Demonstrating DVS: Dynamic Virtual-Real Simulation Platform for Mobile Robotic Tasks

Zijie Zheng[1*]   Zeshun Li[1*]   Yunpeng Wang[1*]   Qinghongbing Xie[1]   Long Zeng[1†]

[1]Tsinghua University   *Equal contribution   †Corresponding author

Dynamic Virtual Real Simulation Platform Website

Fig. 1: Overview of DVS platform, which offers a variety of large-scale indoor scene types and dynamic element plugins on the left, enabling users to construct dynamic environments. In the middle, the platform supports various data types that can be generated, such as RGB, depth, and semantic labels. On the right, the data created using this platform can be applied to train robots for tasks such as navigation, trajectory prediction, and grasping. Through a virtual-real fusion feedback mechanism, the platform allows bidirectional mapping of the states of real and virtual agents, enriching the research scenarios.

Abstract—With the development of embodied artificial intelligence, robotic research has increasingly focused on complex tasks. Existing simulation platforms, however, are often limited to idealized environments, simple task scenarios and lack data interoperability. This restricts task decomposition and multi-task learning. Additionally, current simulation platforms face challenges in dynamic pedestrian modeling, scene editability, and synchronization between virtual and real assets. These limitations hinder real world robot deployment and feedback. To address these challenges, we propose DVS (Dynamic Virtual-Real Simulation Platform), a platform for dynamic virtual-real synchronization in mobile robotic tasks. DVS integrates a random pedestrian behavior modeling plugin and large-scale, customizable indoor scenes for generating annotated training datasets. It features an optical motion capture system, synchronizing object poses and coordinates between virtual and real world to support dynamic task benchmarking. Experimental validation shows that DVS supports tasks such as pedestrian trajectory prediction, robot path planning, and robotic arm grasping, with potential for both simulation and real world deployment. In this way, DVS represents more than just a versatile robotic platform; it paves the way for research in human intervention in robot execution tasks and real-time feedback algorithms in virtual-real fusion environments. More information about the simulation platform is available on https://immvlab.github.io/DVS/.

I. INTRODUCTION

Robot's capabilities are becoming increasingly powerful with advances in perception, decision-making, and execution technologies. These improvements have expanded their potential applications in industrial manufacturing[23, 44], smart homes[24], and other fields[45, 43, 16]. The transition from rule-based operations to end-to-end learning has enabled robots to tackle more complex tasks. However, achieving high efficiency in real world scenarios requires a complete workflow: virtual data collection, simulator training, and real robot deployment. Existing simulation platforms often fail to effectively support this closed-loop research due to their functional limitations.

Data collection in robotics typically relies on two approaches: collecting real world data with physical robots or using virtual agents in simulated environments. Systems such as Mobile ALOHA[6] aim to reduce the cost of collecting data in the real world. However, they still require significant hardware investment and expert labor. Simulators provide task-specific modeling tools for focused research. In contrast, simulation platforms offer a broader framework for

multitask and complex scenario investigations, allowing faster iteration. Platforms such as Habitat[36], iGibson[17, 34], and Arena[12] have facilitated data collection and algorithm training. Nonetheless, their applicability is often constrained by domain-specific assumptions or limited extensibility. Habitat, while extended to support HITL (Human-in-the-Loop)[27] and HRC (Human-Robot Collaboration)[25], focuses mainly on navigation tasks. iGibson enhances data richness and realism through interactive environments but lacks support for dynamic scenarios. Arena specializes in navigation, while tasks like grasping rely on other simulators.

Existing simulation platforms are typically tailored to specific tasks or static environments, limiting their ability to support complex, long-horizon scenarios that demand deep environmental understanding and cross-domain collaboration. For instance, completing a task like *retrieving a bottle from the fridge and placing it on a desk* involves navigation, manipulation, and environment interaction. Current methods decompose such tasks into sub-tasks across multiple simulators, increasing workload and reducing coherence. Furthermore, most platforms lack support for dynamic scenarios, such as modeling pedestrian behaviors, and do not incorporate real world feedback. This limitation exacerbates the sim-to-real gap, often leading to significant performance degradation during deployment.

We propose DVS (Dynamic Virtual-Real Simulation Platform), a novel framework tailored for dynamic, closed-loop robotic research across diverse tasks. DVS addresses existing limitations through three key features. First, it supports complex long-horizon tasks with dynamic pedestrian modeling and flexible indoor scene editing. This enables high-fidelity simulation environments for multi-stage operations. Second, it establishes a virtual-real fusion workflow, combining high-accuracy optical motion capture and ROS-based communication. This ensures synchronized validation between virtual and physical robots, facilitating optimization based on simulated feedback. Third, it introduces an intervention-based process. Researchers can adjust virtual scenarios in real-time during physical execution, enhancing task flexibility and robustness, and extending HRC research capabilities.

Key contributions of this work are as follows:

- We present a virtual-real fusion simulation platform (DVS) for robotic research, which enables closed-loop sim-to-real transfer validation through virtual-physical synchronization and ROS-based communication. It supports a wide range of tasks.
- We provide dynamic environmental modeling, including pedestrian behavior simulation and flexible scene editing. These capabilities enhance complex task execution through diverse and high-quality data generation.
- We introduce an intervention-enabled workflow. This supports real-time scenario adjustments during physical deployment. The virtual-real synchronization mechanism improves adaptability in dynamic environments, demonstrated through manipulation tasks.

II. RELATED WORK

Simulation platforms have become integral to the development and validation of embodied artificial intelligence algorithms, enabling researchers to train and test robotic systems in controlled environments before deployment in real world tasks. These platforms have seen significant advancements over the past decade, particularly in the areas of physical modeling, scene realism, and task-specific benchmarks.

The emergence of embodied intelligence has catalyzed significant advances in robotics and AI, especially in tasks involving real world interaction and navigation. Such tasks ranging from obstacle avoidance and path planning to human-robot collaboration demand rigorous testing and training frameworks. In this context, simulation environments have emerged as indispensable tools, that offer safe, scalable, and cost-effective platforms for developing and validating embodied AI algorithms. These environments not only enable exploration of high-risk scenarios and faster algorithm iteration but also address the critical challenge of sim2real generalization, where models trained in simulation must effectively transfer to real world robotic systems.

During the past decade, the development of simulation platforms has been instrumental in advancing embodied intelligence. Platforms such as Gazebo[14], MuJoCo[38], and NVIDIA Isaac Sim[26] have excelled in robotics control and high-precision physical simulation, enabling accurate modeling of robot dynamics and multi agents systems. Meanwhile, tools like Habitat[31], AI2-THOR[15], and iGibson[17] have prioritized photorealistic environments for navigation and task planning, supporting benchmarks for tasks like object rearrangement, manipulation, and visual question answering. Recent systems such as DialFRED[8] and TEACh[28] have expanded the scope of these benchmarks by integrating natural language dialogue, encouraging richer agent-environment interactions. Despite these advancements, several persistent challenges remain unresolved, hindering the broader applicability of these platforms to dynamic and real world scenarios.

A major limitation of existing platforms is their inability to model dynamic, stochastic environments that capture realistic human behaviors and evolving scene conditions. Platforms like Habitat and AI2-THOR, while robust for static or semi-static environments, rely heavily on predefined tracks and scripted object interactions, which constrain their generalizability to real world, unpredictable conditions. Another challenge is the gap in the sim2real generalization. Although simulators like Pybullet[3] and MuJoCo excel in physical modeling, they often lack the diversity and randomness required to robustly train algorithms for real world deployment. Moreover, the growing emphasis on human-robot collaboration has exposed the limitations of existing platforms, which rarely support real-time interactions such as gesture-based commands, shared workspaces, or natural language dialogue. Systems such as HumanTHOR[41] and SEAN[39] have made notable progress toward dynamic interaction modeling, but their focus remains limited to basic social navigation or static collaborative tasks,

TABLE I: Comparison of Simulation Platforms. For the sensor, S refers to semantic, L refers to Lidar

Simulation Platform	Sensors	Dynamic Scenes		VR Interaction	ROS
		Pedestrians	Objects		
Arena[12]	RGB-D, L	✓	✗	✗	✓
AI2THOR[15]	RGB-D, S	✗	✗	✗	✗
Gibson series[17][34]	RGB-D, S, L	✗	✗	✓	✓
HoME[1]	RGB-D, S	✗	✗	✗	✗
Habitat[33][36][31]	RGB-D, S	✗	✗	✓	✗
SAPIEN[42]	RGB-D, S	✗	✗	✓	✗
ThreeDWorld[7]	RGB-D, S	✗	✗	✓	✗
VirtualHome[30]	RGB-D, S	✗	✗	✗	✗
DVS(Ours)	**RGB-D, S, L**	✓	✓	✓	✓

leaving ample room for advancement. Moreover, most existing platforms specialize either in physical modeling or in photorealistic simulation, yet rarely integrate both capabilities within a unified framework—highlighting a critical gap in tools capable of holistically supporting embodied intelligence research.

To address these limitations, we propose a novel virtual-physical integration platform that combines the strengths of high-fidelity physics, dynamic scene modeling, and real-time human-robot collaboration. By introducing stochastic pedestrian behavior modeling—including adjustable avoidance radii, randomized spawning points, and variable motion patterns—our platform supports dynamic and unpredictable environments, enhancing the robustness and generalization of robot algorithms. Additionally, a optical motion capture system provides submillimeter precision data for sim2real transfer, ensuring better deployment of simulation-trained models to real world systems. Real time human-in-the-loop (HITL) interactions[27], including gesture commands, natural language dialogue, and shared workspace collaboration, further enable realistic HRC experiments. Finally, the integration of annotated synthetic data with real world motion capture enables simultaneous development and validation across both virtual and physical domains, effectively bridging the gap between simulation and reliable real world deployment in embodied AI.

By tackling key challenges in dynamic scene modeling, sim-to-real transfer, and human-robot collaboration, our proposed platform provides a unified solution for advancing embodied intelligence. Its capacity to simulate complex, real world environments and enable seamless robot deployment positions it as a powerful enabler for future research in navigation, manipulation, and collaborative tasks. A detailed comparison of existing simulation platforms is delineated in Table I.

III. SYSTEM FRAMEWORK

In this section, we describe the key components of our Dynamic Virtual-Real Simulation Platform, which integrates virtual-real fusion and dynamic scene generation to support advanced robotic research. These two capabilities are designed to address the limitations of existing simulation platforms, enabling more effective training and evaluation of algorithms

Fig. 2: Virtual-Real Data Synchronization Framework. The central demonstrates the synchronization of object pose and robot motion through VRPN and ROS. The left and right parts depict the virtual simulation environment and physical real world scene, respectively.

in real world conditions. By combining high-fidelity virtual simulation with real-time interactions and dynamic scene modeling, DVS provides a comprehensive environment for testing mobile robots.

A. Virtual-Real Fusion for Seamless Interaction

Virtual-real fusion is a core feature of DVS, enabling precise bidirectional synchronization between the virtual and physical environments. This synchronization is critical for ensuring that algorithms trained in the virtual world can be directly applied to physical robots, thus reducing the sim-to-real gap.

The virtual-real fusion module consists of two primary components: object pose alignment and robot state synchronization. These components work together to ensure that both the objects and robots in the simulation environment align accurately with their real world counterparts.

1) Object Pose Synchronization: Object pose synchronization is a critical feature for bridging the gap between virtual and real environments, enabling accurate interactions between robots and their surroundings in both domains. In DVS, we achieve precise synchronization using 14 motion capture cameras, which provide real-time tracking with 0.1 mm positional accuracy and 0.1 ° rotational precision. This allows for high-fidelity pose alignment, essential for ensuring that physical objects, such as robot end-effectors, align accurately with their

virtual counterparts in simulation.

The synchronization process begins with extrinsic calibration of the motion capture system. By calibrating the system's extrinsic parameters, we can establish a unified world coordinate system that aligns the virtual and real spaces. This calibration is achieved through the following transformation:

$$T_{\text{virtual}} = R \cdot T_{\text{real}} + t \tag{1}$$

Where:

- R is the rotation matrix derived from the spatial calibration process, defining how the real world orientation maps to the virtual space.
- T_{real} is the translation vector representing the position of the real world object.
- t is the translation vector that compensates for any misalignment, ensuring that both spaces share a common origin.

Through this method, the physical object trajectories, such as those of a robot's end-effector, are directly mapped into the virtual environment. This enables precise interaction with virtual objects, improving the realism of simulations and ensuring the accuracy of robotic tasks that require interaction between the real and virtual worlds.

B. Dynamic Scene Generation

The dynamic scene generation module of DVS significantly enhances the realism and complexity of training environments, creating scenarios that more accurately reflect real world conditions. This module incorporates dynamic pedestrian agents and mobile robotic proxies, both of which are key to simulating the unpredictability and complexity of real world environments.

1) Dynamic Pedestrian Plug-in: DVS features a pedestrian simulation plugin that introduces human-like agents into the virtual environment. These agents feature variable motion accelerations and socially compliant avoidance behaviors, enabling dense, collision-free movement. By incorporating mechanisms such as variable speeds, random spawning points, and obstacle avoidance, the system realistically simulates human interaction dynamics. Adding dynamic pedestrians to different static scenes allows our platform to closely replicate crowded environments like supermarkets and busy restaurants, significantly enhancing the realism of indoor simulations. This provides a richer learning environment for developing mobile robots' navigation and collaboration capabilities in human-populated spaces.

Such dynamic pedestrian behaviors are crucial for training indoor mobile robots in navigation and human-robot interaction tasks. The human-like agents interact with robots in real time, enabling researchers to collect diverse and realistic datasets for optimizing navigation strategies, path planning, and collaborative algorithms. This feature is particularly valuable for enhancing robots' ability to adapt to sudden environmental changes or unpredictable pedestrian behaviors.

2) Multi-Robot Plug-in: In addition to pedestrian simulation, DVS supports the integration of multiple mobile robotic agents within the same environment. This capability allows researchers to study multi-robot collaboration and competition in dynamic settings. Simulation of multiple robots operating in close proximity enables the development of cooperative algorithms for tasks such as resource sharing, coordinated navigation, and joint manipulation.

The ability to simulate multi-robot environments in dynamic, cluttered spaces is critical for advancing robotics research. By mimicking real world challenges such as managing crowded environments or dealing with unexpected obstacles, DVS helps researchers develop more robust algorithms that can handle complex tasks in unpredictable settings.

Together, dynamic pedestrians and multi-robot integration ensure that DVS provides a training environment that closely mirrors real world operational conditions. These capabilities are essential for developing robots that can navigate complex spaces, collaborate with humans, and adapt to dynamic changes in their environments.

IV. APPLICATIONS OF DVS PLATFORM

Our platform supports the full workflow, from data generation to real world validation. In the previous chapter, we introduced two core modules of our system. This chapter discusses the construction of a large-scale virtual-real fusion dataset and explores the experimental data generation process, along with its application in task training and testing.

A. Data Perception and Generation

Fig. 3: The interactive interface of the simulation platform: The left panel adjusts dynamic pedestrian parameters while the right selects perception data types.

In robotics research, virtual environments provide clear task representations, enabling agents to perform tasks in controlled settings. Data generation is a core feature of simulation platforms. As shown in Fig. 3, our platform facilitates the generation and processing of various data formats, including RGB images, depth maps, 2D/3D bounding boxes, semantic and instance segmentation, and trajectory data, all via a user-friendly interface [37]. These data types support foundational tasks and enable complex research scenarios, such as path planning [11], relocation[2], and grasping [47].

To enhance data quality and usability in simulation, we optimize the data generation pipeline by ensuring smooth camera trajectories and precise depth-to-RGB alignment. Specifically, we employ Bézier curves to generate smooth camera motion, minimizing abrupt directional changes—particularly at trajectory corners—which significantly improves frame-to-frame

Fig. 4: The robotic arm is interrupted while executing Prompt A and is requested to execute Prompt B. The first row shows the robotic arm in the virtual platform, and the second row shows the real robotic arm.

feature matching and point cloud reconstruction. Depth data is temporally aligned with RGB frames to guarantee precise synchronization, which is critical for multi-sensor fusion and accurate scene modeling.

We evaluated the impact of these optimizations by collecting camera trajectories both with and without smoothing, and analyzed feature matching performance using LightGlue[19] and SuperPoint[4]. The results in Table II show that smoothed trajectories yield a higher number of feature correspondences between frames, particularly in small and cluttered indoor environments such as bedrooms. As demonstrated in our experiments, these improvements in camera handling directly enhance downstream tasks like 3D reconstruction and semantic segmentation, supporting more robust and scalable robotics research as task complexity increases.

B. Robotic Tasks Learning

1) Virtual-Real Intervention Grasping: A key weakness of learned policies in robotic manipulation [9, 46, 5, 47]

TABLE II: Camera Trajectories With and Without Smoothing

Scene	Trajectory Type	Avg. Features ↑
Bedroom	Straight	751.35
	Smooth	1484.29
Livingroom	Straight	1190.47
	Smooth	1631.73

TABLE III: Task Success Rates by Module and Prompt Order

Module	Prompt Order		Success Rate (%)	
	First	Second	First Task	Second Task
OpenVLA-7B	A	B	0.0	100.0
	B	A	0.0	90.0
RDT-1B	A	B	0.0	80.0
	B	A	0.0	90.0

is that their success rate in task execution is low when deployed in practice, even with domain adaptation [48]. In heterogeneous deployments using only pretrained weights, the success rate of robots performing tasks across different models tends to approach zero. Even when data collection for specific tasks is done using the robot being deployed and fine-tuned, the success rate of task execution is still only around 90%, making it difficult to apply in the industry. However, due to the characteristics of our platform, which includes virtual-real mapping and benchmark alignment between the virtual environment and the real world, and the fact that the robot has a ROS communication interface, we can supervise and intervene in the robot's tasks in the real world through the platform to improve the success rate of task execution. We set up experimental conditions based on the common manipulation task of grasping. As shown in Fig. 4, in order to reflect the characteristics of our platform supervision and intervention, we provide the gripper with wrong instructions at the beginning of the experiment, and interrupt the task and provide new tasks based on virtual scenes through the platform when the gripper is performing the task. We utilized a seven-degree-of-freedom Kinova Gen3 robotic arm to collect nearly a hundred grasping data points on a planar surface. The data was then fine-tuned on the pre-trained models released by OpenVLA-7B[13] and RDT-1B[22], enabling our robotic arm to achieve a high success rate in performing tasks in specific scenarios. At the beginning of the experiment, we provided the robotic arm with prompts to grasp an apple and a banana, and midway through task execution, we interrupted the task on the platform and assigned a new task.The experiment demonstrated that our platform effectively intervened in the robotic arm's task execution. The experimental results are shown in the Table III. Prompts: A: "Pick up the apple"; B: "Pick up the banana."

2) Real to Sim to Real Learning: In autonomous robotic task execution without human intervention, integrating physical robots and their sensory systems into a virtual-real fusion architecture serves as a key technical strategy for improving task success rates. The real-world component of our platform includes the physical robot and sensing devices used to collect feedback. A notable application of virtual-real synchronization is digital twin monitoring, which enables continuous evaluation and refinement of algorithms using real-world data—effectively bridging the sim-to-real gap.

To achieve precise synchronization, we employ ROS2 to align the real robot's motion, controlled via MoveIt, with that of its virtual counterpart. This ensures the virtual environment reflects physical priors such as joint friction and mechanical latency. We validated this approach in a Virtual-Real Assisted (VLA) grasping task by comparing standard sim-to-real transfer with a fine-tuned version incorporating real-world feedback. As shown in Table IV, the virtual-real fusion method led to significantly better performance, highlighting the value of real-time feedback in simulation refinement.

To support accurate virtual-real synchronization, we use a motion capture system for high-precision calibration be-

TABLE IV: Comparative with Different Finetuning Data

Task	Trials	Finetune Data	Successes
Pick the apple and place	10	virtual	6
at the target point	10	virtual-real	9
Pick the banana and place	10	virtual	4
at the target point	10	virtual-real	8

tween real and virtual coordinate systems. This calibration is essential for aligning the physical robot's position with its virtual counterpart, correcting errors such as odometry drift and IMU noise. For tasks that are less sensitive to positional accuracy, we leverage the built-in mapping capabilities of the Quest system to achieve approximate alignment—a low-cost yet sufficiently accurate alternative for many practical applications.

3) Dynamic Indoor Pedestrian Trajectory Prediction:
Pedestrian trajectory prediction aims to forecast future trajectories based on observed trajectories, while considering complex interactions and environmental layouts [18]. It serves as a crucial connection between the perception system and the planning system.

Three trajectory prediction algorithms, i.e. STGAT [10], Trajectron++ [32] and TUTR [35], are tested on our synthetic indoor scenes (Gym, Office and Supermarket) as well as the official public outdoor dataset (ETH [29]). We use ADE (Average Displacement Error) and FDE (Final Displacement Error) as evaluation metrics, where lower ADE and FDE values indicate better performance. The experimental results are depicted in Table V. Additionally, to analyze pedestrian movement patterns and collision avoidance strategies, we selected two dense indoor scenes (Gym and Office) and visualized the predicted trajectories in Fig. 5.

Overall, all three methods experience a significant performance decrease when applied to indoor scenes compared to the outdoor ETH scene. Specifically, the ADE for STGAT decreases from 0.79 to 1.42 (79.7%) when generalizing from the ETH scene to the Supermarket scene, while the FDE for STGAT decreases from 1.48 to 2.88 (94.5%) in the

Fig. 5: Visualization of pedestrian trajectory prediction, where each color represents a different pedestrian. The accuracy of the prediction is higher when the predicted trajectory (short dashed line) closely aligns with the ground truth (GT, solid line). In environments with dense static obstacles, such as indoors, the predicted future trajectory may result in collisions (red rectangular box).

TABLE V: Experiments on Pedestrian Trajectory Prediction. Gym, Office and Supermarket are our synthetic indoor scenes, while ETH [29] is the official public outdoor dataset.

Scene	Method	ADE ↓	FDE ↓
Gym	STGAT	1.39	3.01
	Trajectron++	0.59	1.02
	TUTR	0.70	1.19
Office	STGAT	1.38	2.75
	Trajectron++	0.89	1.60
	TUTR	0.81	1.40
Supermarket	STGAT	1.42	2.88
	Trajectron++	0.96	1.82
	TUTR	0.83	1.50
ETH	STGAT	0.79	1.48
	Trajectron++	0.52	0.97
	TUTR	0.43	0.83

same scenario. We analyze this performance drop from three perspectives. First, compared to outdoor scenes, narrow indoor spaces are often filled with numerous static obstacles, which can interfere with human trajectory decision-making and lead to collisions. Second, indoor human interactions are more frequent due to communication or obstacles caused by people standing in the way, making predictions more challenging. Third, indoor spaces are generally smaller than outdoor environments, with pedestrian trajectories being less spread out, making predictions more sensitive to small positional changes. If the model was trained in larger, more open outdoor spaces, it may not have learned to adapt to the smaller, more dynamic movements of indoor environments.

The results also underscore the importance of robust spatial-temporal modeling in trajectory prediction tasks. The transformer-based architecture of TUTR appears particularly well suited to capture intricate interactions over time, which leads to its superior performance. Trajectoron provides a balance of stability and accuracy, but lags behind in highly dynamic environments. In contrast, STGAT's graph-based approach, while effective in simpler scenarios, struggles in complex environments, highlighting its limitations in handling high-dimensional spatial-temporal variability. These findings offer valuable insights for future research, emphasizing the need for models that can generalize effectively across diverse scenarios while maintaining low computational overhead.

4) Dynamic Indoor Social Navigation: Social navigation refers to the process by which agents use social cues and prior experiences to determine paths, make decisions, and navigate complex environments. It incorporates interpersonal and collective information, including behavioral patterns, real-time human feedback, and established social norms.

We evaluated three indoor social navigation algorithms ORCA [40], DS-RNN [20], and AttnGraph [21] within synthetic indoor restaurant and store environments in our platform with varying levels of dynamic complexity. Performance was quantified using three primary metrics: Success Rate, average Navigation Time (in seconds) for successful episodes, and Collision Rate with other humans. The experimental results

TABLE VI: Experiments on Social Navigation

Scene	Metric	HumanNumber=10 / 15 / 20		
		ORCA	DS-RNN	AttnGraph
Restaurant	SuccessRate ↑	0.78 / 0.74 / 0.62	0.82 / 0.76 / 0.68	0.83 / 0.77 / 0.67
	CollisionRate ↓	0.01 / 0.06 / 0.08	0.01 / 0.05 / 0.06	0.02 / 0.06 / 0.07
	NavigationTime ↓	42.50 / 43.59 / 43.90	37.48 / 44.96 / 45.24	39.68 / 44.81 / 49.65
Store	SuccessRate ↑	0.96 / 0.85 / 0.51	0.98 / 0.81 / 0.75	0.98 / 0.87 / 0.79
	CollisionRate ↓	0.03 / 0.04 / 0.06	0.01 / 0.07 / 0.09	0.02 / 0.04 / 0.06
	NavigationTime ↓	40.39 / 47.62 / 46.47	34.21 / 39.51 / 39.97	41.56 / 43.75 / 48.22

are shown in Table VI.

The results indicate that with increasing dynamic scene complexity, the performance of all three algorithms deteriorates to varying extents, consistent with our expectations. Specifically, as the number of dynamic pedestrians increases, the success rate decreases, whereas the collision rate and navigation time increase, each to different extents. This demonstrates the influence of dynamic scene complexity on algorithm performance. This further highlights that our online simulation platform is well suited for closed-loop training and testing.

V. CONCLUSION

We propose a dynamic virtual-real simulation platform that integrates configurable pedestrian behavior simulation, large-scale indoor environments, optical motion capture, and ROS-based bidirectional virtual-reality communication. The platform introduces two major innovative modules for virtual reality integration, overcoming current limitations in robotic simulation systems for dynamic scenarios and real world deployment. Experimental results show that DVS supports navigation and human-robot interaction research, achieving closed-loop performance in real world missions. Future work will focus on integrating haptic feedback, developing AI-driven intervention strategies, and improving compatibility with industrial robotic arms. This platform creates a new paradigm for closed-loop virtual-reality interaction, advancing human-robot collaboration and dynamic environment adaptation.

ACKNOWLEDGMENTS

REFERENCES

[1] Simon Brodeur, Ethan Perez, Ankesh Anand, Florian Golemo, Luca Celotti, Florian Strub, Jean Rouat, Hugo Larochelle, and Aaron Courville. Home: A household multimodal environment. *arXiv preprint arXiv:1711.11017*, 2017.

[2] Fang-xing Chen, Yifan Tang, Cong Tai, Xue-ping Liu, Xiang Wu, Tao Zhang, and Long Zeng. Fusednet: End-to-end mobile robot relocalization in dynamic large-scale scene. *IEEE Robotics and Automation Letters*, 9(5): 4099–4105, 2024.

[3] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[5] Zhikai Dong, Sicheng Liu, Tao Zhou, Hui Cheng, Long Zeng, Xingyao Yu, and Houde Liu. Ppr-net:point-wise pose regression network for instance segmentation and 6d pose estimation in bin-picking scenarios. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 1773–1780. IEEE Press, 2019.

[6] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[7] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

[8] Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S Sukhatme. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics and Automation Letters*, 7(4): 10049–10056, 2022.

[9] Ding-Tao Huang, En-Te Lin, Lipeng Chen, Li-Fu Liu, and Long Zeng. Sd-net: Symmetric-aware keypoint prediction and domain adaptation for 6d pose estimation in bin-picking scenarios. In *2024 IEEE/RSJ International*

Conference on Intelligent Robots and Systems (IROS), pages 2747–2754, 2024.

[10] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6272–6281, 2019.

[11] Xing Hui Jing, Xin Xiong, Fu Hao Li, Tao Zhang, and Long Zeng. A two-stage reinforcement learning approach for robot navigation in long-range indoor dense crowd environments. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5489–5496, 2024.

[12] Linh Kästner, Volodymyir Shcherbyna, Huajian Zeng, Tuan Anh Le, Maximilian Ho-Kyoung Schreff, Halid Osmaev, Nam Truong Tran, Diego Diaz, Jan Golebiowski, Harold Soh, et al. Arena 3.0: Advancing social navigation in collaborative and highly dynamic environments. *arXiv preprint arXiv:2406.00837*, 2024.

[13] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[14] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2149–2154. Ieee, 2004.

[15] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli Vander-Bilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[16] Bo Li, Pingfa Feng, Long Zeng, Chao Xu, and Jianfu Zhang. Path planning method for on-machine inspection of aerospace structures based on adjacent feature graph. *Robotics and Computer-Integrated Manufacturing*, 54: 17–34, 2018.

[17] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021.

[18] Wenzhan Li, Fuhao Li, Xinghui Jing, Pingfa Feng, and Long Zeng. Dual-alignment domain adaptation for pedestrian trajectory prediction. *IEEE Robotics and Automation Letters*, 9(12):10962–10969, 2024.

[19] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023.

[20] Shuijing Liu, Peixin Chang, Weihang Liang, Neeloy Chakraborty, and Katherine Driggs-Campbell. Decen-tralized structural-rnn for robot crowd navigation with deep reinforcement learning. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 3517–3524. IEEE, 2021.

[21] Shuijing Liu, Peixin Chang, Zhe Huang, Neeloy Chakraborty, Kaiwen Hong, Weihang Liang, D. Livingston McPherson, Junyi Geng, and Katherine Driggs-Campbell. Intention aware robot crowd navigation with attention-based interaction graph. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 12015–12021, 2023.

[22] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[23] Zhihao Liu, Quan Liu, Wenjun Xu, Lihui Wang, and Zude Zhou. Robot learning towards smart robotic manufacturing: A review. *Robotics and Computer-Integrated Manufacturing*, 77:102360, 2022.

[24] Matteo Luperto, Javier Monroy, Jennifer Renoux, Francesca Lunardini, Nicola Basilico, Maria Bulgheroni, Angelo Cangelosi, Matteo Cesari, Manuel Cid, Aladar Ianes, et al. Integrating social assistive robots, iot, virtual communities and smart objects to assist at-home independently living elders: the movecare project. *International Journal of Social Robotics*, 15(3):517–545, 2023.

[25] Eloise Matheson, Riccardo Minto, Emanuele GG Zampieri, Maurizio Faccio, and Giulio Rosati. Human–robot collaboration in manufacturing applications: A review. *Robotics*, 8(4):100, 2019.

[26] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023.

[27] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review*, 56(4): 3005–3054, 2023.

[28] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. TEACh: Task-driven Embodied Agents that Chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025, 2022.

[29] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009.

[30] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome:

Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502, 2018.

[31] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

[32] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

[33] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

[34] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, 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)*, pages 7520–7527. IEEE, 2021.

[35] Liushuai Shi, Le Wang, Sanping Zhou, and Gang Hua. Trajectory unified transformer for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9675–9684, 2023.

[36] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021.

[37] Yi-Fan Tang, Cong Tai, Fang-Xing Chen, Wan-Ting Zhang, Tao Zhang, Xue-Ping Liu, Yong-Jin Liu, and Long Zeng. Mobile robot oriented large-scale indoor dataset for dynamic scene understanding. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 613–620, 2024.

[38] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[39] Nathan Tsoi, Alec Xiang, Peter Yu, Samuel S. Sohn, Greg Schwartz, Subashri Ramesh, Mohamed Hussein, Anjali W. Gupta, Mubbasir Kapadia, and Marynel Vázquez. Sean 2.0: Formalizing and generating social situations for robot navigation. *IEEE Robotics and Automation Letters*, 7(4):11047–11054, 2022.

[40] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics Research: The 14th International Symposium ISRR*, pages 3–19. Springer, 2011.

[41] Chenxu Wang, Boyuan Du, Jiaxin Xu, Peiyan Li, Di Guo, and Huaping Liu. Demonstrating humanthor: A simulation platform and benchmark for human-robot collaboration in a shared workspace. *arXiv preprint arXiv:2406.06498*, 2024.

[42] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020.

[43] Leicai Xiao, Long Zeng, Zhaobo Xu, and Xueping Liu. Assembly language design and development for reconfigurable flexible assembly line. *Robotics and Computer-Integrated Manufacturing*, 80:102467, 2023.

[44] Zhaobo Xu, Chaoran Zhang, Song Hu, Zhaochun Han, Pingfa Feng, and Long Zeng. Reconfigurable flexible assembly model and implementation for cross-category products. *Journal of Manufacturing Systems*, 77:154–169, 2024.

[45] Long Zeng, Wei Jie Lv, Xin Yu Zhang, and Yong Jin Liu. Parametricnet: 6dof pose estimation network for parametric shapes in stacked scenarios. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 772–778, 2021.

[46] Long Zeng, Weijie Lv, Zhi-Kai Dong, and Yong Liu. Ppr-net++: Accurate 6-d pose estimation in stacked scenarios. *IEEE Transactions on Automation Science and Engineering*, 19:3139–3151, 2022.

[47] Hao Zhang, Hongzhuo Liang, Lin Cong, Jianzhi Lyu, Long Zeng, Pingfa Feng, and Jianwei Zhang. Reinforcement learning based pushing and grasping objects from ungraspable poses. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3860–3866, 2023.

[48] Liang Zhao, Meng Sun, Wei Jie Lv, Xin Yu Zhang, and Long Zeng. Domain adaptation on point clouds for 6d pose estimation in bin-picking scenarios. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2925–2931, 2023.