Choosing the Arena: A Systematic Review of Simulators for Deep Reinforcement Learning in Mobile Robot Navigation

Zakaria Haja[1]*, Leila Kelmoua[2], Ihababdelbasset Annaki[3], Jamal Berrich[4], Toumi Bouchentouf[5]
Research Laboratory in Applied Sciences-National School of Applied Sciences,
Université Mohammed Premier, Oujda, Morocco[1, 2, 3, 4, 5]
Higher School of Education and Training, Université Mohammed Premier, Oujda, Morocco[3]

***Abstract*—This study presents a formal Systematic Literature Review (SLR) to address a critical methodological question in robotics research: "Which simulator is most suitable for a given Deep Reinforcement Learning (DRL) algorithm and mobile robot navigation task?" The choice of a simulation environment profoundly impacts policy robustness, data efficiency, and sim-to-real transfer, yet the community has lacked an evidence-based guide for this decision. Following PRISMA guidelines, we methodically searched and analyzed 87 peer-reviewed studies published between January 2020 and June 2025 to map the contemporary research landscape. Our synthesis introduces a novel, theory-informed taxonomy that classifies simulators into three archetypes based on their empirical use. Archetype I, ROS-centric standards (e.g., Gazebo), are chosen for algorithmic novelty with low-dimensional sensor inputs. Archetype II, versatile platforms (e.g., CoppeliaSim), are favored for rapid prototyping. Archetype III, GPU-native engines (e.g., NVIDIA Isaac Sim), have emerged for large-scale, perception-heavy challenges, leveraging photorealism and parallelization to mitigate the perception gap and enable zero-shot transfer. This review reveals a paradigm shift towards data-driven methodologies and culminates in a prescriptive decision-making framework, transforming simulator selection from an incidental detail into a strategic choice.**

Keywords*—Simulator; mobile robot; Deep Reinforcement Learning; navigation

I. Introduction

The autonomous navigation of mobile robots remains a cornerstone challenge in robotics, with Deep Reinforcement Learning (DRL) emerging as a powerful paradigm for learning adaptive control policies directly from sensor data [1]. However, the data-intensive nature of DRL makes training on physical hardware impractical due to time, cost, and safety constraints [2], [3], making simulation an indispensable tool. This reliance on simulation introduces the pervasive "sim-to-real gap"—the discrepancy between simulation and reality that hinders policy transfer [4]. The choice of simulator is therefore a foundational methodological decision, as it dictates the available strategies for mitigating this gap, defines the fidelity of sensor models (the "perception gap"), and sets the performance ceiling for data collection, which is critical for modern DRL algorithms [5], [6]. Despite the existence of numerous simulators, researchers lack a systematic, evidence-based guide to inform their selection, often leading to a methodological mismatch between the tool and the research objective. This study addresses this significant void by conducting a formal Systematic Literature Review (SLR) to provide a data-driven answer to the central research question: "Which simulator should a researcher choose for a specific DRL algorithm and mobile robot navigation task?"

While prior surveys have provided excellent overviews of DRL algorithms for navigation [7] or sim-to-real techniques [8], they are largely simulator-agnostic. They identify the sim-to-real gap as a key challenge, but do not offer a comparative analysis of the simulation tools themselves as the primary variable influencing research outcomes. Our work is fundamentally different. By employing a formal SLR methodology focused specifically on the simulator, we move beyond simple cataloging to provide a novel, archetype-based synthesis. This approach allows us to uncover causal links between simulator choice, research focus (e.g., algorithmic novelty vs. perception challenges), and the adopted sim-to-real strategy—insights that remain obscured in conventional narrative reviews. We frame our resulting taxonomy not as a mere descriptive grouping but as a theory-informed abstraction that reveals the underlying structure of current research practices. Furthermore, by analyzing literature from 2020 to 2025, our review captures and contextualizes a critical paradigm shift in the field: the move away from purely CPU-bound simulation towards GPU-native engines and data-driven methodologies for achieving robust real-world deployment.

This review makes several distinct contributions to the field. Our methodological contribution is the application of the rigorous PRISMA framework to a simulator-centric analysis, providing a transparent and reproducible evidence base. Our conceptual contribution is the novel taxonomy of simulator archetypes, which serves as a new analytical lens for understanding research trends and trade-offs. Finally, our practical contribution is a prescriptive, data-driven decision-making framework designed to guide researchers in strategically aligning their choice of simulator with their specific research objectives, thereby enhancing methodological rigor and research efficiency.

To achieve these aims, this study is structured as follows: Section II reviews related survey literature, sharpening the novel positioning of our work. Section III details our systematic review methodology, adhering to the PRISMA

*Corresponding author.

guidelines. Section IV presents the data extraction and synthesis process. In Section V, we present the results, introducing our taxonomy of simulator archetypes and analyzing their use in the literature. Section VI provides a detailed discussion, formalizing our findings into a prescriptive decision-making framework and articulating key research gaps. Finally, Section VII concludes the study with a reflective synthesis of our findings and their implications for future research.

II. Related Work

Several survey papers have explored the intersection of RL and robotics. Seminal works provided a comprehensive overview of RL in robotics before the deep learning era [9]. More recent reviews have focused on DRL for robotics, but often with accomplishments primarily in simulation or with a broad scope covering many tasks like manipulation and locomotion [7]. Other reviews focus specifically on sim-to-real transfer techniques, such as domain randomization and adaptation, but are generally simulator-agnostic [8].

A few studies have compared simulation environments directly. A study in [10] benchmarked the performance of four simulators on different hardware, focusing on computational speed rather than the broader research context. A recent review in [11] also reviewed popular simulation engines, highlighting MuJoCo and Unity, but not focusing specifically on the mobile robot navigation domain. Most closely related are recent reviews such as [5], [7], and [12]. A foundational review of DRL methods and navigation frameworks is provided in [7]. A systematic review focusing on navigation in dynamic environments is offered in [5]. Similarly, a comprehensive overview of DRL's integration in mobile robotics, analyzing its evolution and identifying key challenges and future directions, is presented in [12]. While these studies provide excellent, broad surveys of the algorithmic landscape, they identify the sim-to-real gap as a challenge without offering a comparative analysis of the simulation tools themselves. Our work differentiates itself by using a formal SLR methodology to specifically analyze and synthesize the literature based on the choice of simulator, thereby providing a novel, evidence-based framework that directly links simulator characteristics to DRL algorithms and navigation tasks.

III. Systematic Review Methodology

To ensure a transparent, reproducible, and rigorous review, we adopted the PRISMA 2020 statement as our methodological framework [13]. The process consists of defining research questions, a search strategy, study selection criteria, and a data extraction plan [14].

This review is guided by four central research questions (RQs) designed to systematically map the literature:

- RQ1: What is the distribution of simulators (e.g., Gazebo, Webots, CoppeliaSim, NVIDIA Isaac Sim) used in the recent literature for DRL-based mobile robot navigation?
- RQ2: What are the reported capabilities and limitations of these simulators concerning sensor modeling (perception gap) and performance/scalability for DRL training?
- RQ3: Which DRL algorithms (e.g., PPO, SAC, DQN, DDPG) are predominantly paired with each simulator, and for which specific navigation tasks (e.g., obstacle avoidance, end-to-end visual navigation)?
- RQ4: What trends and patterns emerge from the literature regarding sim-to-real transfer strategies associated with each simulator archetype?

A. Search Strategy

To gather a comprehensive body of literature, we implemented a structured search strategy across several major scientific databases. We conducted a systematic search of three major scientific databases: IEEE Xplore, Scopus, and arXiv. The search was performed in June 2025, covering publications from January 2020 to June 2025 to capture the most recent trends. The following search string was adapted for each database's syntax: ("reinforcement learning" OR "RL") AND ("mobile robot" OR "robot navigation") AND ("simulation" OR "simulator") AND ("Gazebo" OR "Webots" OR "CoppeliaSim" OR "Isaac Sim" OR "PyBullet").

B. Study Selection: Inclusion and Exclusion Criteria

We conducted the study selection in two phases: 1) title and abstract screening, and 2) full-text review[15]. To be included, a paper had to meet all of the following inclusion criteria:

- I1: The paper is a full-text, peer-reviewed conference paper, journal article, or a publicly available preprint (from arXiv).
- I2: The paper was published between January 2020 and June 2025.
- I3: The paper explicitly uses a physics-based simulator for training or evaluating a DRL agent for a mobile robot navigation task.
- I4: The paper is written in English.

Papers were excluded based on the following exclusion criteria:

- E1: The paper is a review, survey, or abstract-only publication.
- E2: The work does not involve a mobile ground robot (e.g., focuses only on manipulators, drones, or underwater vehicles).
- E3: The work uses RL but not deep learning (i.e., no deep neural networks).
- E4: The simulator is used for purposes other than DRL (e.g., only for visualization or classical controller testing).

C. Selection Results

The systematic search and screening process yielded a final corpus of literature that forms the evidence base for this review. Initially, 173 papers were selected based on the research query and inclusion criteria. After the removal of duplicates, 143 unique studies remained. These were subjected to a full-text review, and after applying the exclusion criteria, a final set of 87 papers were deemed eligible for inclusion in our qualitative synthesis. The complete study selection process is visually detailed in the PRISMA flow diagram shown in Fig. 1.

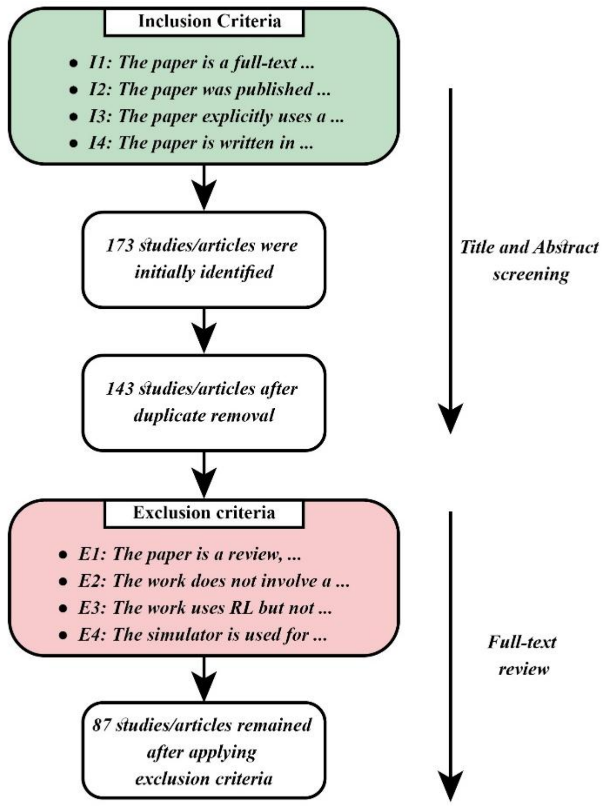

Fig. 1. Study selection process based on inclusion and exclusion criteria.

IV. Data Extraction and Synthesis

For each study that met the criteria, we performed a detailed data extraction process to gather key information for our synthesis. For each included study, we extracted the following data points: 1) Simulator used; 2) Mobile robot platform; 3) Navigation task description; 4) DRL algorithm implemented; 5) Sensor modalities simulated; 6) Stated reasons for simulator choice; 7) Reported performance metrics; and 8) Sim-to-real transfer methodology and results. The extracted data was then synthesized qualitatively to answer the research questions and identify overarching trends. Table I provides a summary of a representative selection of the reviewed literature, illustrating the connections between simulators, algorithms, and tasks that form the basis of our analysis.

The data summarized in Table I reveal several distinct trends that underpin our analysis. A clear majority of the studies leverage Gazebo, pairing it with a wide array of DRL algorithms—from value-based methods like D3QN to policy-gradient methods like PPO and DDPG—for tasks centered on map-less navigation and obstacle avoidance. This reinforces its role as a versatile, community-standard testbed. In contrast, papers utilizing NVIDIA Isaac Sim consistently tackle complex, often end-to-end, navigation tasks and are frequently paired with data-hungry algorithms like PPO or advanced methods like Imitation Learning. This highlights its adoption for research pushing the boundaries of sim-to-real transfer and performance.

Webots and CoppeliaSim are used for a range of tasks, including multi-robot systems and learning from demonstration, often employing algorithms like PPO and GAIL, respectively, showcasing their utility in specialized research areas [37]. Collectively, the table illustrates a strong correlation between the choice of simulator and the scope of the research problem, a central theme explored in this review.

TABLE I. Summary of Selected Literature on Simulators for DRL-based Mobile Robot Navigation

Reference	*Used Simulator(s)*	*DRL Algorithm(s)*	*Task(s)*
[16]	Isaac Sim, Gazebo	PPO	End-to-end local planning and obstacle avoidance.
[17]	Gazebo	DDPG, PPO	Autonomous navigation, target seeking and obstacle avoidance.
[18]	Unity, Gazebo	Double DQN	Map-less navigation to a random target.
[19]	Gazebo	SAC, PPO, DDPG, Q-learning	Path planning and obstacle avoidance
[20]	Gazebo	D3QN (Dueling Double Deep Q-Network)	End-to-end navigation and obstacle avoidance in an unknown environment with static and dynamic obstacles.
[21]	Gazebo	DDPG, PPO	Map-free navigation for an omnidirectional robot.
[22]	Gazebo	DDPG (customized with guiding points)	Navigation in crowded environments with dynamic pedestrians.
[23]	Gazebo	Parallel Distributional Actor-Critic Networks	Map-less navigation for terrestrial mobile robots
[24]	Gazebo	DDQN (Double Deep Q-Network)	Map-less navigation and obstacle avoidance for a TurtleBot3 robot.
[25]	Webots	PPO (enhanced with Curriculum Learning)	Shortest path planning for an E-puck robot using IR sensors.
[26]	Webots	DQN, PPO	Pedestrian avoidance for autonomous vehicles in simulated scenarios.
[27]	Webots	DQN, PPO, A2C	The inverted pendulum task
[28]	Webots	DQN, NSQ, DDQN, PPO	Multi-robot swarm navigation.
[29]	CoppeliaSim	GAIL (Generative Adversarial Imitation Learning)	Learning navigation behaviors from expert demonstrations for mobile robots.
[30]	CoppeliaSim	Adaptive TD3	Navigation in dynamic environments
[31]	Gazebo (via FRobs RL)	PPO, SAC, TD3	Map-less navigation for a mobile robot.
[32]	Isaac Sim	RMA (Rapid Motor	Legged mobile robot walking on different

Reference	Used Simulator(s)	DRL Algorithm(s)	Task(s)
		Adaptation)	types of ground surfaces.
[33]	Isaac Sim	NavACL-Q (curriculum learning method with soft actor-critic algorithm)	Map-less navigation for an Automatic Guided Vehicle (AGV) in a warehouse scenario.
[34]	Isaac Sim	Imitation Learning (IL) and Curriculum Learning (CL)	Map-less navigation of a wheeled robot using open-vocabulary 3D scene graphs.
[35]	Isaac Sim	Imitation Learning (IL) with an auto-regressive world model	Generalizable end-to-end navigation.
[36]	Isaac Sim	Asymmetric Actor Critic	Visual goal-tracking
[37]	Gazebo	DDPG	Map-less path planning in an unknown environment
[38]	Gazebo	DQN, DDQN	Map-less navigation and obstacle avoidance
[39]	Custom Simulator	DWA-RL	Mobile robot navigation without prior exploration

V. Results of the Systematic Review

A. Distribution of Simulators

An analysis of the selected corpus of literature reveals a distinct distribution in the adoption of simulation platforms. Gazebo emerged as the predominant simulation environment, utilized in a substantial majority of the reviewed studies, thereby establishing its position as a de facto standard within the research community.

NVIDIA Isaac Sim was identified as the second most prevalent platform, indicating its significant and growing adoption in recent research. Following these, Webots and CoppeliaSim were also frequently employed, albeit to a lesser extent. A smaller subset of the literature utilized alternative simulators, such as PyBullet, particularly in studies concentrating on specialized applications like mobile navigation.

B. Simulator Capabilities, DRL Algorithms, and Tasks

Our synthesis of the literature revealed distinct patterns that link the choice of simulator with its technical capabilities, the selected DRL algorithms, and the complexity of the navigation task. We present these findings through our proposed taxonomy of simulator archetypes.

1) Archetype I: ROS-Centric Open-Source Standards (Gazebo, Webots). These simulators are predominantly selected for their seamless integration with the ROS ecosystem[40]. The literature consistently cites the benefit of architectural parity, where the same ROS2 control software can be deployed in simulation and on the real robot, simplifying the sim-to-real process[16]. However, their CPU-bound performance is a frequently acknowledged limitation, making them less suitable for training on high-dimensional visual data. Sensor modeling is functional, but studies note that default noise parameters often require manual tuning to match real hardware, presenting a perception gap challenge[41]. Consequently, the majority of studies using these platforms focus on navigation tasks with low-dimensional inputs, primarily 2D LiDAR scans[24]. The most common DRL algorithms are off-policy methods like DDPG/TD3 and SAC, alongside the on-policy PPO[24]. The research contribution in these papers often lies in algorithmic enhancements, such as novel reward shaping [37] or modified network architectures for safer navigation, rather than tackling the perception gap itself.

2) Archetype II: The Versatile Multi-Physics Platform (CoppeliaSim) CoppeliaSim's unique value proposition in the literature is its versatility. Several papers leverage its support for multiple physics engines to conduct comparative analyses. The high-speed PyRep interface is frequently cited as a key advantage for accelerating DRL training cycles compared to the overhead of ROS-based communication1. The tasks explored in CoppeliaSim are diverse, often involving complex kinematics or prototyping novel robot designs. The DRL algorithms used are varied, with studies employing GAIL for imitation learning [28] and DDPG for control tasks. The focus is often on rapid algorithm development and validation in a flexible environment.

3) Archetype III: The GPU-Native High-Performance Engine (NVIDIA Isaac Sim) Papers using Isaac Sim are almost exclusively focused on tackling large-scale DRL problems and bridging the sim-to-real gap for perception-heavy tasks. The ability to run thousands of parallel simulations on the GPU is the most cited advantage, as it enables the collection of massive datasets required to train robust policies[42]. Its use of RTX rendering for photorealistic sensor data is highlighted as a direct strategy to mitigate the perception gap, making it the platform of choice for end-to-end, vision-based navigation[42]. The dominant algorithm in the Isaac Sim literature is PPO, whose on-policy nature benefits immensely from the massive parallel data collection capabilities. The tasks are ambitious, often involving navigation in complex, cluttered environments using only camera images or a fusion of camera and LiDAR data, with the primary research contribution often being the demonstration of successful zero-shot sim-to-real transfer.

C. Sim-to-Real Transfer Strategies

Our review confirms that the choice of simulator strongly influences the sim-to-real strategy.

With Gazebo/Webots, the strategy is an architectural alignment. Researchers leverage the native ROS integration to ensure the software stack is nearly identical between simulation and the real robot, minimizing transfer errors related to control and communication logic.

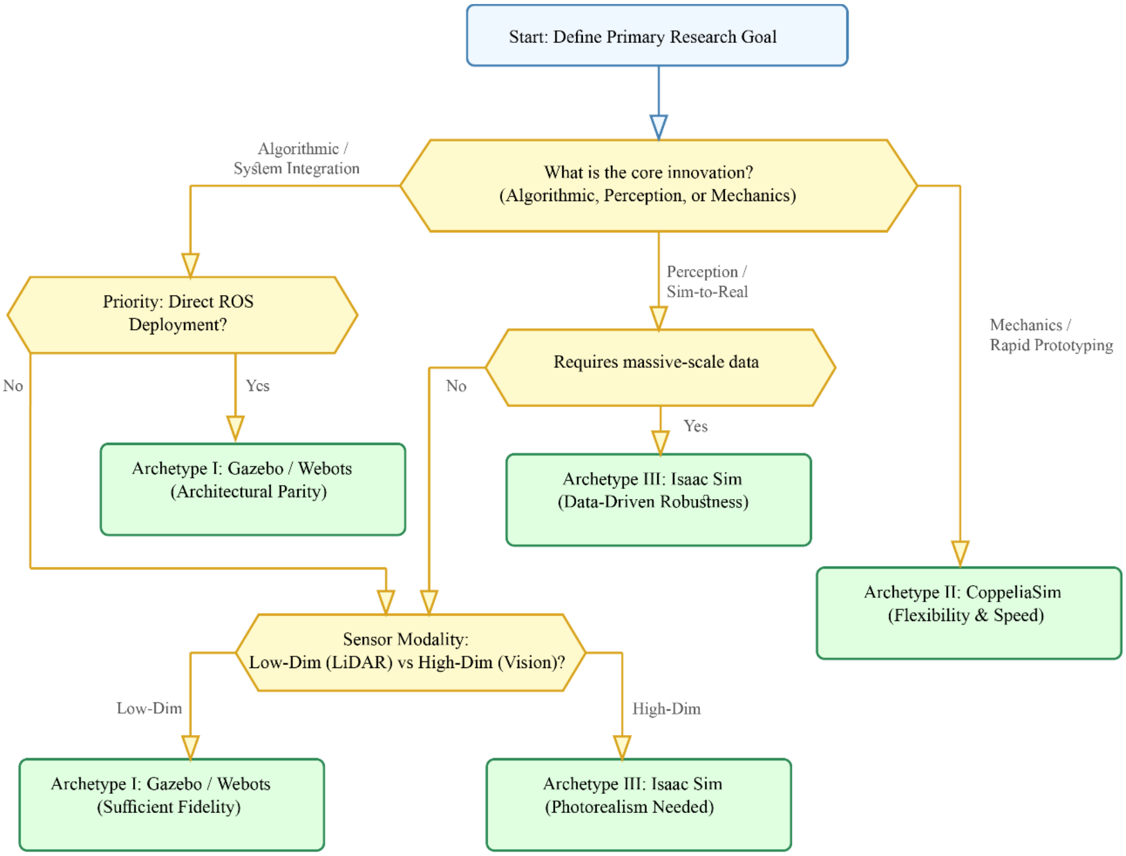

Fig. 2. A prescriptive decision flowchart for simulator selection. This flowchart formalizes the selection logic by guiding a researcher from their primary research goal through key technical requirements to the most appropriate simulator archetype, highlighting the core trade-offs at each stage.

With NVIDIA Isaac Sim, the strategy is data-driven robustness. Researchers leverage massive parallelization and photorealistic rendering to train policies on an extremely wide distribution of scenarios via domain randomization. This forces the policy to become invariant to simulation-specific artifacts, making it robust enough for zero-shot transfer. A powerful emerging trend is the "sim-to-sim-to-real" pipeline, where policies trained in Isaac Sim are first validated in Gazebo before real-world deployment, providing strong evidence of their generalization capabilities.

VI. Discussion: A Decision-Making Framework

Based on our comprehensive literature synthesis, we propose a structured decision-making framework (see Fig. 2) to guide the selection of robotic simulators according to two pivotal dimensions: 1) the primary source of methodological innovation, and 2) the complexity of required sensory data.

When the research novelty lies in algorithmic improvements—such as novel reward functions, enhanced safety-aware exploration strategies [43], or gains in sample efficiency—and the underlying task can be addressed with low-dimensional sensor inputs (for example, 2D LiDAR), established platforms like Gazebo and Webots are recommended (see Fig. 2). These environments not only offer robust, community-validated benchmarks, but their seamless ROS integration also streamlines the transition from simulation to real-world deployment under manageable computational loads.

In contrast, if the principal innovation revolves around comparative physics analyses or the rapid prototyping of intricate mechanical systems, CoppeliaSim emerges as the superior choice. Its unique support for multiple physics engines, combined with the high-speed PyRep API, accelerates development cycles beyond what ROS-centric alternatives typically allow.

Finally, for investigations explicitly targeting sim-to-real transfer—particularly those focused on vision-based, end-to-end navigation in visually complex domains—NVIDIA Isaac Sim provides indispensable capabilities. Its photorealistic rendering and exceptional parallel training performance make it the de facto standard for experiments aiming to achieve zero-shot policy transfer. Through a sim-to-sim-to-real validation pipeline, researchers can rigorously demonstrate methodological advances in direct policy transfer, thus reinforcing the scientific contribution of their work.

VII. Conclusion and Future Directions

This systematic literature review has provided a rigorous, evidence-based analysis of the simulator landscape for DRL-based mobile robot navigation. The central, definitive insight from our synthesis of 87 recent studies is that simulator selection is not an incidental technical detail but a foundational methodological decision that shapes research outcomes. Our work conclusively changes how this choice should be viewed: from a matter of convenience to a strategic alignment of tool, task, and objective. We have formalized the community's implicit practices into an explicit taxonomy of simulator archetypes, demonstrating a clear correlation between a simulator's architectural principles and its application in the literature.

What this review definitively establishes is a tripartite structure of the current research landscape. First, ROS-centric platforms like Gazebo are the established workhorses for algorithmic development with low-dimensional sensors, where architectural parity with the real world is the primary sim-to-real strategy. Second, GPU-native engines, epitomized by NVIDIA Isaac Sim, have become the standard for tackling the perception gap in vision-based navigation, enabling data-driven robustness through massive parallelization and domain randomization. Third, versatile engines like CoppeliaSim occupy a vital niche for rapid prototyping and specialized mechanical analysis. By making these distinctions explicit, our framework provides immediate, practical guidance that can prevent methodological mismatches and enhance research efficiency.

However, our review also illuminates what remains uncertain. While we identify a strong trend towards data-driven sim-to-real transfer, the precise metrics for quantifying the "reality gap" remain ad-hoc and study-specific. It is still an open question how to create a universal benchmark that can compare the transferability of policies from different simulators in a scientifically rigorous manner. Furthermore, the long-term viability and interoperability of vendor-specific, high-performance ecosystems versus open-source standards is an unresolved tension that will shape the field's future accessibility and direction.

Looking forward, the capabilities of Archetype III simulators serve as a crucial technological bridge to the next paradigm: persistent, bi-directional Robotic Digital Twins. Yet, to realize this future and improve scientific rigor, the community must move beyond the current state. We strongly advocate for the development of a Standardized Sim-to-Real Navigation Benchmark. Such a framework would enable true apples-to-apples comparisons, fostering more generalizable and robust solutions. By clarifying the present state of simulation, separating validated insights from open questions, and illuminating the path toward standardized benchmarking, this work aims to accelerate the development of intelligent systems capable of operating safely and effectively in the complexity of the real world.

References

[1] R. Hoseinnezhad, "A Comprehensive Review of Deep Learning Techniques in Mobile Robot Path Planning: Categorization and Analysis," Appl. Sci. 2025, Vol. 15, Page 2179, vol. 15, no. 4, p. 2179, Feb. 2025, doi: 10.3390/APP15042179.

[2] Y. Li, B. W. Min, and H. Liu, "Deep Q-Learning-Based Optimization of Path Planning and Control in Robotic Arms for High-Precision Computational Efficiency," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 1, pp. 1199–1207, Jan. 2025, doi: 10.14569/IJACSA.2025.01601115.

[3] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey," 2020 IEEE Symp. Ser. Comput. Intell., pp. 737–744, Jul. 2021, doi: 10.1109/SSCI47803.2020.9308468.

[4] T. Chaffre, J. Moras, A. Chan-Hon-Tong, and J. Marzat, "Sim-to-real transfer with incremental environment complexity for reinforcement learning of depth-based robot navigation," ICINCO 2020 - Proc. 17th Int. Conf. Informatics Control. Autom. Robot., pp. 314–323, 2020, doi: 10.5220/0009821603140323.

[5] Y. Zhu, W. Z. Wan Hasan, H. R. Harun Ramli, N. M. H. Norsahperi, M. S. Mohd Kassim, and Y. Yao, "Deep Reinforcement Learning of Mobile Robot Navigation in Dynamic Environment: A Review," Sensors (Basel)., vol. 25, no. 11, p. 3394, Jun. 2025, doi: 10.3390/S25113394.

[6] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research," Sensors 2019, Vol. 19, Page 648, vol. 19, no. 3, p. 648, Feb. 2019, doi: 10.3390/S19030648.

[7] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," Tsinghua Sci. Technol., vol. 26, no. 5, pp. 674–691, Oct. 2021, doi: 10.26599/TST.2021.9010012.

[8] D. Han, B. Mulyana, V. Stankovic, and S. Cheng, "A Survey on Deep Reinforcement Learning Algorithms for Robotic Manipulation," Sensors 2023, Vol. 23, Page 3762, vol. 23, no. 7, p. 3762, Apr. 2023, doi: 10.3390/S23073762.

[9] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," Int. J. Rob. Res., vol. 32, no. 11, pp. 1238–1274, Sep. 2013.

[10] M. Körber, J. Lange, S. Rediske, S. Steinmann, and R. Glück, "Comparing Popular Simulation Environments in the Scope of Robotics and Reinforcement Learning," Mar. 2021, [Online]. Available: https://arxiv.org/pdf/2103.04616

[11] M. Kaup, C. Wolff, H. Hwang, J. Mayer, and E. Bruni, "A Review of Nine Physics Engines for Reinforcement Learning Research," Aug. 2024, [Online]. Available: https://arxiv.org/pdf/2407.08590v1

[12] Z. Haja et al., "Deep Reinforcement Learning for Mobile Robots: Overview and Issues," Lect. Notes Networks Syst., vol. 1100 LNNS, pp. 523–532, 2024, doi: 10.1007/978-3-031-68660-3_48.

[13] M. J. Page et al., "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," BMJ, vol. 372, Mar. 2021, doi: 10.1136/BMJ.N71.

[14] S. Leone, F. Lago, D. Pisla, and G. Carbone, "A Systematic Approach for Robotic System Development," Technol. 2025, Vol. 13, Page 316, vol. 13, no. 8, p. 316, Jul. 2025, doi: 10.3390/TECHNOLOGIES13080316.

[15] A. Carrera-Rivera, W. Ochoa, F. Larrinaga, and G. Lasa, "How-to conduct a systematic literature review: A quick guide for computer science research," MethodsX, vol. 9, p. 101895, Jan. 2022, doi: 10.1016/J.MEX.2022.101895.

[16] S. Salimpour, J. Peña, P. Peña-Queralta, D. Paez-Granados, J. Heikkonen, and T. Westerlund, "Sim-to-Real Transfer for Mobile Robots with Reinforcement Learning: from NVIDIA Isaac Sim to Gazebo and Real ROS 2 Robots," Jan. 2025, Accessed: Jan. 25, 2025. [Online]. Available: https://arxiv.org/abs/2501.02902v1

[17] H. Taheri, S. R. Hosseini, and M. A. Nekoui, "Deep Reinforcement Learning with Enhanced PPO for Safe Mobile Robot Navigation," May 2024, [Online]. Available: https://arxiv.org/pdf/2405.16266

[18] E. Marchesini and A. Farinelli, "Discrete Deep Reinforcement Learning for Mapless Navigation", [Online]. Available: http://wiki.ros.org/move

[19] T. Saravanan et al., "'Deep Reinforcement Learning for Autonomous Navigation in Unknown Environments,'" J. Inf. Syst. Eng. Manag., vol. 10, no. 22s, pp. 782–793, Mar. 2025, doi: 10.52783/JISEM.V10I22S.3620.

[20] R. Van Hoa, L. K. Lai, and L. T. Hoan, "Mobile Robot Navigation Using Deep Reinforcement Learning in Unknown Environments," Int. J. Electr. Electron. Eng., vol. Volume 7, no. 8, pp. 15–20, Sep. 2020, doi: 10.14445/23488379/IJEEE-V7I8P104.

[21] M. Heidari, S. Ali, and K. Nylaender, "Autonomous Robot Navigation Using Deep Reinforcement Learning in Office Environments," 2025, [Online]. Available: https://oda.oslomet.no/oda-xmlui/handle/11250/3206164

[22] G. Yang and Y. Guo, "Deep Reinforcement Learning Based Mobile Robot Navigation in Crowd Environments," 2024 21st Int. Conf. Ubiquitous Robot. UR 2024, pp. 513–519, 2024, doi: 10.1109/UR61395.2024.10597481.

[23] V. A. Kich et al., "Parallel Distributional Deep Reinforcement Learning for Mapless Navigation of Terrestrial Mobile Robots," Int. Conf. Control. Autom. Syst., pp. 978–983, 2024, doi: 10.23919/ICCAS63016.2024.10773323.

[24] M. F. R. Lee and S. H. Yusuf, "Mobile Robot Navigation Using Deep Reinforcement Learning," Process. 2022, Vol. 10, Page 2748, vol. 10, no. 12, p. 2748, Dec. 2022, doi: 10.3390/PR10122748.

[25] A. Iskandar and B. Kovács, "Investigating the Impact of Curriculum Learning on Reinforcement Learning for Improved Navigational Capabilities in Mobile Robots," Intel. Artif., vol. 27, no. 73, pp. 163–176, Mar. 2024, doi: 10.4114/INTARTIF.VOL27ISS73PP163-176.

[26] C. Trararak, T.-T. Hoang, and P. Cong-Kha, "Pedestrian Avoidance Simulation by Deep Reinforcement Learning Using Webots," pp. 0734–0739, Mar. 2025, doi: 10.1109/ICAIIC64266.2025.10920794.

[27] A. Šareiko, D. Mažeika, and A. Laukaitis, "Framework for deep reinforcement learning in Webots virtual environments," New Trends Comput. Sci., vol. 3, no. 1, pp. 49–63, Jun. 2025, doi: 10.3846/NTCS.2025.23668.

[28] I. Alaa and K. Béla, "Curriculum learning for deep reinforcement learning in swarm robotic navigation task," Multidiszcip. Tudományok, vol. 13, no. 3, pp. 175–187, Dec. 2023, doi: 10.35925/J.MULTI.2023.3.18.

[29] K. Levrone Rodrigues Machado Silva and R. Calvo, "Mobile Robot Navigation Strategies Through Behavioral Cloning and Generative Adversarial Imitation Learning", doi: 10.5220/0011856700003467.

[30] D. K. Sahoo, "Adaptive TD3: Robust Mobile Robot Navigation in Dynamic Environments with Mixed Obstacles," Recent Trends Autom. Automob. Eng., vol. 6, no. 2, pp. 4–18, 2023, doi: 10.5281/ZENODO.8004578.

[31] J. M. Fajardo, F. G. Roldan, S. Realpe, J. D. Hernández, Z. Ji, and P.-F. Cardenas, "FRobs RL: A Flexible Robotics Reinforcement Learning Library", [Online]. Available: https://github.com/jmfajardod/frobs

[32] X. Hu et al., "Impact of Static Friction on Sim2Real in Robotic Reinforcement Learning," Mar. 2025, [Online]. Available: https://www.arxiv.org/pdf/2503.01255

[33] H. Xue, B. Hein, M. Bakr, G. Schildbach, B. Abel, and E. Rueckert, "Using Deep Reinforcement Learning with Automatic Curriculum Learning for Mapless Navigation in Intralogistics," Appl. Sci. 2022, Vol. 12, Page 3153, vol. 12, no. 6, p. 3153, Mar. 2022, doi: 10.3390/APP12063153.

[34] N. Oskolkov, H. Zhang, D. Makarov, D. Yudin, and A. Panov, "SGN-CIRL: Scene Graph-based Navigation with Curriculum, Imitation, and Reinforcement Learning," Jun. 2025, [Online]. Available: https://arxiv.org/pdf/2506.04505v1

[35] W. Liu et al., "X-MOBILITY: End-To-End Generalizable Navigation via World Modeling," Jul. 2025, [Online]. Available: https://arxiv.org/pdf/2410.17491v2

[36] S. Zhu, L. Mou, D. Li, B. Ye, R. Huang, and H. Zhao, "VR-Robo: A Real-to-Sim-to-Real Framework for Visual Robot Navigation and Locomotion," p. 2025, Jun. 2025, doi: 10.1109/LRA.2025.3575648.

[37] N. Botteghi, B. Sirmacek, J. Ai, K. A. A. Mustafa, M. Poel, and S. Stramigioli, "On Reward Shaping for Mobile Robot Navigation: A Reinforcement Learning and SLAM Based Approach," Feb. 2020, [Online]. Available: https://arxiv.org/pdf/2002.04109

[38] L. D. de Moraes et al., "Double Deep Reinforcement Learning Techniques for Low Dimensional Sensing Mapless Navigation of Terrestrial Mobile Robots," Lect. Notes Networks Syst., vol. 715 LNNS, pp. 156–165, 2023, doi: 10.1007/978-3-031-35507-3_16.

[39] L. Zhang, Z. Hou, J. Wang, Z. Liu, and W. Li, "Robot Navigation With Reinforcement Learned Path Generation and Fine-Tuned Motion Control," IEEE Robot. Autom. Lett., vol. 8, no. 8, pp. 4489–4496, Aug. 2023, doi: 10.1109/LRA.2023.3284354.

[40] S. Fischer, S. Hyeon Hong, Y. Wang, M. Aljamal, S. Patel, and A. Mahmood, "Comprehensive Review of Robotics Operating System-Based Reinforcement Learning in Robotics," Appl. Sci. 2025, Vol. 15, Page 1840, vol. 15, no. 4, p. 1840, Feb. 2025, doi: 10.3390/APP15041840.

[41] T. Zhang and H. Mo, "Reinforcement learning for robot research: A comprehensive review and open issues," Int. J. Adv. Robot. Syst., vol. 18, no. 3, 2021, doi: 10.1177/17298814211007305.

[42] W. Lik Hang Kenny, K. Bai, A. Robots, and G. Jianwei Zhang, "A Survey of Robotic Navigation and Manipulation with Physics Simulators in the Era of Embodied AI," May 2025, [Online]. Available: https://arxiv.org/pdf/2505.01458v1

[43] H. Khan et al., "Neuro-Symbolic Reinforcement Learning for Context-Aware Decision Making in Safe Autonomous Vehicles," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 5, pp. 599–609, May 2025, doi: 10.14569/IJACSA.2025.0160558.