SidewalkBench: Benchmarking Visual Navigation on Urban Sidewalks

Zhizheng Liu[1,*] **Honglin He**[1,*] **Vivek Alumootil**[1,*]
Akshat Pandya[2] **Brad Squicciarini**[2] **Wayne Wu**[1] **Bolei Zhou**[1]
[1]University of California, Los Angeles [2]Coco Robotics
https://vail-ucla.github.io/SidewalkBench

Abstract: Urban sidewalk navigation presents significant challenges due to complex structural layouts, dynamic pedestrian behaviors, and long distances. While recent visual navigation models offer a promising solution, the lack of a unified benchmark hinders quantitative and reproducible evaluation. To bridge this gap, we propose SidewalkBench, a comprehensive benchmark designed for visual navigation on urban sidewalks. Built upon NVIDIA Isaac Sim, SidewalkBench brings GPU-accelerated simulation of diverse, high-fidelity sidewalk environments, including both procedurally generated and real-world scanned scenes. We further populate the scenes with rich, reactive event-based pedestrian behaviors and flexible, efficient animation, enabling standardized model evaluation under realistic real-world settings. We conduct a comprehensive evaluation of 9 visual navigation models on 330 unit-test scenarios, 800 pedestrian-reactive scenarios, and 105 long-horizon scenarios. Our findings highlight that pedestrian interaction and long-horizon robustness remain critical bottlenecks for existing models, and scaling up sidewalk training with synthetic data emerges as a promising solution.

Keywords: Visual Navigation Benchmark, Urban Sidewalk Simulation

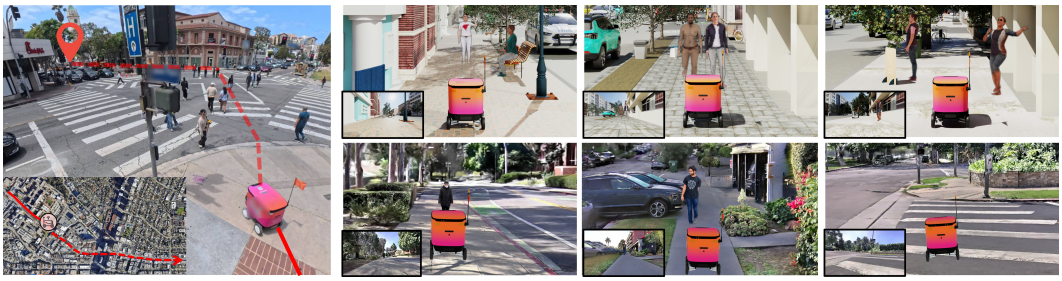

Figure 1: Left: Challenges of sidewalk navigation include static obstacles, dynamic pedestrians, diverse structures and layouts, and long-horizon distances. Right: SidewalkBench can simulate diverse and realistic scenarios for standardized evaluation of visual navigation on urban sidewalks.

1 Introduction

The prevalence of mobile robots and devices on urban sidewalks is rapidly increasing [1]. For instance, sidewalk delivery robots streamline last-mile logistics, electric wheelchairs provide accessible transportation to the elderly and people with disabilities, and e-scooters offer eco-friendly commuting alternatives. Transitioning to the autonomous operation of these machines can further reduce labor costs and improve personal convenience. Yet, the ability to safely navigate complex city streets still remains a significant challenge: As shown in Fig. 1 (left), the robot must navigate long distances through several street blocks with varied structures and layouts, cross intersections while obeying traffic rules and signs, and avoid collisions with obstacles such as lamp posts and stop signs. More importantly, sidewalks are populated with pedestrians who exhibit complex behaviors and movements [2], introducing a high degree of uncertainty that requires the robot to predict pedestrians' intent and interact safely and socially.

[*]Equal contribution.

Recent visual navigation foundation models [3, 4, 5, 6] have provided a promising solution for urban sidewalk navigation, requiring only a monocular RGB camera for perception and demonstrating strong generalization across environments and embodiments. Although a few visual navigation methods have been evaluated on urban sidewalks separately [5, 7, 8, 9], the scale of their experiments is rather limited, and it is difficult to assess the model's capability for real-world deployment. Specifically, their testing scenarios are usually short-horizon trials and lack structural and layout diversity. For instance, NavBench-GS [5] only includes straight paths mostly less than 10 meters. Furthermore, there are no standardized protocols and explicit scenario definitions essential for reproducible, comprehensive analysis. Crucially, no current study has isolated how different pedestrian behaviors would affect the model performance. For example, Citywalker [7] simply classifies the scenarios into forward, left turn, and right turn. While existing social navigation benchmarks [10, 11, 12] do analyze human-robot interaction across various social scenarios, they primarily focus on indoor settings and are not dedicated to urban sidewalks. Consequently, the lack of a unified sidewalk navigation benchmark has become a critical gap for fairly comparing models and isolating the failure modes and bottlenecks that guide the future development of autonomous robots in public spaces.

In this work, we aim to establish a comprehensive benchmark for visual navigation on urban sidewalks: SidewalkBench. We build our benchmark on NVIDIA Isaac Sim [13], taking full advantage of its accurate physics simulation, efficient parallel rendering, and extensive simulation ecosystem such as UrbanVerse [14] and Urban-Sim [15]. To simulate diverse and realistic urban scenes, SidewalkBench introduces two distinct types of environments: (1) procedurally generated urban scenes that compose various sidewalk blocks, layouts, and obstacles into diverse large-scale environments, and (2) real-world scanned scenes reconstructed from 3D Gaussian Splatting (3DGS) [16] via a high-fidelity spatial camera to ensure high visual and geometric realism. To generate rich pedestrian behaviors and movements, we further develop a pedestrian simulation module that includes a set of dynamic, event-based behaviors triggered to generate diverse pedestrian-interactive scenarios, as well as a custom pedestrian animation pipeline with high flexibility and efficiency. Ultimately, as shown in Fig. 1 (right), SidewalkBench enables the simulation of diverse scenes and pedestrian activities, reflecting the challenges of real-world sidewalk navigation.

To comprehensively evaluate different aspects of model performance, SidewalkBench introduces three distinct types of testing scenarios: 330 unit-test scenarios, 800 pedestrian-reactive scenarios, and 105 long-horizon scenarios. We benchmark 9 representative visual navigation models across these scenarios and conduct an in-depth analysis. Our main findings reveal that: (1) scaling urban sidewalk training data is the most important factor for the visual navigation performance; (2) nearly all models struggle with complex pedestrian behaviors and gesture understanding; (3) long-horizon navigation remains far from solved, with the top-performing model still experiencing 1.34 failures per 100m of travel distance; and (4) our simulation platform can serve as a scalable synthetic data generator for model finetuning. These insights highlight a critical gap for future research, particularly in enhancing pedestrian behavior understanding and long-horizon robustness.

We summarize our main contributions as follows:

- A comprehensive benchmark SidewalkBench for visual navigation on urban sidewalks, facilitating standardized model testing and comparison.
- An urban sidewalk simulation platform with diverse and realistic scenes as well as rich and efficient pedestrian simulation, reflecting the challenges in real-world deployment.
- A detailed evaluation and analysis of representative visual navigation models on unit-test scenarios, pedestrian-reactive scenarios, and long-horizon scenarios, revealing the lack of pedestrian behavior understanding and long-horizon robustness in existing models.

2 Related Work

Visual Navigation Foundation Models Visual navigation foundation models [3, 5, 6, 7, 17, 18, 19, 20, 21] use inputs from a monocular RGB camera and predict control signals or waypoints for robot navigation. These models are usually trained end-to-end using a data-driven approach, with many recent advances in model architecture and training recipes. NoMaD [18] and NavDP [22] introduce

Figure 2: **Overview of scene types in SidewalkBench.**

diffusion policies for navigation and exploration. CityWalker [7] and NWM [23] use web-scale video data for broad generalization. S2E [5] introduces reinforcement learning with a simulator for post-training, and recent Vision-Language-Action (VLA) models [6, 8, 9] have emerged to map continuous visual streams and linguistic instructions directly to real-time continuous control signals. Despite promising progress, there is no uniform benchmark for evaluating visual navigation in urban sidewalk scenarios, which is an important downstream application for these models.

Simulation Benchmarks for Visual Navigation Simulation benchmarks for visual navigation enable standardized, reproducible comparisons of different models. Existing benchmarks [24, 25] mainly focus on static indoor scenes, and follow-up works [26, 27] incorporate dynamic human behaviors to evaluate human-aware navigation performance. To bridge the Real2Sim evaluation gap with higher visual realism, Vid2Sim [28], NavBench-GS [5], and Wanderland [29] use reconstructed outdoor scenarios from 3DGS [16] to provide real-world grounded simulation but lack dynamic pedestrians. SocNavBench [30] includes scenarios with urban pedestrians, but it suffers from low rendering quality and limited scene diversity. Furthermore, these benchmarks are typically limited to short-horizon evaluations with trajectories spanning around 10 meters [5, 30]. Our SidewalkBench is the first of its kind that focuses specially on large-scale urban dynamic environments with both procedurally generated and real-world scanned scenes.

Urban Simulators for Embodied AI With the growing interest in developing autonomous agents in urban public spaces, many urban simulators have been developed with a focus on various tasks and embodiments including autonomous driving [31], UAVs [32], sidewalk autonomy [15, 33], and LLM/VLM agents [34, 35]. Some social navigation simulators [11, 36, 37] are also relevant which focus on simulating various pedestrian behaviors. For the evaluation of visual navigation on urban sidewalks, the simulation engine is required to simulate accurate physics and support efficient and realistic rendering of the RGB observations, while many of the simulators above don't support these functionalities using game engines like Unity [38] or Unreal [39]. Built upon Isaac Sim [13] and Isaac Lab [40], our SidewalkBench leverages the high-fidelity, GPU-accelerated physics engines tailored for robot learning to simulate large-scale urban scenes and provide realistic RGB observations in parallel environments. To address the pedestrian rendering bottleneck, we further develop a more flexible and efficient pedestrian animation pipeline compared to prior works [15, 37].

3 SidewalkBench Design

SidewalkBench uses NVIDIA Issac Sim [13] as the simulation engine, which provides GPU acceleration to enable accurate physics simulation and realistic camera rendering at scale. To generate diverse and realistic urban sidewalk testing scenarios, we first introduce our sidewalk scenes in Sec. 3.1. Within these scenes, we further simulate rich pedestrian behaviors with diverse human-robot interactions, which is detailed in Sec. 3.2. Based on the simulation platform, we define a diverse set of scenarios in Sec. 3.3 that the models might often encounter in real-world deployment.

3.1 Scene Types

SidewalkBench includes two complementary scene types: procedurally generated scenes and real-world scanned scenes. The procedurally generated scenes focus on the diversity and controllability of sidewalk structures and layouts, enabling a comprehensive evaluation of the model's generalization performance. The real-world scanned scenes prioritize the realism of scene appearance and geometry, allowing comparisons under photorealistic conditions and facilitating the study of the real-to-sim gap. An overview of the scenes is illustrated in Fig. 2.

Procedurally Generated Scenes We use procedural generation to create large-scale scenes with diverse sidewalk blocks, layouts, and obstacles. Following MetaUrban [33] and Urban-Sim [15], we first define 7 primitive block types, such as straight segments, curves, and intersections with varying lengths, then connect them via spline-based routing to form continuous urban topologies. Next, we split each block into 5 functional zones, including roads, sidewalks, curbs and gutters, road verges, and frontage zones, according to the lateral functional structure of the block. We further randomize the layouts of these zones and add block-specific elements, such as ramps and crosswalks at intersections. Finally, we leverage UrbanVerse-100K [14], a large-scale urban asset database, to sample diverse sky HDRIs, ground textures, physical materials, and zone-specific static objects with randomized placements. Using this pipeline, we generate 100 large-scale environments, each covering an area of 2 km$\times$2 km and comprising a wide variety of sidewalk configurations.

Real-world Scanned Scenes Recent advancements in 3DGS [16] have enabled robot training and evaluation [28, 41] with photo-realistic rendering. To further reconstruct accurate geometry, we use a spatial camera from XGRIDS [42], equipped with a LiDAR and four cameras, to scan and reconstruct many street blocks. The reconstructed scenes feature realistic visual appearance from 3DGS and accurate sidewalk geometry and physics from the scanned mesh. After that, we annotate the sidewalk and crosswalk regions and convert the reconstruction to a simulation-ready format supported by NVIDIA Isaac Sim [13]. In total, we collected 11 real-world scanned scenes with various block types and the average scale is 150 m$\times$150 m.

3.2 Pedestrian Simulation

Pedestrian simulation is crucial for testing a model's capability to safely react to the diverse pedestrian behaviors and motions encountered in real-world urban spaces. SidewalkBench adopts a two-level approach for pedestrian simulation: For high-level behaviors, we introduce event-based behaviors to generate standardized human-interactive scenarios with diverse behaviors. For low-level animation, we propose a new pipeline with superior movement flexibility and rendering efficiency. More details on the pedestrian simulation can be found in Appendix A.

Event-based High-level Behaviors We use behavior graphs [11] to model the high-level pedestrian behavior, where each node represents either a navigation waypoint or a static object that a pedestrian can interact with. A state machine determines the behavior transitions between walking between nodes, staying idle, or interacting with the object or other pedestrians. To further enrich pedestrian behaviors and create human-interactive scenarios for the robot, we propose event-based pedestrian behaviors that are dynamically triggered by the pedestrian's relative position to the robot. This allows us to test the robot's reaction under different pedestrian behaviors and generate standardized scenarios for reproducible benchmarking. Specifically, we classify common interaction behaviors on urban sidewalks, including both existing social scenarios defined in [10, 43] and sidewalk-specific behaviors such as pedestrian crossing and queueing, each with its own triggering condition. We then incorporate these event-based behaviors into the behavior state machine to simulate a highly diverse range of behaviors. An illustration of the behaviors is shown in Fig. 3.

Flexible and Efficient Low-level Animation Pedestrian movements in existing urban simulators [15, 33, 35] primarily rely on fixed animation assets with limited movement types such as walking and standing, which lack the diversity and flexibility to generate arbitrary movements, including the stopping gesture required by the event-based behavior. In addition, the pedestrian rendering speed has become a critical bottleneck for simulation efficiency with the complex graphics pipeline. SidewalkBench designs a new pedestrian animation pipeline in which we represent all pedestrians

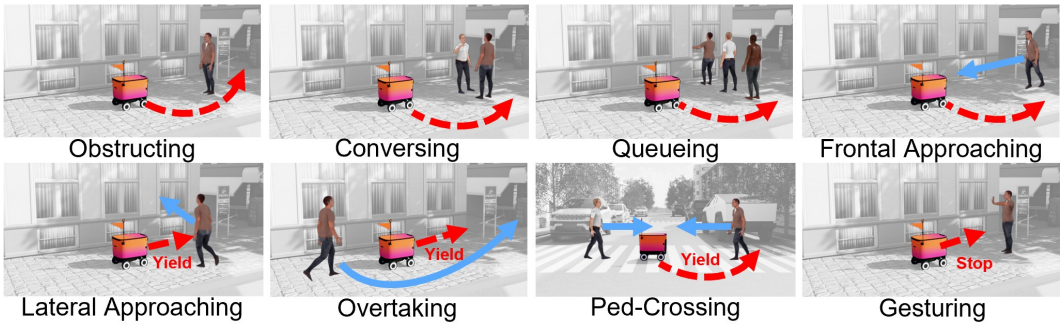

Figure 3: **Illustration of the event-based behaviors**. The blue arrow indicates the moving direction of the pedestrian after the event is triggered and the red arrow indicates the robot's intended reaction.

using the SMPL [44] human body model and directly use the SMPL parameters to control pedestrian animation and render either textured SMPL meshes [45] or SMPL-based human 3DGS avatars [46]. This greatly enhances the flexibility and diversity of pedestrian movements, with full control over detailed motion, by leveraging human motion generation models [47] and large-scale human motion datasets [48, 49]. We further develop a custom pedestrian renderer based on the highly efficient Nvdiffrast library [50] and compose the pedestrian rendering with the background rendered by the simulator, achieving a 60x improvement in rendering efficiency compared to the human animation module [51] in Isaac Sim, thereby enabling large-scale evaluation in parallel environments.

3.3 Scenarios

Current visual navigation models are typically tested on a few short-horizon sidewalk scenarios with limited scene diversity [5, 7], which can hardly reflect real-world deployment performance. Moreover, it is unclear how these models would react to various pedestrian behaviors without a standardized, reproducible benchmark. Based on the simulation platform discussed above, SidewalkBench proposes three complementary scenario types for comprehensive model evaluation: unit-test scenarios, pedestrian-reactive scenarios, and long-horizon scenarios. It also allows users to define custom scenarios to evaluate specific sidewalk navigation tasks. More details are presented in Appendix A.

Unit-test scenarios assess the model's performance in navigating basic topological structures of sidewalks. These scenarios are short-horizon with the distances between the start and the goal ranging from 10 to 20 meters and are sampled from three types of sidewalk blocks: straight blocks, curve blocks, and crosswalks. We focus on evaluating the models' lane-following and static collision-avoidance capabilities in different structures and layouts, and therefore do not simulate pedestrians in these scenarios. The task fails if the robot collides with any obstacles or is outside the sidewalk region, and succeeds if the robot reaches the goal within the time limit (60s). In total, we collect 330 unit-test scenarios with 100 scenarios for each block type in the procedurally generated scenes and 10 scenarios for each block type in the real-world scanned scenes.

Pedestrian-reactive scenarios evaluate the model's ability to safely react to the various types of pedestrian behaviors. The setting is similar to the unit-test navigation, where we additionally simulate one or more pedestrians with different event-based behaviors and place them on the path to the goal to ensure the event can be triggered. We only evaluate on the procedurally generated scenes, as they are more structured and controllable for standardized testing. Except for the pedestrian crossing behavior, which is evaluated at the crosswalk, all other behaviors are tested on the straight sidewalk block to isolate the effects of structural variations. In total, we collect 800 pedestrian-reactive scenarios with 100 scenarios for each type of pedestrian behavior.

Long-horizon scenarios require the robot to traverse large-scale environments to mimic real-world deployment. We sample the start and goal at a long distance ($>$100m) and ensure there is only a single path from the start to the goal by blocking the other routes with obstacles. As this is a challenging task for current visual navigation models, we do not terminate the episode upon failure; instead, we reset the robot to the closest waypoint and count the failure modes. In total, we collect

Method	Data (# Hours)	Goal	Encoder	Decoder	Resolution	FPS	Model Size
ViNT [3]	General	Image	CNN	Regression	85×64	4	24 M
NoMaD [18]	General	Image / None	CNN	Diffusion	96×96	4	16 M
MBRA [4]	General	Point	CNN	Regression	96×96	5	63 M
InternVLA-N1 [6]	General	Language	VLM	Diffusion	384×384	5	7 B
MIMIC [5]	Sidewalk (50)	Point	ViT	Reg. + Cls.	256×256	5	79 M
S2E [5]	Sidewalk (100)	Point	ViT	Reg. + Cls.	256×256	5	95 M
CityWalker [7]	Sidewalk (300)	Point	ViT	Regression	630×350	5	201 M
FlowPilot [54]	Sidewalk (300)	Point / None	ViT	FM. + Cls.	512×288	20	230 M
OpenPilot [55]	Sidewalk (1000)	None	ViT	Regression	352×128	20	8 M

Table 1: **Detailed comparison of visual navigation models evaluated on SidewalkBench.**

105 long-horizon scenarios with 100 scenarios from procedurally generated scenes and 5 scenarios from real-world scanned scenes—covering a total route distance of 36.5 km.

Evaluation Metrics For unit-test scenarios, we follow existing navigation benchmarks to evaluate route completion rate [52, 25, 29]. For pedestrian-reactive scenarios, we evaluate the success rate for overall task completion. For long-horizon scenarios, similar to how human intervention frequency is used to evaluate real-world autonomous driving [53], we evaluate the average failure counts per 100 meters traveled, including collision, out-of-lane, and freezing. We also evaluate the average velocity (m/s) for efficiency evaluation. Full results, including additional metrics such as social compliance, are presented in the Appendix.

4 Experiments

We discuss the list of visual navigation models evaluated on SidewalkBench in Sec. 4.1. The results of SidewalkBench including unit-test scenarios, pedestrian-reactive scenarios and long-horizon scenarios are presented in Sec. 4.2, Sec. 4.3, Sec. 4.4, respectively. Some qualitative results are shown in Fig. 5. To show the usefulness of our platform in generating synthetic data for model training, we conduct a preliminary finetuning experiment in Sec. 4.5 for the pedestrian-reactive scenarios.

4.1 Navigation Models

We evaluate 9 recent visual navigation models on our benchmark. We choose ViNT [3], MBRA [4], NoMaD [18], and InternVLA-N1 [6] as representative models for general navigation which are trained on diverse data sources. We also include CityWalker [7], S2E [5], MIMIC [19], FlowPilot [54], and OpenPilot [55] as recent sidewalk navigation models trained specifically on urban sidewalk scenarios. Detailed statistics of the models can be found in Tab 1. Besides training data, the models also differ in the architecture with various vision encoders and action decoders as well as input frequency and model size. This allows us to examine the roles of both data and architecture play in the benchmark performance. For example, OpenPilot [55] has the smallest size of only 8M parameters, yet its inference frequency is the highest with 20 FPS. Conversely, the VLA-based InternVL-N1 [6] is the largest at 7B parameters and operates at a much slower rate of 5 FPS. More detailes of the models are presented in Appendix A

4.2 Unit-test Scenarios

We show results of the unit test scenarios on different topological structures of sidewalks in Fig. 4. Overall, the general visual navigation foundation models perform much worse than the models trained on sidewalk-specific data. For example, ViNT [3], which is trained on diverse data sources, achieve 0.21 and 0.33 route completion rate on the straight blocks of procedurally generated scenes and real-world scanned scenes, while MIMIC [19] with only 50 hours of sidewalk training data achieves a better performance with 0.59 and 0.39 route completion rate in the same setting. This underlines the importance of sidewalk-specific data in model training.

Moreover, the data-scaling effect applies to sidewalk navigation. For instance, FlowPilot [54], trained on 300 hours of data, outperforms MIMIC and achieves a route completion of 0.83 and 0.88 on the straight blocks. This is further surpassed by OpenPilot [55], which achieves a 0.87 and 0.96 average route completion rate using 1,000 hours of training data.

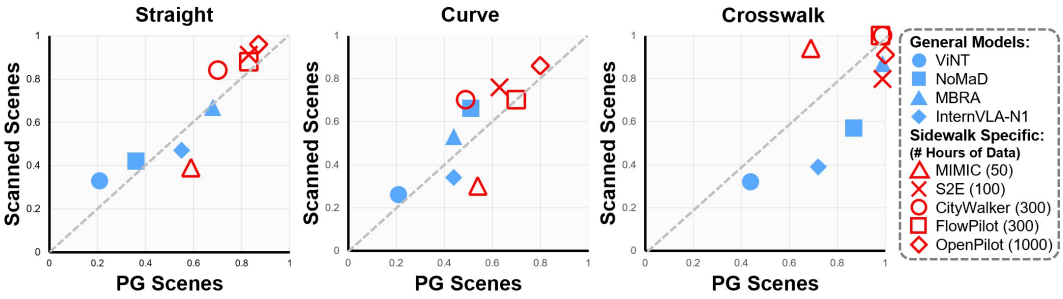

Figure 4: **Evaluation results of unit-test scenarios.** We show route completion rate on different sidewalk structures in both procedurally generated (PG) and real-world scanned scenes.

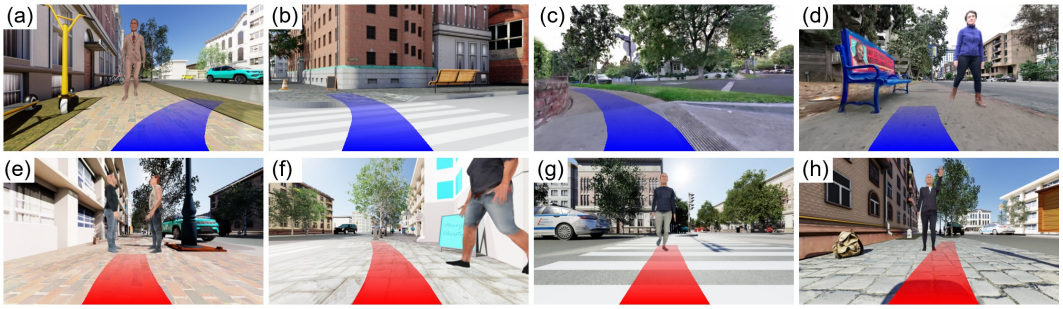

Figure 5: **Qualitative results**. (a)-(d): Success examples. (a) Avoiding a frontal approaching pedestrian. (b) Identifying the ramp. (c) Following the sidewalk curve. (d) Narrow passing, (e)-(h): Typical failure cases. (e) Ignoring social groups. (f) Failure to yield to a laterally crossing pedestrian. (g) Colliding with pedestrians on the crosswalk. (h) Inability to interpret pedestrian gestures.

Fig. 4 also demonstrates a strong correlation between performance in procedurally generated scenes and real-world scanned scenes. This implies that despite the real-to-sim visualization gap, procedurally generated environments remain highly useful for evaluating a model's real-world deployment performance. More results are presented in Appendix B.

4.3 Pedestrian-reactive Scenarios

As shown in Tab. 2, we evaluate the model's performance under different sidewalk pedestrian behaviors introduced in Sec. 3.2. From the comparison between the 'Static' scenario and the 'Obstructing' scenario, we can see that even a single static pedestrian could bring significant challenge to most models, with the average success rate dropping from 0.42 to 0.23. From the comparison between the 'Obstruction' scenario and the 'Conversing' and 'Queuing' scenarios, we observe that identifying social groups introduces additional challenges, and the average success rate drops further from 0.23 to 0.16 and 0.14, respectively.

Next, the 'Frontal', 'Lateral', and 'Overtaking' scenarios represent the three possible directions in which a pedestrian can approach the robot. Their comparison results indicate that the lateral direction is the most challenging, requiring the robot to yield instantly to avoid collision due to the limited camera field of view. The other two directions are less challenging because the model has enough time to yield to the pedestrian or take a detour.

The 'Ped-Crossing' scenario ends up being the most challenging scenario with only a 0.01 success rate. This highlights the significant challenge of compounding factors: pedestrians approaching from multiple directions, adhering to the crosswalk lane, and locating the sidewalk ramp. The 'Gesturing' scenario is almost equally challenging, with most methods failing to stop before the pedestrian. This underlines the lack of human gesture understanding with existing models, including the VLA-based model InternVLA-N1 [6]. Some common failure modes are visualized in Fig. 5, and more results and analysis are presented in Appendix C.

Method	Static	Obstructing	Conversing	Queueing	Frontal	Lateral	Overtaking	Ped-Crossing	Gesturing
ViNT [3]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NoMaD [18]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MBRA [4]	0.41	0.18	0.04	0.03	0.19	0.00	0.16	0.01	0.00
InternVLA-N1 [6]	0.30	0.19	0.18	0.07	0.22	0.05	0.12	0.01	0.00
MIMIC [19]	0.39	0.06	0.01	0.01	0.01	0.07	0.52	0.00	0.27
S2E [5]	0.69	0.52	0.40	0.28	0.17	0.03	0.66	0.00	0.00
CityWalker [7]	0.48	0.05	0.02	0.01	0.00	0.22	0.55	0.00	0.00
FlowPilot [54]	0.70	0.42	0.39	0.45	0.45	0.08	0.60	0.11	0.12
OpenPilot [55]	0.78	0.65	0.44	0.40	0.38	0.61	0.77	0.00	0.09
Avg	0.42	0.23	0.16	0.14	0.16	0.12	0.38	0.01	0.05

Table 2: **Model success rates of pedestrian-reactive scenarios**. The scenarios follow the definitions in Fig. 3, and **Static** denotes a scenario without pedestrians as reference. We highlight the top-three models from dark to light blue.

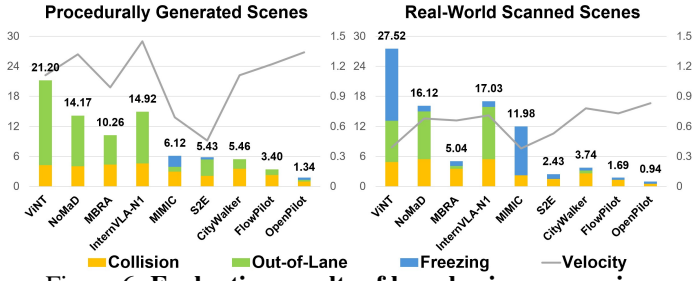

Figure 6: **Evaluation results of long-horizon scenarios**.

Scenario	Before	After
Ped-Crossing (Sim)	0.11	**0.69**
Ped-Crossing (Real)	0.00	**0.40**
Gesturing (Sim)	0.12	**0.49**
Gesturing (Real)	0.00	**0.50**

Table 3: Success rates of FlowPilot [54] before and after finetuning using our simulation platform for synthetic data generation.

4.4 Long-horizon Scenarios

The long-horizon navigation performance of the evaluated models is presented in Fig. 6. We can observe similar trends in procedurally-generated scenes and real-world scanned scene, and OpenPilot [55]—trained on 1,000 hours of sidewalk navigation data—outperforms all the other models with the lowest failure rate of 1.34 per 100 meters and the second highest average speed of 1.34 m/s in procedurally-generated scenes. It is also worth noting that OpenPilot only uses a lightweight vision encoder paired with a simple regression-based action decoder. These results underscore the importance of scaling sidewalk training data for robust long-horizon navigation, which outweighs other factors such as model architecture. Nevertheless, our results imply that deploying Openpilot on a food-delivery robot, which averagely needs to drive 2,000 meters to reach the target [56], would still require 26.6 human takeovers, averaging roughly 1.07 interventions per minute, demonstrating that a substantial gap remains for future research before achieving fully autonomous operation. We visualize the failure cases of the two best performing models, FlowPilot [54] and OpenPilot [55] in the long-horizon scenarios in Fig. 7. We can see that the failures in long-horizon scenarios combines those in the unit-test and pedestrian-reactive scenarios, and existing models still lack the robustness for long-horizon sidewalk navigation.

Figure 6 also demonstrates that average speed decreases in real-world scanned scenes, with an increase in freezing-induced failures. This is primarily because real-world terrains are rougher and more prone to trapping the robot, unlike the simulation terrains that are mostly flat ground. Future work will focus on increasing terrain complexity in procedural generation to narrow this sim-to-real gap. More results are available in Appendix D.

4.5 Synthetic Training Data Generation

As analyzed in the previous sections, scaling sidewalk training data is crucial for model performance. While collecting real-world pedestrian interaction data is expensive, our simulation platform can serve as a scalable synthetic data generator for model training. To this end, we conduct a preliminary experiment by fine-tuning on the two lowest-performing pedestrian-reactive scenarios: Ped-Crossing and Gesturing, which utilizes an expert planner to generate ground-truth robot actions for the FlowPilot [54] model. The results shown in Tab. 3 and Fig. 8 demonstrate that fine-tuning yields substantial performance gains in both simulated environments and the real world, showing the

Figure 7: **Qualitative results of the long-horizon scenarios.** We visualize the failure cases of the two best performing models, FlowPilot [54] and OpenPilot [55]. (a) Failure to follow the crosswalk. (b) Collision when facing both a pedestrian and an obstacle in the front. (c) Out of the boundary while turning. We strongly encourage viewing these results as videos on the project page.

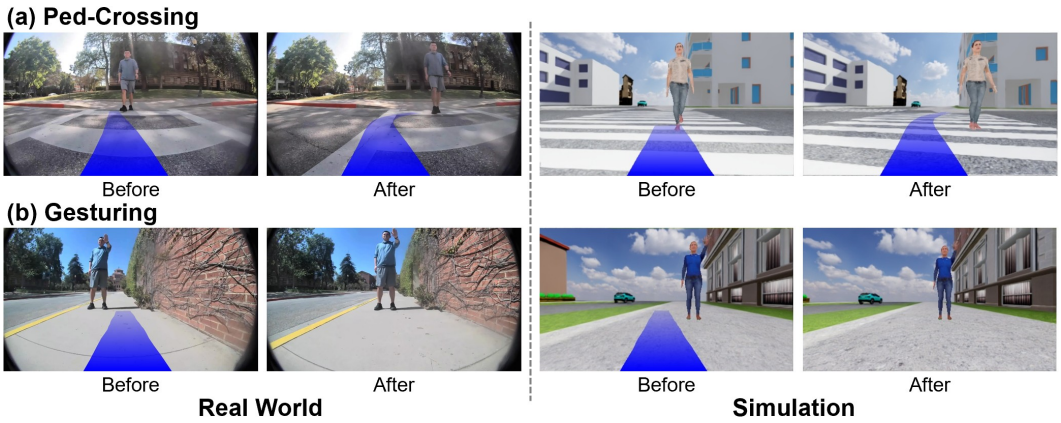

Figure 8: **Qualitative results of finetuning with synthetic data.** We visualize the model predictions of FlowPilot [54] before and after finetuning using our simulation platform for synthetic data generation. We strongly encourage viewing these results as videos on the project page.

huge potential of scaling synthetic data to address current models' limitations in pedestrian behavior understanding and long-horizon robustness. More details are presented in Appendix E.

5 Conclusion

We introduce SidewalkBench, a comprehensive benchmark for evaluating visual navigation on urban sidewalks. By simulating diverse and realistic sidewalk environments and further populating these environments with rich and reactive pedestrian behaviors and movements, SidewalkBench allows standardized testing and comparison of different models. To reflect challenges in real-world sidewalk navigation, we further introduce the unit-test, pedestrian-reactive, and long-horizon testing scenarios and conduct a comprehensive evaluation and analysis of representative visual navigation models. Our findings show that current models suffer from pedestrian interaction and long-term robustness while scaling-up synthetic training data is a promising solution, and further research is needed before these models can be safely deployed in the real world.

6 Limitations

First, our event-based pedestrian behaviors are governed by rule-based trajectories. While allowing standardized testing, these behaviors could lack realism compared to the more diverse and subtle robot-pedestrian interaction in the real world. Second, our pedestrian animation pipeline acts as a standalone module and trades off visual quality for efficiency and flexibility, which could introduce lighting artifacts and lead to a larger real-to-sim gap. Lastly, our benchmark currently only has a limited number of real-world scanned scenes and we plan to collect more in the future.

References

[1] R. L. Abduljabbar, S. Liyanage, and H. Dia. The role of micro-mobility in shaping sustainable cities: A systematic literature review. *Transportation research part D: transport and environment*, 92:102734, 2021.

[2] W. Daamen and S. P. Hoogendoorn. Experimental research of pedestrian walking behavior. *Transportation research record*, 1828(1):20–30, 2003.

[3] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023.

[4] N. Hirose, L. Ignatova, K. Stachowicz, C. Glossop, S. Levine, and D. Shah. Learning to drive anywhere with model-based reannotation. *IEEE Robotics and Automation Letters*, 11(2): 1242–1249, 2025.

[5] H. He, Y. Ma, W. Wu, and B. Zhou. From seeing to experiencing: Scaling navigation foundation models with reinforcement learning. *arXiv preprint arXiv:2507.22028*, 2025.

[6] M. Wei, C. Wan, J. Peng, X. Yu, Y. Yang, D. Feng, W. Cai, C. Zhu, T. Wang, J. Pang, and X. Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GK4rznYwhn.

[7] X. Liu, J. Li, Y. Jiang, N. Sujay, Z. Yang, J. Zhang, J. Abanes, J. Zhang, and C. Feng. Citywalker: Learning embodied urban navigation from web-scale videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6875–6885, 2025.

[8] A. Payandeh, A. Pokhrel, D. Song, M. Zampieri, and X. Xiao. Narrate2nav: Real-time visual navigation with implicit language reasoning in human-centric environments. *arXiv preprint arXiv:2506.14233*, 2025.

[9] Z. Huang, Y. Zhang, J. Liu, R. Song, C. Tang, and J. Ma. Tic-vla: A think-in-control vision-language-action model for robot navigation in dynamic environments. *arXiv preprint arXiv:2602.02459*, 2026.

[10] S. Pirk, E. Lee, X. Xiao, L. Takayama, A. Francis, and A. Toshev. A protocol for validating social navigation policies. *arXiv preprint arXiv:2204.05443*, 2022.

[11] N. Tsoi, A. Xiang, P. Yu, S. S. Sohn, G. Schwartz, S. Ramesh, M. Hussein, A. W. Gupta, M. Kapadia, and M. Vázquez. Sean 2.0: Formalizing and generating social situations for robot navigation. *IEEE Robotics and Automation Letters*, 7(4):11047–11054, 2022.

[12] N. Pérez-Higueras, R. Otero, F. Caballero, and L. Merino. Hunavsim: A ros 2 human navigation simulator for benchmarking human-aware robot navigation. *IEEE robotics and automation letters*, 8(11):7130–7137, 2023.

[13] NVIDIA. Isaac Sim. URL https://github.com/isaac-sim/IsaacSim.

[14] M. Liu, H. He, E. Ricci, W. Wu, and B. Zhou. Urbanverse: Scaling urban simulation by watching city-tour videos. In *The Fourteenth International Conference on Learning Representations*, 2026.

[15] W. Wu, H. He, C. Zhang, J. He, S. Z. Zhao, R. Gong, Q. Li, and B. Zhou. Towards autonomous micromobility through scalable urban simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[16] B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[17] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

[18] A. Sridhar, D. Shah, C. Glossop, and S. Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024.

[19] H. He, Y. Ma, B. Squicciarini, W. Wu, and B. Zhou. Learning sidewalk autopilot from multi-scale imitation with corrective behavior expansion. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026.

[20] M. Wei, C. Wan, J. Peng, X. Yu, Y. Yang, D. Feng, W. Cai, C. Zhu, T. Wang, J. Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. *arXiv preprint arXiv:2512.08186*, 2025.

[21] M. Wei, C. Wan, X. Yu, T. Wang, Y. Yang, X. Mao, C. Zhu, W. Cai, H. Wang, Y. Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

[22] W. Cai, J. Peng, Y. Yang, Y. Zhang, M. Wei, H. Wang, Y. Chen, T. Wang, and J. Pang. Navdp: Learning sim-to-real navigation diffusion policy with privileged information guidance. *arXiv preprint arXiv:2505.08712*, 2025.

[23] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025.

[24] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

[25] K. Yadav, J. Krantz, R. Ramrakhya, S. K. Ramakrishnan, J. Yang, A. Wang, J. Turner, A. Gokaslan, V.-P. Berges, R. Mootaghi, O. Maksymets, A. X. Chang, M. Savva, A. Clegg, D. S. Chaplot, and D. Batra. Habitat challenge 2023. https://aihabitat.org/challenge/2023/, 2023.

[26] Z. Gong, T. Hu, R. Qiu, and J. Liang. From cognition to precognition: A future-aware framework for social navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9122–9129. IEEE, 2025.

[27] H. Li, M. Li, Z.-Q. Cheng, Y. Dong, Y. Zhou, J.-Y. He, Q. Dai, T. Mitamura, and A. G. Hauptmann. Human-aware vision-and-language navigation: Bridging simulation to reality with dynamic human interactions. *Advances in Neural Information Processing Systems*, 37: 119411–119442, 2024.

[28] Z. Xie, Z. Liu, Z. Peng, W. Wu, and B. Zhou. Vid2sim: Realistic and interactive simulation from video for urban navigation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1581–1591, 2025.

[29] X. Liu, J. Li, Y. Deng, R. Chen, Y. Zhang, Y. Ma, L. Guo, Y. Li, J. Zhang, and C. Feng. Wanderland: Geometrically grounded simulation for open-world embodied ai. *arXiv preprint arXiv:2511.20620*, 2025.

[30] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni. Socnavbench: A grounded simulation testing framework for evaluating social navigation. *ACM Transactions on Human-Robot Interaction (THRI)*, 11(3):1–24, 2022.

[31] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[32] X. Ge, Y. Pan, Y. Zhang, X. Li, W. Zhang, D. Zhang, Z. Wan, X. Lin, X. Zhang, J. Liang, et al. Airsim360: A panoramic simulation platform within drone view. *arXiv preprint arXiv:2512.02009*, 2025.

[33] W. Wu, H. He, J. He, Y. Wang, C. Duan, Z. Liu, Q. Li, and B. Zhou. Metaurban: An embodied ai simulation platform for urban micromobility. *International Conference on Learning Representation*, 2025.

[34] C. Gao, B. Zhao, W. Zhang, J. Zhang, J. Mao, Z. Zheng, F. Man, J. Fang, Z. Zhou, J. Cui, X. Chen, and Y. Li. Embodiedcity: A benchmark platform for embodied agent in real-world city environment. *arXiv preprint*, 2024.

[35] Y. Zhuang, J. Ren, X. Ye, J. Shen, T. Yue, M. Faayez, X. He, X. Zhang, Z. Ma, L. Qin, et al. Simworld-robotics: Synthesizing photorealistic and dynamic urban environments for multimodal robot navigation and collaboration. *Advances in Neural Information Processing Systems*, 38:51854–51895, 2026.

[36] L. Kästner. Arena 5.0: A photorealistic ros2 simulation framework for developing and benchmarking social navigation. In *Learning to Simulate Robot Worlds*.

[37] M. Escudero-Jiménez, N. Pérez-Higueras, A. Martínez-Silva, F. Caballero, and L. Merino. Hunavsim 2.0: An enhanced human navigation simulator for human-aware robot navigation. *arXiv preprint arXiv:2507.17317*, 2025.

[38] Unity Technologies. *Unity Game Engine*. Unity Technologies, San Francisco, CA, 2026. Version 2023.2, Available at https://unity.com.

[39] Epic Games. *Unreal Engine*. Epic Games, Cary, NC, 2026. Version 5.4, Available at https://www.unrealengine.com.

[40] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Muñoz, X. Yao, R. Zurbrügg, N. Rudin, L. Wawrzyniak, M. Rakhsha, A. Denzler, E. Heiden, A. Borovicka, O. Ahmed, I. Akinola, A. Anwar, M. T. Carlson, J. Y. Feng, A. Garg, R. Gasoto, L. Gulich, Y. Guo, M. Gussert, A. Hansen, M. Kulkarni, C. Li, W. Liu, V. Makoviychuk, G. Malczyk, H. Mazhar, M. Moghani, A. Murali, M. Noseworthy, A. Poddubny, N. Ratliff, W. Rehberg, C. Schwarke, R. Singh, J. L. Smith, B. Tang, R. Thaker, M. Trepte, K. V. Wyk, F. Yu, A. Millane, V. Ramasamy, R. Steiner, S. Subramanian, C. Volk, C. Chen, N. Jawale, A. V. Kuruttukulam, M. A. Lin, A. Mandlekar, K. Patzwaldt, J. Welsh, H. Zhao, F. Anes, J.-F. Lafleche, N. Moënne-Loccoz, S. Park, R. Stepinski, D. V. Gelder, C. Amevor, J. Carius, J. Chang, A. H. Chen, P. de Heras Ciechomski, G. Daviet, M. Mohajerani, J. von Muralt, V. Reutskyy, M. Sauter, S. Schirm, E. L. Shi, P. Terdiman, K. Vilella, T. Widmer, G. Yeoman, T. Chen, S. Grizan, C. Li, L. Li, C. Smith, R. Wiltz, K. Alexis, Y. Chang, D. Chu, L. J. Fan, F. Farshidian, A. Handa, S. Huang, M. Hutter, Y. Narang, S. Pouya, S. Sheng, Y. Zhu, M. Macklin, A. Moravanszky, P. Reist, Y. Guo, D. Hoeller, and G. State. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025. URL https://arxiv.org/abs/2511.04831.

[41] A. Escontrela, J. Kerr, A. Allshire, J. Frey, R. Duan, C. Sferrazza, and P. Abbeel. Gaussgym: An open-source real-to-sim framework for learning locomotion from pixels. *arXiv preprint arXiv:2510.15352*, 2025.

[42] XGRIDS. Portal Cam. https://xgrids.com/us/portalcam, 2026. Accessed: May 23, 2026.

[43] A. Francis, C. Pérez-d'Arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra, et al. Principles and guidelines for evaluating social robot navigation algorithms. *ACM Transactions on Human-Robot Interaction*, 14(2):1–65, 2025.

[44] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.

[45] D. Casas and M. Comino-Trinidad. Smplitex: A generative model and dataset for 3d human texture estimation from single image. *arXiv preprint arXiv:2309.01855*, 2023.

[46] A. Moreau, J. Song, H. Dhamo, R. Shaw, Y. Zhou, and E. Pérez-Pellitero. Human gaussian splatting: Real-time rendering of animatable avatars. In *CVPR*, 2024.

[47] W. Dai, L.-H. Chen, J. Wang, J. Liu, B. Dai, and Y. Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. In *European Conference on Computer Vision*, pages 390–408. Springer, 2024.

[48] J. Lin, A. Zeng, S. Lu, Y. Cai, R. Zhang, H. Wang, and L. Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 36:25268–25280, 2023.

[49] Z. Liu, J. Lin, W. Wu, and B. Zhou. Learning to generate diverse pedestrian movements from web videos with noisy labels. In *International Conference on Learning Representations*, volume 2025, pages 65682–65697, 2025.

[50] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020.

[51] NVIDIA Corporation. Isaac sim documentation: Actor control (character behavior), 2024. URL https://docs.isaacsim.omniverse.nvidia.com/5.1.0/action_and_event_data_generation/ext_replicator-agent/actor_control.html#ira-character-behavior. Accessed: May 23, 2026.

[52] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

[53] E. Kohanpour, S. R. Davoodi, and K. Shaaban. Trends in autonomous vehicle performance: A comprehensive study of disengagements and mileage. *Future Transportation*, 5(2):38, 2025.

[54] H. He, Z. Liu, Y. Ma, and B. Zhou. From imitation to alignment: Human-preference flow policies for long-horizon sidewalk navigation. *arXiv preprint*, 2026.

[55] comma.ai. openpilot: open source advanced driver assistance system. https://github.com/commaai/openpilot, 2026. Accessed: 2026-05-20.

[56] D. Jennings and M. Figliozzi. Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel. *Transportation Research Record*, 2673(6):317–326, 2019.

[57] B. De Wilde, A. W. Ter Mors, and C. Witteveen. Push and rotate: a complete multi-agent pathfinding algorithm. *Journal of Artificial Intelligence Research*, 51:443–492, 2014.

[58] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics research: the 14th international symposium ISRR*, pages 3–19. Springer, 2011.

[59] Coco Robotics. Coco robotics, 2026. URL https://www.cocodelivery.com/. Accessed: 2026-06-01.

[60] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[61] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Appendix

We present the implementation details of SidewalkBench in Sec. A. More results and analysis of unit-test scenarios, pedestrian-reactive scenarios, and long-horizon scenarios are demonstrated in Sec. B, Sec. C and Sec. D, respectively. We present additional details of synthetic training data generation in Sec. E. More video results are available on the project page.

A Implementation Details

Pedestrian Simulation Detailed descriptions of the event-based behaviors are list in Tab. 4. Note that both the trajectory and the speed of the pedestrian are specified by hand-crafted rules in event-based behaviors to ensure the robot has to react correspondingly to the pedestrian. For example, in the scenario "Lateral Approaching", we control the trajectory of the pedestrian so it is perpendicular to the current robot heading and adjust the speed so the robot would collide with the pedestrian if it keeps the currrent velocity. For other regular pedestrian behaviors like walking towards a target point, we follow MetaUrban [33] to use the Push and Rotate algorithm [57] for global path generation and ORCA [58] for local collision avoidance. The pedestrian speed is sampled with common walking speed (1.1 m/s – 1.65 m/s).

For the local pedestrian movements, we use an efficient motion generation model Motion-LCM [47] to generate common pedestrian movements such as walking and standing with high diversity on-the-fly during simulation. For less common movements such as a stopping gesture, we use exist human motion datasets such as Motion-X [48] to retrieve the corresponding motion. As mentioned in the main paper, our pedestrian animation is highly efficient, and a comparison of the rendering efficiency is shown in Fig. 9, which shows the amortized rendering FPS including motion generation, rigging, skinning, and final rendering. As our pedestrian animation is controlled by the SMPL model [44], we developed two different rendering approaches. For the procedurally-generated scenes, we use mesh-based rendering that applies SMPL-compatible human textures [45] to the SMPL body mesh. It can achieve a remarkable 226.8 FPS in a single environment with 50 pedestrians, which is much higher than prior works, including the native human animation pipeline [51] in Issac Sim with only a 3.4 FPS. This drastic improvement facilitates highly efficient simulation and evaluation of large, crowded urban scenes. For the real-world scanned scenes, we use SMPL 3DGS avatars [16] with better fidelity and are more compatible with the 3DGS background scene. Our GS-based pedestrian rendering can still achieve a 10.7 FPS in the same setting and is comparable to other simulators like MetaUrban [33] with a 7.5 FPS. Since we use an independent pedestrian simulation pipeline, the pedestrian collision detections are also handled separately in our simulation platform, where we use a cylinder of radius 0.3m to represent the collision mesh of pedestrians to improve efficiency. This is reasonable for visual navigation that does not involve physical interactions with humans.

Testing Scenario Configurations We use a four-wheeled delivery robot [59] as the main robot platform for our benchmark due to its simple dynamics and strong practicality in the real world. The robot has a maximum speed of 2.5 m/s and a maximum angular velocity of 0.65 rad/s. Our simulation platform also supports the evaluation on other robot embodiments such as the robot dog or the humanoid (shown in Fig. 10). Following existing works [3, 7, 5], we use the same PD controller to convert the robot waypoints predicted by the model into uniform velocity commands including the linear and the angular velocity. To ensure reproducible results, we used synchronized testing, i.e. the simulator would wait for a robot velocity command at each step and ignore the inference latency. It is worth noting that the models do have large discrepancies in inference speed and could lead to different results in real-world deployment as shown in Tab. 1. All benchmarking experiments are conducted in 10 parallel environments on a NVIDIA L40S GPU in Isaac-Sim 6.0 [13]. The physics simulation step is set to 0.005s to ensure high physics realism. For the 'Gesturing' scenario, the success criteria is the robot stops within 3m in front of the pedestrian instead of reaching the goal. For the 'Conversing' and the 'Queueing' scenarios, we only count success when the robot avoids the whole pedestrian group instead of passing between the pedestrians.

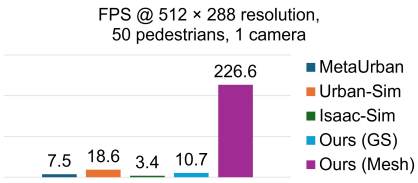

Figure 9: **Comparisons of the pedestrian rendering efficiency**. We report the amortized rendering FPS of pedestrians including motion generation, rigging, skinning, and rendering.

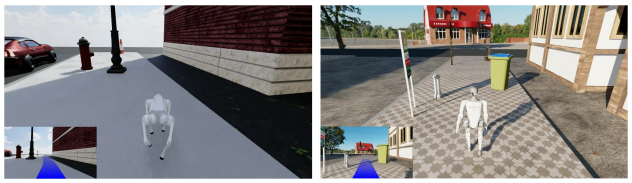

Figure 10: **Qualitative results of other robot embodiments**. We show examples of the Unitree Go2 robot dog (left) and the Unitree G1 humanoid (right) evaluated with the FlowPilot [54] model. Videos results are available on the project page.

Scenario	Description	Triggering Condition
Obstructing	A stationary pedestrian stands at the center of the sidewalk looking around.	Within 10m ahead of the robot
Conversing	Multiple pedestrians stand facing each other at the center of the path, forming a social group.	Within 10m ahead of the robot
Queueing	Pedestrians form a structured line starting from one side of the sidewalk with tight spacing.	Within 10m ahead of the robot
Frontal Approaching	A pedestrian walks directly toward the current position of the robot from the opposite direction.	Within 20m ahead of the robot
Lateral Approaching	A pedestrian crosses the robot's projected path laterally, requiring the robot to yield.	Within 4m ahead of the robot
Overtaking	A pedestrian approaches and passes the robot from behind with a faster speed.	Within 1m behind the robot
Ped-Crossing	One or more pedestrians crossing from front and behind across a designated crosswalk.	Robot enters the crosswalk zone
Gesturing	A pedestrian continuously faces the robot while executing a distinct stopping gesture.	Within 10m ahead of the robot

Table 4: **Descriptions of event-based pedestrian behaviors.**

Models During testing, we adjust the inference frequency and the image resolution of simulation to match the specifications of each model. For evaluation, we mainly focus on the navigation-relevant capabilities of each model, including visual scene understanding, sidewalk and obstacle awareness, and socially compliant behavior around pedestrians. We do not directly evaluate goal-following ability as an isolated capability, since providing explicit global goals, dense route commands, or future observations can leak privileged task information in many scenarios. For goal-free models, including FlowPilot [54] and OpenPilot [55], we use visual observations as the only input. For goal-oriented models, we provide goal information according to their original input modalities while avoiding direct leakage of future trajectories or task-specific privileged information. Specifically, for point-goal-based models, including MBRA [4], MIMIC [19], S2E [5] and CityWalker [7], we randomly sample intermediate goals in the robot-centric frame, with the longitudinal distance sampled from $(5, 20)$ meters and the lateral offset sampled from $(-5, 5)$ meters. For image-goal-based models, including ViNT [3] and NoMaD [18], we provide randomly generated images as visual goal inputs instead of using future observations sampled from the reference route. This protocol allows each model to operate under its intended interface while reducing unfair advantages from explicit future information. For the VLA model InternVLA-N1 [6], we use a goal-agnostic navigation instruction, "Follow the sidewalk" to avoid leaking route-specific or task-specific information. This instruction is inserted into the model's standard navigation prompt, which asks the agent to predict the next waypoint in the image or output `STOP` when the task is completed. At each control step, we provide the current RGB observation to InternVLA-N1 and extract its latent output, which is then passed to its diffusion-based trajectory decoder NavDP [22] to produce continuous waypoints in the robot frame. These waypoints are converted to velocity commands using the same low-level controller interface as the other trajectory-output baselines.

B More Results of Unit-test Scenarios

Full evaluation results of the unit-test scenarios are shown in Tab. 5, where we report success rate (SR) and success weighted by path length (SPL) [60] as addition metrics commonly used in visual navigation besides route completion rate (RC). We can see that the success rates for ViNT [3] and NoMaD [18] are almost 0 under all settings, meaning these two models lack the basic lane following and obstacle avoidance abilities on urban sidewalks despite being a general foundation model for visual navigation. In contrast, MBRA [4] achieves significantly better performance, with the highest SR and SPL of 0.41 and 0.40 among all the general navigation models on the straight blocks in procedurally-generated scenes. It is worth mentioning that the main difference between MBRA and

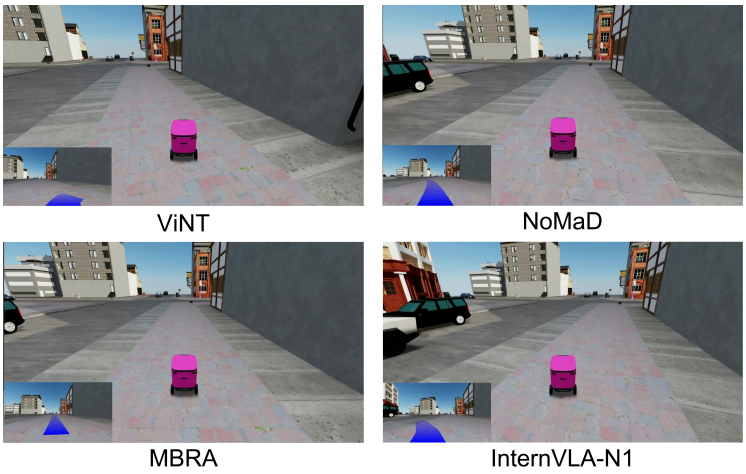

Figure 11: **Qualitative comparisons of the unit-test scenarios.** ViNT [3], NoMaD [18], and InternVLA-N1 [6] are not able to follow the straight sidewalk lane. Only MBRA [4] has some basic lane-following ability. We strongly encourage viewing these results (and other models) as videos on the project page.

	Procedurally Generated Scenes									**Real-World Scanned Scenes**								
	Straight			**Curve**			**Crosswalk**			**Straight**			**Curve**			**Crosswalk**		
Method	**SR ↑**	**SPL ↑**	**RC ↑**	**SR ↑**	**SPL ↑**	**RC ↑**	**SR ↑**	**SPL ↑**	**RC ↑**	**SR ↑**	**SPL ↑**	**RC ↑**	**SR ↑**	**SPL ↑**	**RC ↑**	**SR ↑**	**SPL ↑**	**RC ↑**
ViNT [3]	0.00	0.00	0.21	0.00	0.00	0.21	0.00	0.00	0.44	0.00	0.00	0.33	0.00	0.00	0.26	0.00	0.00	0.32
NoMaD [18]	0.00	0.00	0.36	0.02	0.02	0.51	0.00	0.00	0.87	0.00	0.00	0.42	0.30	0.30	0.66	0.00	0.00	0.57
MBRA [4]	0.41	0.40	0.68	0.00	0.00	0.44	0.68	0.67	0.99	0.40	0.40	0.67	0.00	0.00	0.53	0.60	0.60	0.87
InternVLA-N1 [6]	0.30	0.29	0.55	0.01	0.01	0.44	0.01	0.01	0.72	0.00	0.00	0.47	0.00	0.00	0.34	0.00	0.00	0.39
CityWalker [7]	0.48	0.47	0.70	0.02	0.02	0.49	**0.99**	0.95	0.99	0.80	0.79	0.84	0.10	0.10	0.70	**1.00**	0.98	**1.00**
MIMIC [19]	0.39	0.39	0.59	0.23	0.23	0.54	0.56	0.55	0.69	0.30	0.30	0.39	0.00	0.00	0.30	0.40	0.40	0.94
S2E [5]	0.69	0.68	0.83	0.28	0.28	0.63	0.75	0.74	0.99	**0.90**	**0.90**	0.91	0.50	0.50	0.76	0.30	0.30	0.80
FlowPilot [19]	0.70	0.69	0.83	0.47	0.47	0.70	0.68	0.66	0.98	0.80	0.79	0.88	0.20	0.20	0.70	**1.00**	**1.00**	**1.00**
OpenPilot [55]	**0.78**	**0.77**	**0.87**	**0.57**	**0.57**	**0.80**	0.98	**0.97**	**1.00**	**0.90**	0.89	**0.96**	**0.60**	**0.59**	**0.86**	0.90	0.88	0.91

Table 5: **Full evaluation results of unit-test scenarios.**

the previous two models is that MBRA uses a specific data filtering procedure that ensures high label quality, which further highlights the importance of training data for sidewalk navigation.

Meanwhile, despite the heavy model architecture and slower inference speed, InternVLA-N1 [6] still underperforms compared to MBRA and the sidewalk-specific models. This shows that while VLM backbones offer promising high-level reasoning capabilities, current VLA-based models fall short in unit-testing scenarios that demand more on timely and precise control which could be learned from sidewalk-specific training data. Some qualitative comparisons of the general models are shown in Fig. 11.

We also observe a minimal gap between SR and SPL, implying all the models achieve a relatively high path efficiency. This is expected as sidewalk navigation requires the model to follow the lane without taking much detour except for avoiding obstacles, and thus the completion time weights more than the path length in the overall efficiency evaluation. Among the three structures, the curve block is the most challenging, which requires the robot to follow a curved path and avoid static obstacles. For instance, the best model OpenPilot [55] results in a 0.78, 0.57 and 0.98 SR on the straight, curve, and crosswalk blocks respectively. While the crosswalk has the highest success rate due to free of obstacles, we observe identifying the ramp at the end of the crosswalk remains challenging for many models (illustrated in Fig. 12, resulting in a high route completion rate but a lower success rate. For example,

Figure 12: Failure to identify the ramp at the end of the crosswalk.

Metric	Method	Obstructing	Conversing	Queueing	Frontal	Lateral	Overtaking	Ped-Crossing	Gesturing
Pedestrian Collision Rate ↓	ViNT [3]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	NoMaD [18]	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00
	MBRA [4]	0.42	0.50	0.61	0.48	0.76	0.42	0.98	0.31
	InternVLA-N1 [6]	**0.06**	0.11	0.30	**0.05**	0.33	0.17	**0.17**	0.11
	MIMIC [19]	0.13	**0.07**	**0.06**	0.78	0.58	0.01	0.94	**0.07**
	S2E [5]	0.12	0.32	0.44	0.69	0.85	**0.00**	1.00	0.11
	CityWalker [7]	0.73	0.74	0.76	0.78	0.41	**0.00**	1.00	0.62
	FlowPilot [54]	0.28	0.33	0.30	0.38	0.73	0.10	0.84	0.32
	OpenPilot [55]	0.13	0.36	0.31	0.52	**0.26**	**0.00**	1.00	0.36
	Avg	0.22	0.27	0.31	0.41	0.44	0.10	0.68	0.21
Minimum Pedestrian Distance (m) ↑	ViNT [3]	–	–	–	–	–	–	–	–
	NoMaD [18]	–	–	–	–	–	–	–	–
	MBRA [4]	0.79	0.59	0.54	0.69	–	0.66	0.52	–
	InternVLA-N1 [6]	**1.15**	**1.36**	0.83	**1.43**	0.83	0.95	**0.65**	–
	MIMIC [19]	0.88	0.47	1.02	0.51	0.96	**0.97**	–	**2.01**
	S2E [5]	0.88	0.80	0.93	0.82	0.72	**0.97**	–	–
	CityWalker [7]	0.68	0.68	**1.19**	–	0.92	0.96	–	–
	FlowPilot [54]	0.95	0.81	0.75	0.92	0.80	0.85	0.61	1.37
	OpenPilot [55]	0.93	0.71	0.64	0.92	**1.03**	0.95	–	1.04
	Avg	0.88	0.77	0.84	0.88	0.86	0.90	0.59	1.47

Table 6: **More evaluation results of pedestrian-reactive scenarios**. We exclude ViNT [3] and NoMaD [18] from the model comparison as they achieve 0 success in all scenarios.

S2E [5] achieves a 0.99 route completion rate on crosswalks in procedurally generated scenes, but its success rate drops to only 0.75 in the same setting.

C More Results of Pedestrian-reactive Scenarios

More evaluation results of the pedestrian-reactive scenarios are demonstrated in Tab. 6. We additionally compute pedestrian collision rate and average minimum pedestrian distance in the successful trails for evaluating the social compliance of the model. Comparing to Tab. 1 in the main paper, we can see a strong correlation between a high success rate, a low pedestrian collision rate, and a high minimum pedestrian distance. For example, the easiest scenario 'Overtaking' has an average success rate of 0.38, a pedestrian collision rate of 0.10, and a minimum pedestrian distance of 0.90m, while the most challenging scenario ped-crossing has an average success rate of 0.01, a pedestrian collision rate of 0.68 and a minimum pedestrian distance of only 0.59m. This indicates that avoiding collision with pedestrians while maintaining a safe social distance is crucial for the successful completion of pedestrian-reactive scenarios.

Moreover, we find that InterVLA-N1 [6], a VLA-based model achieves low pedestrian collision rate and high minimum pedestrian distance in most scenarios despite a relatively low success rate. For example, InterVLA-N1 only achieves a 0.22 overall success rate on the challenging 'Frontal Approaching' scenario. However, its pedestrian collision rate of 0.05 and minimum pedestrian distance of 1.43m are significantly better than all the other methods, with the second place method having a 0.38 pedestrian collision rate and a 0.92m minimum pedestrian distance. Combining with the results in Tab. 5, we can conclude that while VLA-based models are not good at precise and timely control, they may still be useful in reasoning about the social situations and understanding the pedestrian intents. Therefore, an important future step is combing the low-level sidewalk navigation capabilities of the more light-weight model such as OpenPilot [55] trained on large-scale data and the high-level reasoning capabilities from the VLA to better handle the dynamic interaction with the pedestrians. Some qualitative comparisons of the models are shown in Fig. 13.

D More Results of Long-horizon Scenarios

Full evaluation results of the long-horizon scenarios are presented in Tab. 7, which corresponds to Fig. 6 in the main paper.

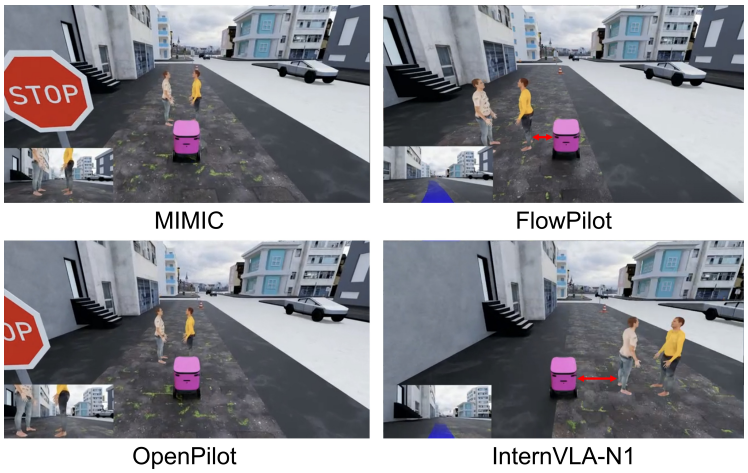

Figure 13: **Qualitative comparisons of the pedestrian-reactive scenarios.** MIMIC [19] and OpenPilot [55] freezes in front of a conversing pedestrian group. InternVLA-N1 [6] leaves a much larger space when passing the pedestrian group compared to FlowPilot [54], demonstrating its better social compliance capability than the sidewalk-specific models. We strongly encourage viewing these results (and other models) as videos on the project page.

	Procedurally Generated Scenes					Real-World Scanned Scenes				
Method	**FC ↓**	**C ↓**	**OL ↓**	**F ↓**	***V*** **↑**	**FC ↓**	**C ↓**	**OL ↓**	**F ↓**	***V*** **↑**
ViNT [3]	21.20	4.29	16.90	**0.00**	1.11	27.52	4.87	8.24	14.42	0.39
NoMaD [18]	14.17	4.03	10.14	**0.00**	1.32	16.12	5.44	9.56	1.12	0.68
MBRA [4]	10.26	4.36	5.91	**0.00**	0.99	5.04	3.55	0.56	0.93	0.66
InternVLA-N1 [6]	14.92	4.59	10.33	**0.00**	**1.45**	17.03	5.43	10.48	1.12	0.71
MIMIC [19]	6.12	2.96	0.99	2.16	0.69	11.98	2.25	**0.00**	9.73	0.38
S2E [5]	5.43	2.12	3.25	0.06	0.46	2.43	1.50	**0.00**	0.94	0.53
CityWalker [7]	5.46	3.53	1.93	**0.00**	1.11	3.74	2.62	0.56	0.56	0.78
FlowPilot [54]	3.40	2.30	1.10	**0.00**	1.22	1.69	1.31	**0.00**	**0.37**	0.73
OpenPilot [55]	**1.34**	**1.01**	**0.33**	0.01	1.34	**0.94**	**0.56**	0.00	**0.37**	**0.83**

Table 7: **Evaluation results of long-horizon scenarios.** We report the average failure counts (FC) per 100 meters traveled, including collision (C), out-of-lane (OL), and freezing (F). We also report the average velocity (V, m/s).

E Details of Synthetic Training Data Generation

For the experiments of synthetic training data generation, we collect 500 successful demonstration episodes for both the "Ped-Crossing" and "Gesturing" scenarios in our simulation platform. These demonstrations are generated under the same observation and action representation as the real policy, using ORCA [58] to generate collision-free reference trajectories and velocity commands for the robot. Next, we finetune FlowPilot [54] on the collected synthetic demonstrations. During finetuning, we freeze the visual backbone and only update the lightweight task-adaptation and action-generation modules, mainly the noisy action encoder and action decoder. We use AdamW [61] with a learning rate of 5×10^{-6} for all trainable modules. The learning rate is decayed with a cosine annealing schedule for 50 finetuning epochs, with a minimum learning rate of 1×10^{-6}. We also apply gradient clipping with a maximum norm of 1.0 to stabilize optimization. For the real-world experiments, we perform 10 trials for each scenario.