ORCA	Straight	NSP	SocialGAIL	Spread
CADRL	.57	.57	.68	.60	.55	.70	.15
SARL	.55	.57	.47	.50	.50	.57	.10
DS-RNN	.35	.35	.25	.28	.25	.23	.12
DRL-VO	.30	.25	.50	.50	.47	.55	.30

(0.43). On successful trials, CADRL also reaches the goal fastest, with a time-to-goal ratio of 0.97 on nav and 0.83 on BT scenarios. DRL-VO follows at 1.09 in both regimes, then DS-RNN at 1.52 and 1.33, and SARL at 1.66 and 1.47.

b) *Sensitivity to pedestrian driver:* Figure 5 shows per-driver success rate, PSV, and robot collisions per trial for each policy.

Driver choice introduces performance variation. CADRL and DS-RNN remain relatively stable (within 6 percentage points in success rate), whereas SARL and DRL-VO vary by up to 13 and 14 percentage points across drivers, and in dense navigation the spread of DRL-VO reaches 30 percentage points, from 0.25 under HSFM to 0.55 under SocialGAIL (Table V).

For instance, in the trial of Figure 3 (b) and (c), CADRL reaches the goal among ORCA pedestrians and stalls among SFM pedestrians under the same seed.

Different metrics expose different sensitivities. Success rate is policy-dominated, with CADRL leading and DRL-VO second under all but one driver. PSV varies by up to $1.5\times$ over drivers. Robot collision counts depend most on the driver, by up to $1.7\times$ for one policy: ORCA induces the highest counts and SFM the fewest under the same policy. *Driver choice can change policy ranking on safety metrics.* On robot collision counts, the ranking between CADRL and SARL depends on the driver: under SFM, the ordering is DS-RNN > CADRL > DRL-VO > SARL, whereas under other drivers it becomes DS-RNN > SARL > CADRL > DRL-VO. The same instability appears in success rate on nav scenarios, where SARL outperforms DRL-VO under the force-based drivers (SFM, HSFM), whereas DRL-VO outperforms SARL under ORCA, Straight, and NSP (Figure 3(a)). Per driver, the paired SARL-versus-DRL-VO difference is significant under ORCA only ($p = 0.03$ after Holm–Bonferroni over 110 paired trials per driver). The gap between the force-based and the other drivers is 0.17 with 90% CI [0.09, 0.26] over trials and $[-0.07, 0.43]$ over scenarios. The flip holds for this scenario set and is not a population statement. None of the six per-driver CADRL-versus-SARL collision differences survives correction, which leaves the collision ordering under SFM as a point estimate. Overall, these results show that both the magnitude and the interpretation of policy performance depend on the pedestrian driver, making evaluation under a single fixed pedestrian model insufficient for reliable comparison.

c) *Sensitivity to scenario regime:* Policy ranking is also unstable across scenario regimes. When results are pooled across all scenarios, DS-RNN appears to outperform SARL (Figure 5). However, when restricted to pure-navigation sce-

TABLE VI: Failure breakdown by robot policy and pedestrian load. Bold: column bests among the four reactive policies within each load. Success requires ending at the goal; Table IV counts reaching it at any time, including DS-RNN trials that later drift off.

Load	Policy	Success	Col.-S.	Col.-H.	Freeze	Oversh.	Timeout
nav	CADRL	0.69	0.12	0.08	0.04	0.00	0.06
	SARL	0.53	0.19	0.11	0.17	0.00	0.00
	DS-RNN	0.17	0.22	0.39	0.06	0.05	0.12
	DRL-VO	0.58	0.06	0.14	0.08	0.00	0.14
BT	CADRL	1.00	0.00	0.00	0.00	0.00	0.00
	SARL	0.54	0.01	0.11	0.34	0.00	0.01
	DS-RNN	0.70	0.00	0.14	0.00	0.11	0.05
	DRL-VO	0.94	0.01	0.05	0.00	0.00	0.00

nav: pure navigation; BT: behavior-tree-loaded. Causes are exclusive and assigned in this order: Success (ends within 0.5 m of the goal), Col.-S. and Col.-H. (first frame with disc overlap, static if the hit agent moves below 0.05 m/s), Oversh. (came within 0.5 m and ends beyond 1 m), Freeze (mean speed below 0.05 m/s over the last 5 s), Timeout (the rest).

narios (Table IV), this ordering reverses, with SARL achieving higher success rates. This discrepancy arises because DS-RNN performs strongly in behavior-tree-loaded scenarios, where both CADRL and DS-RNN often saturate at perfect success, while SARL struggles. As a result, aggregated metrics obscure regime-specific behavior and can lead to misleading conclusions about relative performance.

d) *Failure modes:* Table VI breaks failures down by cause, and each policy has a distinct pattern. DRL-VO fails mainly in nav scenarios, through no-progress timeouts and collisions with humans (0.14 each). SARL frequently becomes stuck in behavior-tree scenarios, suggesting that its policy does not generalize well to structured social interactions such as waiting, queuing, or group conversation due to its reward shaping. DS-RNN tends to collide with humans in navigation scenarios. CADRL performs reliably overall but still incurs some collisions in dense navigation scenarios. These patterns indicate that the benchmark successfully exposes systemic vulnerabilities, and that different policies struggle with different aspects of the environments.

V. CONCLUSION

We proposed PEDS, a benchmarking framework that treats pedestrian drivers as a controllable and reportable experimental variable. Through a plug-and-play driver interface and paired-seed execution, PEDS isolates the effect of pedestrian behavior. Varying only the pedestrian driver under otherwise identical conditions alters crowd trajectories, robot policy performance, and relative policy rankings. These results demonstrate that pedestrian models are not passive environment details, but active determinants of evaluation outcomes, challenging the common practice of benchmarking navigation policies against a single driver.

a) *Limitations:* PEDS’s determinism is bounded. The executor removes scheduling and message-ordering nondeterminism and fixes the sampled population across paired runs, which we verified on the released sweeps. Floating-point nondeterminism of the GPU kernels in the learned drivers remains. Bitwise repeatability of whole trajectories was not measured and no second machine was tested. Reproduction elsewhere is

therefore expected only at the level of aggregate statistics. On pure-navigation scenarios, a change of seed moves trajectories more than a change of driver ($K_{\text{seed}} = 0.75$), which makes the driver effect there a statement at a fixed crowd. The class-level intervals rest on 9 and 11 scenario clusters and on two within-class pairs. Correspondence tests cover the drivers Straight, ORCA, and SFM, of which only Straight matches its reference on every scenario, while HSFM, NSP, and SocialGAIL have no steppable public reference. Validation against real pedestrian data is limited to distributional statistics on ETH and ATC, so it does not establish trajectory-level realism.

b) *Future work*: Future work will expand the diversity and realism of the pedestrian suite with richer social behaviors and data-driven interaction models, integrate driver sensitivity directly into evaluation metrics for robustness-aware policy comparison, and extend the framework beyond ROS 2 Jazzy with cross-platform determinism.

REFERENCES

- [1] C. Mavrogiannis, F. Baldini, A. Wang, *et al.*, “Core challenges of social robot navigation: A survey,” *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, 2023.
- [2] P. T. Singamaneni, P. Bachiller-Burgos, L. J. Manso, *et al.*, “A survey on socially aware robot navigation: Taxonomy and future challenges,” *The International Journal of Robotics Research*, vol. 43, no. 10, pp. 1533–1572, 2024.
- [3] A. Francis, C. Pérez-D’Arpino, C. Li, *et al.*, “Principles and guidelines for evaluating social robot navigation algorithms,” *ACM Transactions on Human-Robot Interaction*, vol. 14, no. 2, pp. 1–65, 2025.
- [4] S. Macenski, T. Foote, B. Gerkey, *et al.*, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, 2022.
- [5] D. Helbing and P. Molnár, “Social force model for pedestrian dynamics,” *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.
- [6] F. Farina, D. Fontanelli, A. Garulli, *et al.*, “Walking ahead: The headed social force model,” *PLOS ONE*, vol. 12, no. 1, p. e0169734, 2017.
- [7] J. van den Berg, S. J. Guy, M. Lin, *et al.*, “Reciprocal n-body collision avoidance,” in *Robotics Research: The 14th International Symposium ISRR*, ser. Springer Tracts in Advanced Robotics, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Springer, 2011, vol. 70, pp. 3–19.
- [8] M. Moussaïd, D. Helbing, and G. Theraulaz, “How simple rules determine pedestrian behavior and crowd disasters,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 17, pp. 6884–6888, 2011.
- [9] A. Alahi, K. Goel, V. Ramanathan, *et al.*, “Social lstm: Human trajectory prediction in crowded spaces,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 961–971.
- [10] T. Salzmann, B. Ivanovic, P. Chakravarty, *et al.*, “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data,” in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science. Springer, 2020, pp. 683–700.
- [11] S. Pellegrini, A. Ess, K. Schindler, *et al.*, “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in *2009 IEEE 12th International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 261–268.
- [12] A. Lerner, Y. Chrysanthou, and D. Lischinski, “Crowds by example,” *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, 2007.
- [13] P. Kothari, S. Kreiss, and A. Alahi, “Human trajectory forecasting in crowds: A deep learning perspective,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7386–7400, 2021.
- [14] J. Yue, D. Manocha, and H. Wang, “Human trajectory prediction via neural social physics,” in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, vol. 13664. Springer, 2022, pp. 376–394.
- [15] B. Ling, Y. Lyu, D. Li, *et al.*, “Socialgail: Faithful crowd simulation for social robot navigation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 873–16 880.
- [16] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, 2016, pp. 4565–4573, arXiv:1606.03476.
- [17] L. Kästner, R. Carstens, H. Zeng, *et al.*, “Arena-rosnav 2.0: A development and benchmarking platform for robot navigation in highly dynamic environments,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 11 257–11 264.
- [18] V. Shcherbyna, L. Kästner, D. Diaz, *et al.*, “Arena 4.0: A comprehensive ROS2 development and benchmarking platform for human-centric navigation using generative-model-based environment generation,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 9138–9144.
- [19] L. Kästner, V. Shcherbyna, H. Soh, *et al.*, “Demonstrating arena 5.0: A photorealistic ros2 simulation framework for developing and benchmarking social navigation,” in *Robotics: Science and Systems XXI (RSS)*. Robotics: Science and Systems Foundation, 2025.
- [20] N. Pérez-Higueras, R. Otero, F. Caballero, *et al.*, “Hunavsim: A ros 2 human navigation simulator for benchmarking human-aware robot navigation,” *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7130–7137, 2023.
- [21] A. Biswas, A. Wang, G. Silvera, *et al.*, “Socnavbench: A grounded simulation testing framework for evaluating social navigation,” *ACM Transactions on Human-Robot Interaction*, vol. 11, no. 3, pp. 1–24, 2022.
- [22] N. Tsoi, A. Xiang, P. Yu, *et al.*, “Sean 2.0: Formalizing and generating social situations for robot navigation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 047–11 054, 2022.
- [23] F. Grzeskowiak, D. Gonon, D. Dugas, *et al.*, “Crowd against the machine: A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3879–3885.
- [24] R. Martín-Martín, M. Patel, H. Rezatofighi, *et al.*, “JRDB: A dataset and benchmark of egocentric robot visual perception of humans in built environments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6748–6765, 2023.
- [25] D. Bršićić, T. Kanda, T. Ikeda, *et al.*, “Person tracking in large public spaces using 3-d range sensors,” *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 6, pp. 522–534, 2013.
- [26] H. Li, M. Li, Z.-Q. Cheng, *et al.*, “Human-aware vision-and-language navigation: Bridging simulation to reality with dynamic human interactions,” in *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024, spotlight.
- [27] M. Savva, A. Kadian, O. Maksymets, *et al.*, “Habitat: A platform for embodied AI research,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 9338–9346.
- [28] B. Shen, F. Xia, C. Li, *et al.*, “iGibson 1.0: A simulation environment for interactive tasks in large realistic scenes,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7520–7527.
- [29] W. Wu, H. He, J. He, *et al.*, “MetaUrban: An embodied AI simulation platform for urban micromobility,” in *International Conference on Learning Representations (ICLR)*, 2025, spotlight.
- [30] M. Everett, Y. F. Chen, and J. P. How, “Motion planning among dynamic, decision-making agents with deep reinforcement learning,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.
- [31] C. Chen, Y. Liu, S. Kreiss, *et al.*, “Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning,” in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6015–6022.
- [32] S. Liu, P. Chang, W. Liang, *et al.*, “Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3517–3524.
- [33] Z. Xie and P. Dames, “Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles,” *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2700–2719, 2023.
- [34] U. Weidmann, “Transporttechnik der fussgänger: Transporttechnische eigenschaften des fussgängerverkehrs, literaturauswertung,” Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau (IVT), ETH Zürich, Zürich, IVT Schriftenreihe 90, 1992.
- [35] L. F. Henderson, “The statistics of crowd fluids,” *Nature*, vol. 229, no. 5284, pp. 381–383, 1971.
- [36] E. T. Hall, *The Hidden Dimension*. Garden City, NY: Doubleday, 1966.
- [37] J. Rios-Martinez, A. Spalanzani, and C. Laugier, “From proxemics theory to socially-aware navigation: A survey,” *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, 2015.