METAURBAN: AN EMBODIED AI SIMULATION PLATFORM FOR URBAN MICROMOBILITY

Anonymous authors
Paper under double-blind review

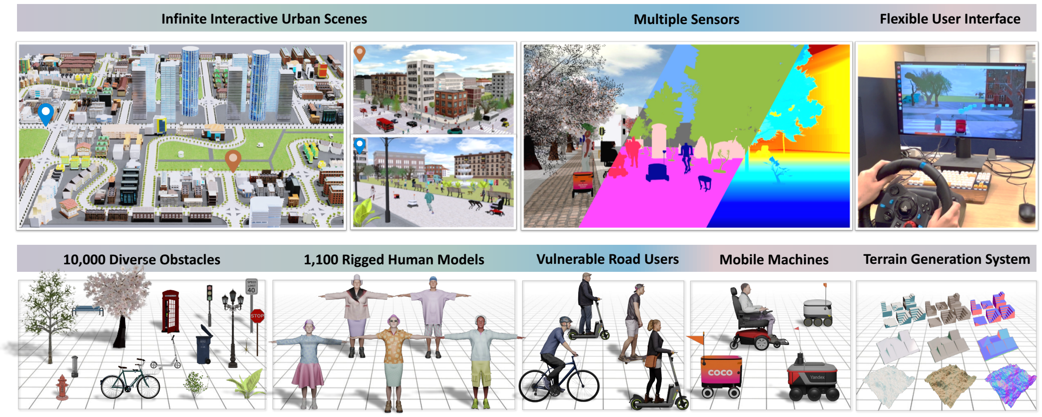


Figure 1: **MetaUrban** enables the construction of *infinite interactive urban scenes*, supports *multiple sensors*, and offers *flexible user interfaces* such as a mouse, keyboard, joystick, and racing wheel. The platform includes *10,000 diverse obstacles* in urban scenes, *1,100 rigged human models* each with 2,314 movements, *vulnerable road users*, *mobile machines* with varied mechanical structures, and a *terrain generation system* to create complex ground conditions. We highly recommend visiting our project page[1] for video demonstrations.

ABSTRACT

Public urban spaces like streetscapes and plazas serve residents and accommodate social life in all its vibrant variations. Recent advances in Robotics and Embodied AI make public urban spaces no longer exclusive to humans. Food delivery bots and electric wheelchairs have started sharing sidewalks with pedestrians, while robot dogs and humanoids have recently emerged in the street. **Micromobility** enabled by AI for short-distance travel in public urban spaces plays a crucial component in the future transportation system. Ensuring the generalizability and safety of AI models maneuvering mobile machines is essential. In this work, we present **MetaUrban**, a *compositional* simulation platform for the AI-driven urban micromobility research. MetaUrban can construct an *infinite* number of interactive urban scenes from compositional elements, covering a vast array of ground plans, object placements, pedestrians, vulnerable road users, and other mobile agents' appearances and dynamics. We design point navigation and social navigation tasks as the pilot study using MetaUrban for urban micromobility research and establish various baselines of Reinforcement Learning and Imitation Learning. We conduct extensive evaluation across mobile machines, demonstrating that heterogeneous mechanical structures significantly influence the learning and execution of AI policies. We perform a thorough ablation study, showing that the compositional nature of the simulated environments can substantially improve the generalizability and safety of the trained mobile agents. MetaUrban will be made publicly available to provide research opportunities and foster safe and trustworthy embodied AI and micromobility in cities. The code and dataset are released[2].

1 INTRODUCTION

Public urban spaces (Whyte, 2012) vary widely in type, form, and size, encompassing streetscapes, plazas, and parks. They are crucial spaces for transit and transport (Geddes, 1949), as well as

---

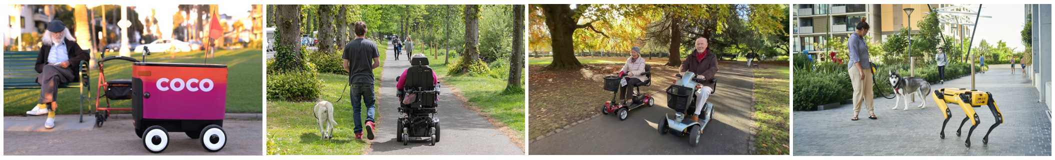
[1] Project Page (anonymous): `http://metaurban-iclr-2025.github.io`

[2] Code and Dataset (anonymous): `https://github.com/metaurban-iclr-2025/MetaUrban`

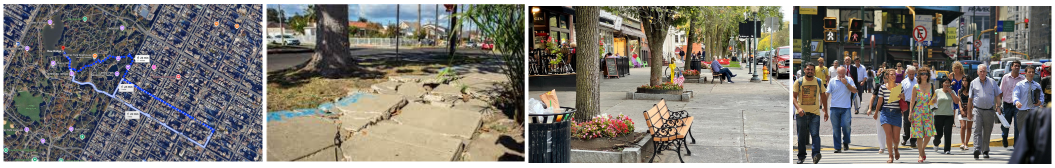
providing stages to host various social events (Park et al., 1925). In recent years, these spaces have also become key zones for the growing trend of **micromobility** (Mitchell et al., 2010; Abduljabbar et al., 2021; Oeschger et al., 2020), a term that refers to small, lightweight vehicles like electric scooters, e-bikes, and other mobile machines designed for short-distance travel. Micromobility is becoming an increasingly important solution for improving urban transportation efficiency, reducing environmental impact, and offering flexible alternatives to car ownership in cities.

As shown in Figure 2 (Top), food delivery bots navigate on the sidewalk to accomplish the last-mile food delivery task, while elders and physically disabled people maneuver electronic wheelchairs and mobility scooters on the street. Various legged robots like robot dog Spot from Boston Dynamics and humanoid robot Optimus from Tesla are also forthcoming. We can imagine a future of such *automated micromobility* that harnesses advanced AI models to improve situational awareness and maneuver various mobile machines more intelligently and safely in complex urban environments.

Simulation platforms have played a crucial role in enabling systematic and scalable training of embodied AI agents and safety evaluation before real-world deployment. However, most of the existing simulators focus either on *indoor household environments* (Puig et al., 2018; Kolve et al., 2017; Savva et al., 2019; Shen et al., 2021; Li et al., 2024; Gan et al., 2021) or *outdoor driving environments* (Krajzewicz et al., 2002; Li et al., 2022b; Dosovitskiy et al., 2017). For example, platforms like AI2-THOR (Kolve et al., 2017), Habitat (Savva et al., 2019), and iGibson (Shen et al., 2021) are designed for household assistant robots in which the environments are mainly apartments or houses with furniture and appliances; platforms like SUMO (Krajzewicz et al., 2002), CARLA (Dosovitskiy et al., 2017), and MetaDrive (Li et al., 2022b) are designed for research on autonomous driving and transportation focusing on roadways and highways. Yet, simulating complex *urban environments* for micromobility tasks, with diverse layouts, terrains, obstacles, and complex dynamics of pedestrians, is much less explored.



Delivery Robot — Electric Wheelchair — Mobility Scooter — Robot Dog



Long Horizon Tasks in Large Scale Scenes — Multifarious Terrains — Diverse Obstacles — Dense Pedestrians

Figure 2: **Motivation.** (Top) Emerging automated micromobility. (Bottom) Unique challenges in micromobility.

Distinct from the indoor household and driving tasks, micromobility plays an essential role in providing accessibility (*e.g.*, electric wheelchairs) and convenience (*e.g.*, food delivery bots) in the public urban space, while it also brings ***unique challenges*** for mobile machines and the underlying embodied AI agents. Let's follow the adventure of a last-mile delivery bot, who aims to deliver a lunch order from a nearby pizzeria to the campus (Figure 2 (Bottom)). First, it faces a long-horizon task in **large scale** scenes across several street blocks, which span a significantly larger space than the indoor household environment. Second, it needs to deal with **multifarious terrains**, such as fragmented curbs and rugged ground caused by tree roots on sidewalks, which are seldom seen in indoor and driving environments. Then, it must safely navigate the cluttered street full of **diverse obstacles** like trash bins, parked scooters, and potted plants, which is absent in driving scenarios while with large obstacles variations compared to indoor scenes. In addition, it needs to handle **dense pedestrians** on sidewalks to avoid collisions, especially taking care of disabled people in wheelchairs, which do not exist in indoor and driving environments. Thus, the large scales, multifarious terrains, diverse obstacles, and dense pedestrians bring unique challenges to AI-driven mobile machines moving in cities, as well as the design of simulation environments for the training and evaluation of the embodied AI models.

In this work, we present **MetaUrban** – a compositional simulation platform aiming to facilitate the research of AI-driven micromobility. First, we introduce *Hierarchical Layout Generation*, a procedural generation approach that can generate infinite layouts hierarchically from street blocks to sidewalks, functional zones, and object locations. It can generate scenes at an arbitrary scale with various connections and divisions of street blocks, obstacle locations, and complex terrains. Then, we design the *Scalable Obstacle Retrieval*, an automatic pipeline for acquiring an arbitrary number of high-quality objects with real-world distribution, to fill the urban space. We first compute the object category distribution from worldwide urban scene data to form a description pool. Then, with the sampled descriptions from the pool, we design a VLM-based open-vocabulary searching schema, which can effectively retrieve objects from large-scale 3D asset repositories. These two modules are critical for improving the *generalizability* of trained agents.

Finally, we propose the *Cohabitant Populating* method to generate complex dynamics in urban spaces. We first tailor recent 3D digital human and motion datasets to get 1,100 rigged human models, each with 2,314 movements. Then, to form safety-critical scenarios, we integrate Vulnerable Road Users (VRUs) like bikers, skateboarders, and scooter riders. As the subjects of micromobility, we include various mobile machines – the delivery bot, electric wheelchair, mobility scooter, robot dog, and humanoid robot. Then, based on path planning algorithms, we can get complex trajectories among hundreds of environmental agents simultaneously with collision and deadlock avoidance. Also, enabled by MetaUrban's flexible user interfaces (mouse, keyboard, joystick, and racing wheel), users can directly apply human-operated trajectories to agents, which provides an easy way to collect demonstration data for agent training. In addition, we impose a series of traffic rules to regulate all agent behaviors. It is critical for enhancing the *safety* of the mobile agents.

Based on MetaUrban, we construct a large-scale dataset, MetaUrban-12K, that includes 12,800 training scenes and 1,000 test scenes. The mean area size is 20,000m^2, while the episode length is 410m on average. As a pilot study, we introduce Point Navigation and Social Navigation, which are the two most fundamental tasks for mobile machines moving in urban spaces, as a starting point for AI-driven micromobility research. We build comprehensive benchmarks for these two tasks, in which we establish extensive baseline models, covering Reinforcement Learning, Safe Reinforcement Learning, Offline Reinforcement Learning, and Imitation Learning. Then, we make extensive evaluations across mobile machines to delve into the performance influence of varied mechanical structures (such as engine force, wheel friction, and wheelbase) on the learning and execution of AI policies. In the ablation study, we demonstrate that the compositional nature of the simulated environments can substantially improve the generalizability and safety of the trained mobile agents. We will make MetaUrban publicly available to enable more research opportunities for the community and foster safe and trustworthy embodied AI and micromobility in cities.

2 RELATED WORK

Many simulation platforms have been developed for embodied AI research, depending on the target environments – such as indoor homes and offices, driving roadways and highways, and crowds in warehouses and squares. We compare representative ones with the proposed MetaUrban simulator.

Indoor Environments. Platforms for indoor environments are mainly designed for household assistant robots, emphasizing the affordance, realism, and diversity of objects, as well as the interactivity of environments. VirtualHome (Puig et al., 2018) pivots towards simulating routine human activities at home. AI2-THOR (Kolve et al., 2017) and its extensions, such as ManipulaTHOR (Ehsani et al., 2021), RoboTHOR (Deitke et al., 2020), and ProcTHOR (Deitke et al., 2022b), focus on detailed agent-object interactions, dynamic object state changes, and procedural scene generation, alongside robust physics simulations. Habitat (Savva et al., 2019) offers environments reconstructed from 3D scans of real-world interiors. Its subsequent iterations, Habitat 2.0 (Szot et al., 2021) and Habitat 3.0 (Puig et al., 2023b), introduce interactable objects and deformable humanoid agents, respectively. iGibson (Shen et al., 2021) provides photorealistic environments. Its upgrades, Gibson 2.0 (Li et al., 2021), and OmniGibson (Li et al., 2024), focus on household tasks with object state changes and a realistic physics simulation of everyday activities, respectively. ThreeDWorld (Gan et al., 2021) targets real-world physics by integrating high-fidelity simulations of liquids and deformable objects. However, unlike MetaUrban, these simulators are focused on indoor environments with particular tasks like object rearrangement and manipulation.

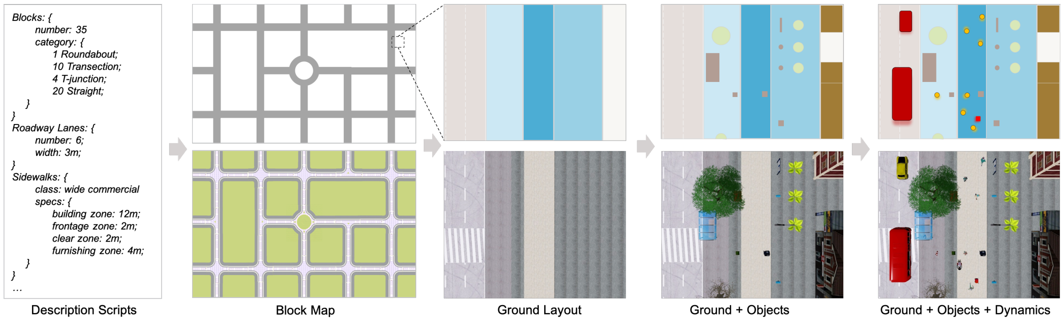
Driving Environments. Platforms for driving environments are mainly designed for autonomous vehicle research and development. Simulators like GTA V (Martinez et al., 2017), Sim4CV (Müller et al., 2018), AIRSIM (Shah et al., 2018), CARLA (Dosovitskiy et al., 2017), and its extension SUMMIT (Cai et al., 2020) offer realistic environments that mimic the physical world's detailed visuals, weather conditions, and day-to-night transitions. Other simulators enhance efficiency and extensibility at the expense of visual realism, such as Udacity (Team, b), DeepDrive (Team, a), Highway-env (Leurent, 2018), and DriverGym (Kothari et al., 2021). MetaDrive (Li et al., 2022b) trades off between visual quality and efficiency, offering a lightweight driving simulator that can support the research of generalizable RL algorithms for vehicles. Although some of the simulators (Martinez et al., 2017; Dosovitskiy et al., 2017) involve traffic participants other than vehicles, such as pedestrians and cyclists, all of them focus on vehicle-centric driving scenarios and neglect the stage for urban micromobility – public urban spaces like sidewalks and plazas.

Social Navigation Environments. Other than indoor and driving environments, social navigation platforms emphasize the social compatibility of robots. Simulators like Crowd-Nav (Chen et al., 2019), Gym-Collision-Avoidance (Everett et al., 2018), and Social-Gym 2.0 (Sprague et al., 2023), model scenes and agents in 2D maps, focusing more on the development of path planning algorithms. Other simulators, such as HuNavSim (Pérez-Higueras et al., 2023), SEAN 2.0 (Tsoi et al., 2022), and SocNavBench (Biswas et al., 2022), upgrade the environment to 3D space and introduce human pedestrians to support the development of more complex algorithms. Social navigation platforms focus on crowd navigation, with oversimplified objects and surrounding environmental structures in the scenes, making them not applicable to complex urban micromobility tasks. In contrast, MetaUrban supports large-scale urban space simulation with real-world scenes (such as street facilities and terrains), providing significantly rich semantics and superior complexity of environments. In addition, MetaUrban supports the cross-machine evaluation of generalizability and safety with different mechanical structures. These features make it a unique choice for urban micromobility.

In summary, none of the recent simulation platforms have been constructed for urban micromobility. The proposed simulator MetaUrban is the first simulator designed for AI-driven urban micromobility research. It differs from previous simulators significantly in terms of complex scenes (with large scales and multifarious terrains), diverse obstacles, vibrant dynamics, different types of mobile machines like delivery bots, electric wheelchairs, and mobility scooters, and intricate interactions simulated. Please refer to Appendix B.6 for a detailed comparison table with existing simulators in the dimensions of scale, sensor, and features, where MetaUrban shows a remarkable superiority. We believe MetaUrban can provide a lot of new research opportunities for AI-driven urban micromobility.

3 METAURBAN SIMULATOR

MetaUrban is a compositional simulation platform that can generate infinite training and evaluation environments for AI-driven urban micromobility. We propose a procedural generation pipeline, as the basis of MetaUrban, for constructing infinite interactive scenes with different specifications. As shown in Figure 3, MetaUrban uses a structured description script to create urban scenes. Based on the script information about street blocks, sidewalks, objects, agents, and more, it starts with the street block map, then plans the ground layout by dividing different function zones, then places static objects, and finally populates dynamic agents.



```
Blocks: {
      number: 35
      category: {
            1 Roundabout;
            10 Transection;
            4 T-junction;
            20 Straight;
   }
}
Roadway Lanes: {
      number: 6;
      width: 3m;
}
Sidewalks: {
      class: wide commercial
      specs: {
            building zone: 12m;
            frontage zone: 2m;
            clear zone: 2m;
            furnishing zone: 4m;
   }
}
...
```

Figure 3: **Procedural generation.** MetaUrban can automatically generate complex urban scenes with its compositional nature. From the second to the fourth column, the top row shows the 2D road maps, and the bottom row shows the bird-eye view of 3D scenes.

This section highlights three key designs in the MetaUrban simulator to support exhibiting three unique characteristics of urban spaces respectively – complex scenes (with large scales and multifarious terrains), diverse obstacles, and vibrant dynamics. Section 3.1 introduces **Hierarchical Layout Generation**, which can infinitely generate diverse layouts with different functional zone divisions, object locations, and terrains that are essential for the *generalizability* to scene diversity of agents.

Section 3.2 introduces **Scalable Obstacle Retrieval**, which harnesses worldwide urban scene data to obtain real-world object distributions in different places, and then builds large-scale, high-quality static objects set with VLM-enabled open-vocabulary searching. It is crucial for further enhancing the *generalizability* to obstacle diversity of agents. Section 3.3 introduces **Cohabitant Populating**, in which we leverage the advancements in digital humans to enrich the appearances, movements, and trajectories of pedestrians and vulnerable road users, as well as incorporate other agents to form a vivid cohabiting environment. It is critical for improving the *safety* of the mobile agents.

3.1 HIERARCHICAL LAYOUT GENERATION

The complexity of scene layouts, *i.e.*, the connection and categories of blocks, the specifications of sidewalks and crosswalks, the placement of objects, as well as the terrains, is crucial for enhancing the generalizability of trained agents maneuvering in public spaces. In the hierarchical layout generation framework, we start with a ground plan that samples categories of street blocks and divides sidewalks into different functional zones. Then, we allocate various objects procedurally conditioned on functional zones. Finally, we implement a terrain generation system to synthesize various ground conditions. With the above procedures, we can get infinite urban scene layouts with different specifications of sizes, object locations, and terrains.

Ground plan. We design 5 typical street block categories, *i.e.*, straight, intersection, roundabout, circle, and T-junction. In the simulator, to form a large map with several blocks, we can sample the category, number, and order of blocks, as well as the number and width of lanes in one block, to get different maps. Then, each block can simulate its own walkable areas – sidewalks and crosswalks, which are key areas for urban spaces with plenty of interactions.

As shown in Figure 4 (Left), according to the Global Street Design Guide (Initiative & of City Transportation Officials, 2016) provided by the Global Designing Cities Initiative, we divide the sidewalk into four functional zones – building zone, frontage zone, clear zone, and furnishing zone. Based on their different combinations of functional zones, we further construct 7 typical templates for sidewalks (Figure 4 (Right)). To form a sidewalk, we can first sample the layout from the templates and then assign proportions for different function zones. For crosswalks, we provide candidates at the start and the end of each roadway, which support specifying the needed crosswalks or sampling them by a density parameter.

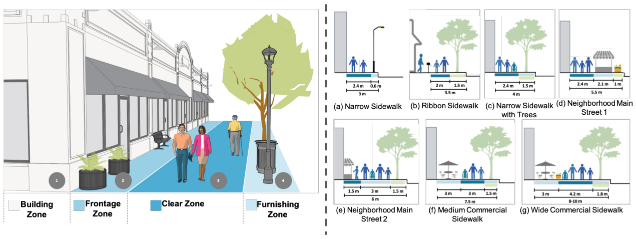


Figure 4: **Ground plan.** (Left) Sidewalk is divided into four functional zones – building, frontage, clear, and furnishing zone. (Right) Seven typical sidewalk templates – from (a) to (g).

Terrain generation. We develop a procedural terrain generator by connecting sampled terrain primitives similar to the method adopted in (Lee et al., 2024), which uses the Wave Function Collapse (WFC) method (Gumin, 2016) to ensure smooth transitions between neighboring terrain primitives. We defined five types of terrain primitives, including slops, steps, stairs, ramps, and rough at different heights. After the mesh was generated, textures with different friction coefficients were added to the terrain to simulate different materials of the ground. This method allows for the generation of a wide variety of terrain combinations, reflecting the complex environments that agents may encounter.

Object placement. After determining the ground layout, we can place objects on the ground. We divide objects into three classes. 1) Standard infrastructure, such as poles, trees, and signs, are placed periodically along the road. 2) Non-standard infrastructure, such as buildings, bonsai, and trash bins, are placed randomly in the designated function zones. 3) Clutter, such as drink cans, bags, and bicycles, are placed randomly across all functional zones. We can get different street styles by specifying an object pool while getting different compactness by specifying a density parameter. Please refer to Appendix B.1 for more details.

3.2 SCALABLE OBSTACLE RETRIEVAL

Hierarchical layout generation decides the scene's layout and *where* to place the objects. However, to make the trained agents generalizable when navigating through scenes composed of various objects in the real world, *what* objects to place is another crucial question. In this section, we propose the Scalable Obstacle Retrieval pipeline, in which we first get real-world object distributions from web data, and then retrieve objects from 3D asset repositories through an open-vocabulary search schema

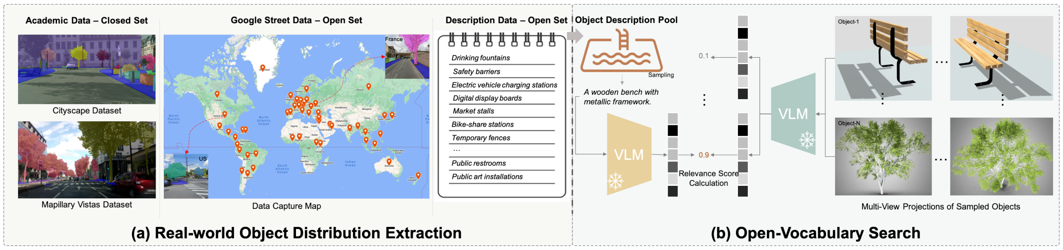

Academic Data – Closed Set
Cityscape Dataset
Mapillary Vistas Dataset
Google Street Data – Open Set
France
US
Data Capture Map
Description Data – Open Set
Drinking fountains
Safety barriers
Electric vehicle charging stations
Digital display boards
Market stalls
Bike-share stations
Temporary fences
…
Public restrooms
Public art installations
Object Description Pool
Sampling
A wooden bench with metallic framework.
VLM
0.1
0.9
Relevance Score Calculation
VLM
Object-1
Object-N
Multi-View Projections of Sampled Objects
(a) Real-world Object Distribution Extraction
(b) Open-Vocabulary Search

Figure 5: **Scalable obstacle retrieval.** (a) Real-world distribution extraction. We get object distribution for urban spaces from three sources: academic datasets, Google Street data, and text description data. (b) Open-vocabulary search. We use the VLM to get image and text embedding, respectively. Then, based on the relevant scores, we can get the objects with high rankings.

based on VLMs. This pipeline is flexible and extensible: the retrieved objects can be scaled to arbitrary sizes as we continue to exploit more web data for scene descriptions and include more 3D assets as the candidate objects.

Real-world object distribution extraction. Urban spaces have unique structures and object distributions, such as the infrastructure built by the urban planning administration and clutters placed by people. Thus, we design a real-world distribution extraction method to get a description pool depicting the frequent objects in urban spaces. As illustrated in Figure 5 (a), we first leverage off-the-shelf academic datasets for scene understanding, CityScape (Cordts et al., 2016) and Mapillary Vistas (Neuhold et al., 2017), to get a list of 90 objects that are with high frequency to be put in the urban space. However, the number of objects is limited because of the closed-set definitions in the image datasets. We introduce two open-set sources to get broader object distribution from the real world. 1) Google Street data. We first collect 25,000 urban space images from 50 countries across six continents. Then, we harness GPT-4o (OpenAI, 2024) and open-set segmentation model Grounded-SAM (Ren et al., 2024) to get 1,075 descriptions of objects in the public urban space. 2) Urban planning description data. We further get a list of 50 essential objects in public urban spaces through a thorough survey of 10 urban design handbooks. Finally, by combining these three data sources, we can get an object description pool with 1,215 items of descriptions that form the real-world object category distribution.

Open-vocabulary search. The recent development of large-scale 3D object repositories (Deitke et al., 2023b; 2024; Wu et al., 2023) enables efficiently constructing a dataset for a specific scene. However, these large repositories have three intrinsic issues to harness these repositories: 1) most of the data is unrelated to the urban scene, 2) the data quality in large repositories is uneven, and 3) the data has no reliable attribute annotations. To this end, we introduce an open-vocabulary search method to tackle these issues. As shown in Figure 5 (b), the whole pipeline is based on an image-text retrieval architecture. We first sample objects from Objaverse (Deitke et al., 2023b) and Objaverse-XL (Deitke et al., 2024) repositories to get projected multi-view images. Here, a naive uniform view sampling will bring low-quality harmful images. Following (Luo et al., 2023; 2024), we select and prioritize informative viewpoints, which significantly enhance retrieval effectiveness. Then, we leverage the encoder of a Vision Language Model BLIP (Li et al., 2022a) to extract features from projected images and sampled descriptions from the object description pool, respectively, to calculate relevant scores. Then, we can get target objects with relevant scores up to a threshold. This method lets us get an urban-specific dataset with 10,000 high-quality obstacles in real-world category distributions.

3.3 COHABITANT POPULATING

Different from indoor household spaces and driving spaces, urban spaces have complex interactions brought by humans and other mobile agents. Thus, the simulation of the dynamics of diverse environmental agents in urban spaces is critical for the ego-agent's safety. So far, we get static urban scenes with hierarchical layout generation and scalable obstacle retrieval. In this section, we will describe how to populate these static urban scenes with varied agents regarding appearances, movements, and trajectories through Cohabitant Populating.

Appearances. Following BEDLAM (Black et al., 2023) and AGORA (Patel et al., 2021), we represent humans as parametric human model SMPL-X (Pavlakos et al., 2019), in which the 3D human body is controlled by a set of parameters for pose θ, shape β, and facial expression ϕ, respectively. Then, built upon SynBody (Yang et al., 2023)'s asset repository, we procedurally generate 1,100 3D rigged human models sampled from 68 garments, 32 hairs, 13 beards, 46 accessories, and 1,038 cloth and

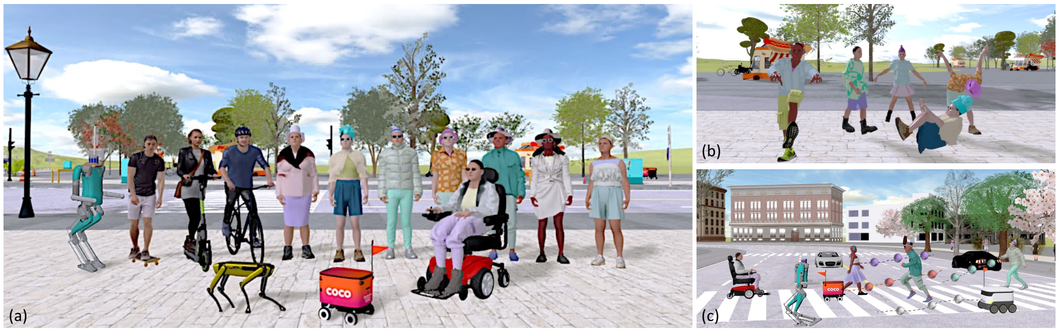


Figure 6: **Cohabitant populating.** (a) Examples of cohabitants in MetaUrban: pedestrians, vulnerable road users like bikers, skateboarders, scooter riders, and mobile machines. (b) Examples of human movements. (c) Examples of trajectories of humans and mobile agents in complex interaction scenarios.

skin textures. Figure 6 (a) shows randomly sampled pedestrians, demonstrating rich diversity in height, build, gender, clothes, hairstyles, and accessories. To form safety-critical scenarios, we also include vulnerable road users like bikers, skateboarders, and scooter riders. For the other agents, we incorporate the 3D assets of COCO Robotics, Starship's and Yandex's delivery bots, Drive Medical's electric wheelchair, Pride Go-Go's mobility scooter, Boston Dynamic's robot dog, and Agility Robotics' humanoid robot. In a simulated environment, depending on different training scenarios, we can randomly sample the target density of agents from the 3D human models, vulnerable road users, and mobile agents.

Movements. We provide two kinds of human movements in the simulator – daily movements and unique movements. Daily movements provide the basic human dynamics in daily life, *i.e.*, idle, walking, and running. Professionals manually produce them to make them in a natural cycle. Special movements are the complicated dynamics that appear randomly in public spaces, such as dancing, singing, and exercising. We harness the BEDLAM dataset (Black et al., 2023) to obtain this movement. Since naively using BEDLAM's movements to MetaUrban will bring axis and mesh offset issues, we manually check and repair all of the 2,311 movements before the integration. In the simulator, for human models, we can specify part of them with daily movements that are switched by their speeds, while the others with unique movements are sampled from special movement clips. Figure 6 (b) shows randomly sampled movements – that are vivid and natural in urban scenes. For legged and wheeled agents, we provide walking and maneuvering movements, respectively.

Trajectories. Realistic trajectories of mobile agents are crucial for urban scene simulation. We simulate the human and other mobile agents' trajectories using the ORCA (Van Den Berg et al., 2011) social forces model. ORCA (Van Den Berg et al., 2011) uses a joint optimization and a centralized controller that guarantees that agents will not collide with each other or any other objects identified as obstacles. However, the ORCA model is non-cooperative, *i.e.* each agent pursues its own goal and does not consider other agents' goals. It will easily lead to deadlocks, especially in confined areas, such as narrow sidewalks and successive roadblocks. Thus, we further integrate the Push and Rotate (P&R) algorithm (De Wilde et al., 2014) into the trajectory planning pipeline – a multi-agent path-finding algorithm that can resolve any potential deadlock by local coordination. With MetaUrban's flexible user interfaces (mouse, keyboard, joystick, and racing wheel), users can also directly apply human-operated trajectories to agents, which supports the demonstration data collection for agent training. Finally, we regulate all agents to make them comply with several *traffic rules*, such as traffic lights and speed limit signs. It can further enhance the social compliance of their trajectories. In the simulator, we can randomly sample the start and end locations within walkable areas for humans and all other agents to get their trajectories. Figure 6 (c) shows planned trajectories, demonstrating natural social movements in bustling urban scenes.

4 EXPERIMENTS

In this section, we first introduce the data and evaluation metrics used in all the experiments. In Section 4.1, we build comprehensive benchmarks for two core tasks in micromobility – point navigation and social navigation, across seven typical baseline models. In Section 4.2, we make evaluations across different mobile machines to delve into the influence of mechanical structures, such as engine force, wheel friction, and wheelbase, on their capability when learning and executing policies. In Section 4.3, we evaluate the generalizability and scaling ability of the MetaUrban platform, and reveal the effects of the density of static objects and dynamic agents.

Data. Based on the MetaUrban simulator, we construct the MetaUrban-12K dataset, including 12,800 interactive urban scenes for training (*MetaUrban-train*) and 1,000 scenes for testing (*MetaUrban-test*). Scenes in this dataset are connected by one to three street blocks covering an average of 20,000m^2 areas. There are an average of 0.03 static objects per m^2 and the average distance of objects is 0.7m. There are 10 dynamic agents per street block, including pedestrians, vulnerable road users, and other agents. The average episode length is 410m. These form significantly challenging scenes for agents to navigate through, which are crowded and have long horizons. We further construct an unseen test set (*MetaUrban-unseen*) with 100 scenes for the unseen evaluations. To enable the fine-tuning experiments, we construct a training set of 1,000 scenes with the same distribution of MetaUrban-unseen, termed *MetaUrban-finetune*. Appendix C provides detailed descriptions, and statistics of the MetaUrban-12K dataset.

Evaluation metrics. The agent is evaluated using the Success Rate (SR) and Success weighted by Path Length (SPL) (Anderson et al., 2018; Batra et al., 2020) metrics, which measure the success and efficiency of the path taken by the agent. For SocialNav, except Success Rate (SR), the Social Navigation Score (SNS) (Deitke et al., 2022a), is also used to evaluate the social complicity of the agent. For both tasks, we further report the Cumulative Cost (CC) (Li et al., 2022b) to evaluate the safety properties of the agent. It records the crash frequency to obstacles or environmental agents.

Table 1: **Benchmarks.** The benchmark of PointNav and SocialNav tasks on the MetaUrban-12K dataset. Seven representative methods of RL, safe RL, offline RL, and IL are evaluated for each benchmark. 🟥 indicate the best performance among online methods (RL and Safe RL). 🟦 indicates the best performance among offline methods (offline RL and IL).

Category	Method	PointNav						SocialNav					
		Test			Unseen			Test			Unseen		
		SR↑	SPL↑	Cost↓	SR↑	SPL↑	Cost↓	SR↑	SNS↑	Cost↓	SR↑	SNS↑	Cost↓
RL	PPO (Schulman et al., 2017)	66%	0.64	0.51	49%	0.45	0.78	34%	0.64	0.66	24%	0.57	0.51
Safe RL	PPO-Lag (Ray et al., 2019)	60%	0.58	0.41	60%	0.57	0.53	17%	0.51	0.33	8%	0.47	0.50
	PPO-ET (Sun et al., 2021)	57%	0.53	0.47	53%	0.49	0.65	5%	0.52	0.26	2%	0.50	0.62
Offline RL	IQL (Kostrikov et al., 2021)	36%	0.33	0.49	30%	0.27	0.63	36%	0.67	0.39	27%	0.62	3.05
	TD3+BC (Fujimoto & Gu, 2021)	29%	0.28	0.77	20%	0.20	1.16	26%	0.61	0.62	32%	0.64	1.53
IL	BC (Bain & Sammut, 1995)	36%	0.28	0.83	32%	0.26	1.15	28%	0.56	1.23	18%	0.54	0.58
	GAIL (Ho & Ermon, 2016)	47%	0.36	1.05	40%	0.32	1.46	34%	0.63	0.71	28%	0.61	0.67

4.1 BENCHMARKS

In micromobility, the core and foundational function for a mobile machine is *to navigate from point A to point B within a bustling urban environment* (Mitchell et al., 2010; Gössling, 2020). In this function, there are two main commands: 1) not collide with static objects (infrastructures and clutters), and 2) not bump into moving objects (pedestrians and other agents). To this end, we design two foundational tasks – point navigation and social navigation, to evaluate the generalizability and safety of recent state-of-the-art embodied AI models on a typical wheeled mobile machine[3], *i.e.*, COCO Robotics Delivery Bot[4].

Tasks. We design two common tasks in urban scenes: Point Navigation (PointNav) and Social Navigation (SocialNav). In PointNav, the agent's goal is to navigate to the target coordinates in static environments without access to a pre-built environment map. In SocialNav, the agent is required to reach a point goal in dynamic environments that contain moving environmental agents. The agent shall avoid collisions or proximity to environmental agents beyond thresholds to avoid penalization (distance <0.2 meters). The agent's action space in the experiments consists of acceleration, brake, and steering. The observations contain a vector denoting the LiDAR signal, a vector summarizing the agent's state, and the navigation information that guides the agent toward the destination.

Models. We build two benchmarks on the MetaUrban-12K dataset for PointNav and SocialNav tasks. We evaluate 7 typical baseline models, across Reinforcement Learning (PPO (Schulman et al., 2017)), Safe Reinforcement Learning (PPO-Lag (Ray et al., 2019), and PPO-ET (Sun et al., 2021)), Offline Reinforcement Learning (IQL (Kostrikov et al., 2021) and TD3+BC (Fujimoto & Gu, 2021)), and Imitation Learning (BC (Bain & Sammut, 1995) and GAIL (Ho & Ermon, 2016)). We train all

---

[3] Note that MetaUrban provides an easy-to-use interface for users to benchmark the navigation ability of any mobile machine they want to evaluate, such as electric wheelchairs, robot dogs, and humanoids. In a navigation-locomotion framework (Lee et al., 2024), an off-the-shelf locomotion module can be fixed for one machine; users can change the navigation module independently to benchmark.

[4] `https://www.cocodelivery.com/`

these baseline models on the MetaUrban-train dataset and then evaluate them on the MetaUrban-test set. We use the demonstration data provided in MetaUrban-12K for offline RL and IL training. We further make unseen evaluations on the MetaUrban-unseen set to demonstrate the generalizability of models trained on the MetaUrban-12K dataset while directly tested on unseen environments. Please refer to Appendix D for details of models, rewards, and hyperparameters.

Results. Table 1 shows the results in the PointNav and SocialNav benchmarks. We can draw two key observations from the results. 1) The tasks are far from being solved. The highest success rates are only 66% and 36% for PointNav and SocialNav tasks achieved by the baselines, indicating the difficulty of these tasks in the urban environments composed by MetaUrban. Note that these benchmarks are built on a medium level of object and dynamic density; increasing the density will further degrade the performances shown in ablation studies. 2) Models trained on MetaUrban-12K have strong generalizability in unseen environments. With unseen evaluation, models can still achieve 41% and 26% success rates on average for PointNav and SocialNav tasks. These results are strong since the models generalize to not only unseen objects and layouts but also unseen dynamics of environmental agents. It demonstrates that the compositional nature of MetaUrban, supporting the coverage of a large spectrum of complex urban scenes, can successfully empower generalization ability to the trained models. In addition, other interesting observations include: SocialNav is much harder than PointNav due to the dynamics of the mobile environmental agents, and Safe RL remarkably improves the safety properties at the expense of effectiveness.

4.2 EVALUATION ACROSS MOBILE MACHINES

Different mobile machines have heterogeneous mechanical structure design specifications, which have a huge influence on their navigation behaviors and capabilities. MetaUrban can evaluate different mechanical structures and designs of mobile machines before deployment and volume production. In this section, we evaluate three typical wheeled mobile machines (a delivery bot, an electric wheelchair, and a mobility scooter) with large design variations, and investigate the relationship between mechanical structures and performance in *policy learning*. Further, we also investigate the influence of mechanical structures in *policy execution* on different terrains, as detailed in Appendix D.3.

Table 2: **Evaluation of policy learning across mobile machines.** 🟥 and ⬜ indicate the best and worst performance among machines in the straight street block ("S"). 🟦 and 🟩 indicate the best and worst performance among machines in the intersection street block ("X").

	COCO (base)		COCO (mod-1)		COCO (mod-2)		Wheelchair		Mobility Scooter	
Max speed v_{max} (km/h)	30		10		30		5		45	
Max steering s_{max} $(degree)$	40		40		10		5		45	
Wheel friction μ	0.7		0.7		0.7		0.7		0.7	
Engine force F (N)	$350 \sim 550$		$350 \sim 550$		$350 \sim 550$		$100 \sim 200$		$450 \sim 650$	
Brake force B (N)	$35 \sim 80$		$35 \sim 80$		$35 \sim 80$		$35 \sim 80$		$35 \sim 80$	
SR↑ (S ∣ X)	47%	41%	🟥 62%	🟦 56%	17%	22%	⬜ 13%	🟩 16%	36%	31%
SPL↑ (S ∣ X)	0.46	0.38	🟥 0.61	🟦 0.53	0.15	0.20	⬜ 0.11	🟩 0.14	0.33	0.28
Cost↓ (S ∣ X)	0.32	1.50	🟥 0.28	🟩 1.64	⬜ 0.35	1.21	0.30	🟦 1.10	0.34	1.43

In this experiment, we evaluate the influence of mechanical structures on policy learning in a static obstacle avoidance task. We follow the setting of PointNav experiments in Section 1. We perform PPO model training on three mobile machines, having significant disparities in max speed, max steering, wheel friction, engine force, and brake force. In addition, apart from comparing different machines, we compare different variations of one machine, *i.e.*, COCO (base), COCO (mod-1), and COCO (mod-2), to find a better specification that can satisfy the demands of different use cases. We evaluate all five models in straight street blocks (mainly static obstacles) and intersection street blocks (dense interactions with pedestrians) to test their behaviors in different scenarios, which are noted as "S" and "X" in Table 2. We use "Success Rate (SR)" and "Success weighted by Path Length (SPL)" to measure the mobility of agents, while use "Cost" to measure the safety of agents.

Results are shown in Table 2. The wheelchair has the most *conservative design*, with the lowest max speed, max steering angle, and engine force. It achieves the best performance in Cost at the intersection scenario, which indicate the conservative design significantly improve its safety when navigating through the crowd. However, it has the worst performance in both straight and intersection street blocks, which indicates conservative design will influence the mobility of a machine to some degree. The mobility scooter has the most *aggressive design*, with the highest max speed, max steering angle, and engine force. It can achieve better performance than the wheelchair and COCO (mod-2). However, it has a larger Cost than the wheelchair and COCO (mod-2) on average, especially in the intersection scenario, which indicates a degradation of safety ability caused by its aggressive design. The COCO (base) and its variations have a *medium design* between the wheelchair and mobility scooter. Comparing COCO (base) with its variations, we can observe that based on the raw design, decreasing its max speed (mod-1) to $10km/h$ can significantly improve its performance in both mobility and safety, while decreasing its max steering angle will diminish its performance remarkably. These results could assist designers in finding improved mechanical structures for mobile machines to meet various application scenarios.

4.3 ABLATION STUDY

In this section, we evaluate the generalizability, scaling ability, and effects of the density of static objects and dynamic agents. For unified evaluations, we use PPO for all ablation studies. Except for the results on dynamic density, we use the PointNav task. Observations and hyperparameters remain the same for model training across different evaluations.

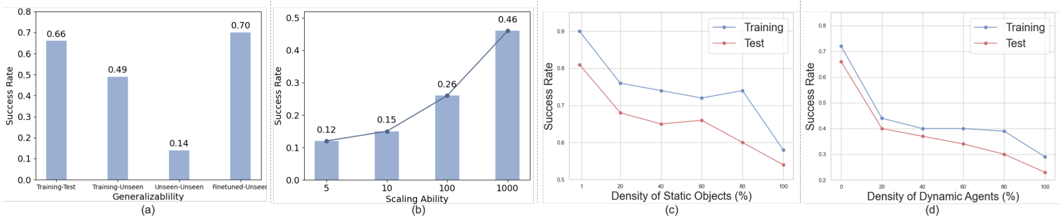


Figure 7: **Ablation study.** (a) Evaluation of generalizability. (b) Evaluation of scaling ability. (c) Evaluation of the density of static objects. (d) Evaluation of the density of dynamic agents.

Evaluation of generalizability. To evaluate the generalizable ability of agents trained on data generated by MetaUrban, we compare the success rate of four settings in Figure 7 (a). Setting-1 and Setting-2 are the results of training on MetaUrban-train while testing on MetaUrban-test and MetaUrban-unseen, respectively. We can observe a performance drop on MetaUrban-unseen. However, the unseen evaluation results still achieve 49% success rate facing various out-of-distribution scenes, demonstrating the strong generalizability of models trained on large-scale data created by MetaUrban. Setting-3 and Setting-4 are the results of direct training on MetaUrban-finetune, and fine-tuning on MetaUrban-finetune from the pre-trained model on MetaUrban-train. Compared between Setting-2 and Setting-3, we can observe an obvious performance drop, which is caused by an underfitting of the insufficient and complex fine-tuning data. Setting-4 outperforms Setting-3 by a large margin, demonstrating that the model trained on the MetaUrban-12K dataset can provide informative priors as good initializations for quick tuning.

Evaluation of scaling ability. To evaluate the scaling ability of MetaUrban's compositional architecture, we train models on a different number of generated scenes, from 5 to 1,000. As shown in Figure 7 (b), the performance improves remarkably from 12% to 46%, as we include more scenes for training, demonstrating the strong scaling ability of MetaUrban. MetaUrban's compositional nature has the potential to extend more diverse scenes with a larger element repository in the future, which could further boost the agent's performance.

Evaluation of static and dynamic density. To evaluate the influence of static object density and dynamic environmental agents, we evaluate the different proportions of them on the PointNav and SocialNav tasks, respectively, from 1% to 100%. As shown in Figure 7 (c) and (d), with the increasing density of both static objects and dynamic agents, the success rates of both train and test experience dramatic degradations, demonstrating the challenges for embodied agents when facing crowded scenes. In our experiments, we observe many interesting failure cases that can indicate promising future directions to improve AI's performance. We showcase some videos on the project page.

5 CONCLUSION

We propose a new compositional simulator, MetaUrban, to facilitate embodied AI and micromobility research in urban scenes. MetaUrban can generate infinite interactive urban environments with complex scenes, diverse obstacles, and diverse movements of pedestrians and other mobile agents. These environments used as training data can significantly improve the generalizability and safety of the embodied AI underlying mobile machines. We commit ourselves to developing the open-source simulator and fostering community efforts to turn it into a sustainable infrastructure.

REFERENCES

Rusul L Abduljabbar, Sohani Liyanage, and Hussein Dia. The role of micro-mobility in shaping sustainable cities: A systematic literature review. *Transportation research part D: transport and environment*, 2021. 2

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 44

Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on robot learning*, 2023. 36

Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 8, 33

Michael Bain and Claude Sammut. A framework for behavioural cloning. In Koichi Furukawa, Donald Michie, and Stephen H. Muggleton (eds.), *MI*, 1995. 8, 34

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 8, 33

Abhijat Biswas, Allan Wang, Gustavo Silvera, Aaron Steinfeld, and Henny Admoni. Socnavbench: A grounded simulation testing framework for evaluating social navigation. *THRI*, 2022. 4, 31

Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *CVPR*, 2023. 6, 7, 30, 42

Panpan Cai, Yiyuan Lee, Yuanfu Luo, and David Hsu. Summit: A simulator for urban driving in massive mixed traffic. In *ICRA*, 2020. 3

Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *ICRA*, 2019. 4

Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *arXiv preprint arXiv:2309.16585*, 2023. 30

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6, 27

Boris De Wilde, Adriaan W Ter Mors, and Cees Witteveen. Push and rotate: a complete multi-agent pathfinding algorithm. *JAIR*, 2014. 7, 30, 32, 35

Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. Robothor: An open simulation-to-real embodied AI platform. In *CVPR*, 2020. 3

Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angel X. Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez-D'Arpino, Kiana Ehsani, Ali Farhadi, Li Fei-Fei, Anthony G. Francis, Chuang Gan, Kristen Grauman, David Hall, Winson Han, Unnat Jain, Aniruddha Kembhavi, Jacob Krantz, Stefan Lee, Chengshu Li, Sagnik Majumder, Oleksandr Maksymets, Roberto Martín-Martín, Roozbeh Mottaghi, Sonia Raychaudhuri, Mike Roberts, Silvio Savarese, Manolis Savva, Mohit Shridhar, Niko Sünderhauf, Andrew Szot, Ben Talbot, Joshua B. Tenenbaum, Jesse Thomason, Alexander Toshev, Joanne Truong, Luca Weihs, and Jiajun Wu. Retrospectives on the embodied ai workshop. *arXiv preprint arXiv:2210.06849*, 2022a. 8, 35

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *NeuIPS*, 2022b. 3, 31, 38, 44

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023a. 30

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023b. 6

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023c. 30

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *NeuIPS*, 2024. 6

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, 2017. 2, 3, 4, 31

Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *CVPR*, 2021. 3

Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pp. 226–231, 1996. 27

Michael Everett, Yu Fan Chen, and Jonathan P. How. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *IROS*, 2018. 4

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *NeuIPS*, 2021. 8, 34

Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, Kuno Kim, Elias Wang, Michael Lingelbach, Aidan Curtis, Kevin T. Feigelis, Daniel Bear, Dan Gutfreund, David D. Cox, Antonio Torralba, James J. DiCarlo, Josh Tenenbaum, Josh H. McDermott, and Dan Yamins. Threedworld: A platform for interactive multi-modal physical simulation. In *NeurIPS Datasets and Benchmarks*, 2021. 2, 3, 31

Timnit Gebru, Jamie H. Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé, and Kate Crawford. Datasheets for datasets. *Comm. of the ACM*, 2021. 42

Patrick Geddes. *Cities in Evolution*. 1949. 1

Mike Goslin and Mark R Mine. The panda3d graphics engine. *Computer*, 2004. 31

Stefan Gössling. Integrating e-scooters in urban transportation: Problems, policies, and the prospect of system change. *Transportation Research Part D: Transport and Environment*, 2020. 8

Maxim Gumin. Wave function collapse algorithm. `https://github.com/mxgmn/`, 2016. 5

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *NeuIPS*, 2016. 8, 34

Global Designing Cities Initiative and National Association of City Transportation Officials. *Global street design guide*. Island Press, 2016. 5, 26

Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. Omnisafe: An infrastructure for accelerating safe reinforcement learning research. *arXiv preprint arXiv:2305.09304*, 2023. 34

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 2023. 30

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Kembhavi Aniruddha, Gupta Abhinav, and Farhadi Ali. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 2, 3, 31, 44

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021. 8, 34

Parth Kothari, Christian Perone, Luca Bergamini, Alexandre Alahi, and Peter Ondruska. Drivergym: Democratising reinforcement learning for autonomous driving. *arXiv preprint arXiv:2111.06889*, 2021. 4

Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. Sumo (simulation of urban mobility)-an open-source traffic simulation. In *MESM*, 2002. 2, 31

Joonho Lee, Marko Bjelonic, Alexander Reske, Lorenz Wellhausen, Takahiro Miki, and Marco Hutter. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Science Robotics*, 2024. 5, 8

Edouard Leurent. An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`, 2018. 4

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Elliott Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, C. Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In *CoRL*, 2021. 3, 31

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, Hang Yin, Michael Lingelbach, Minjune Hwang, Ayano Hiranaka, Sujay Garlanka, Arman Aydin, Sharon Lee, Jiankai Sun, Mona Anvari, Manasi Sharma, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Yunzhu Li, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A human-centered, embodied ai benchmark with 1, 000 everyday activities and realistic simulation. *CoRL*, 2024. 2, 3, 31

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022a. 6

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 30

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *TPAMI*, 2022b. 2, 4, 8, 17, 30, 31, 32, 35

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeuIPS*, 2023a. 44

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023b. 30

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023c. 27

Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *NeuIPS*, 2023. 6, 30

Tiange Luo, Justin Johnson, and Honglak Lee. View selection for 3d captioning via diffusion ranking. *arXiv preprint arXiv:2404.07984*, 2024. 6, 30

Mark Martinez, Chawin Sitawarin, Kevin Finch, Lennart Meincke, Alex Yablonski, and Alain Kornhauser. Beyond grand theft auto v for training, testing and enhancing deep learning in self driving cars. *arXiv preprint arXiv:1712.01397*, 2017. 3, 4

William J Mitchell, Chris E Borroni-Bird, and Lawrence D Burns. *Reinventing the automobile: Personal urban mobility for the 21st century*. MIT press, 2010. 2, 8

Matthias Müller, Vincent Casser, Jean Lahoud, Neil Smith, and Bernard Ghanem. Sim4cv: A photo-realistic simulator for computer vision applications. *IJCV*, 2018. 3

Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 6, 27

Giulia Oeschger, Páraic Carroll, and Brian Caulfield. Micromobility and public transport integration: The current state of knowledge. *Transportation Research Part D: Transport and Environment*, 2020. 2

OpenAI. Gpt-4o. `https://openai.com/index/hello-gpt-4o/`, 2024. 6, 27

OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017. 27

Alexander Osterwalder and Yves Pigneur. *Business model generation: a handbook for visionaries, game changers, and challengers*, volume 1. John Wiley & Sons, 2010. 45

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *UIST*, 2023. 44

Robert Ezra Park, Ernest Watson Burgess, Roderick Duncan McKenzie, and Louis Wirth. *The City*. 1925. 2

Priyanka Patel, Chun-Hao P Huang, Joachim Tesch, David T Hoffmann, Shashank Tripathi, and Michael J Black. Agora: Avatars in geography optimized for regression analysis. In *CVPR*, 2021. 6

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019. 6

Noé Pérez-Higueras, Roberto Otero, Fernando Caballero, and Luis Merino. Hunavsim: A ros 2 human navigation simulator for benchmarking human-aware robot navigation. *arXiv preprint arXiv:2305.01303*, 2023. 4, 31

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 30

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *CVPR*, 2018. 2, 3

Xavier Puig, Tianmin Shu, Joshua B Tenenbaum, and Antonio Torralba. Nopa: Neurally-guided online probabilistic assistance for building socially intelligent home assistants. In *ICRA*, 2023a. 44

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimir Vondrus, Théophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *ICLR*, 2023b. 3, 31, 38

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 2019. 8, 33

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL `https://arxiv.org/abs/1908.10084`. 27

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 6, 27

Manolis Savva, Jitendra Malik, Devi Parikh, Dhruv Batra, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, and Vladlen Koltun. Habitat: A platform for embodied AI research. In *ICCV*, 2019. 2, 3, 44

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 8, 33, 37

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *FSR*, 2018. 3

Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, Tchapmi Micael, Vainio Kent, Wong Josiah, Fei-Fei Li, and Savarese Silvio. igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In *IROS*, 2021. 2, 3

Zayne Sprague, Rohan Chandra, Jarrett Holtz, and Joydeep Biswas. Socialgym 2.0: Simulator for multi-agent social robot navigation in shared human spaces. *arXiv preprint arXiv:2303.05584*, 2023. 4

Hao Sun, Ziping Xu, Meng Fang, Zhenghao Peng, Jiadong Guo, Bo Dai, and Bolei Zhou. Safe exploration by solving early terminated mdp. *arXiv preprint arXiv:2107.04200*, 2021. 8, 33

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John M. Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel X. Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *NeuIPS*, 2021. 3

Deepdrive Team. Deepdrive: a simulator that allows anyone with a pc to push the state-of-the-art in self-driving. `https://github.com/deepdrive/deepdrive`, a. 4

Udacity Team. Udacity's self-driving car simulator: A self-driving car simulator built with unity. `https://github.com/udacity/self-driving-car-sim`, b. 4

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. *RAL*, 2022. 4, 31, 36

Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *ISRR*, 2011. 7, 30, 32, 35

William H Whyte. *City: Rediscovering the center*. University of Pennsylvania Press, 2012. 1

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, 2023. 6

Zhitao Yang, Zhongang Cai, Haiyi Mei, Shuai Liu, Zhaoxi Chen, Weiye Xiao, Yukun Wei, Zhongfei Qing, Chen Wei, Bo Dai, Wayne Wu, Chen Qian, Dahua Lin, Ziwei Liu, and Lei Yang. Synbody: Synthetic dataset with layered human models for 3d human perception and modeling. In *ICCV*, 2023. 6, 24, 42

Appendix In the appendix, we present more details of MetaUrban. In Section A, we illustrate samples of static and dynamic scenes, as well as static and dynamic assets in MetaUrban. In Section B, we elaborate on the technical details of the MetaUrban simulator. In Section C, we will give the construction details and statistics of the MetaUrban-12K dataset. In Section D, we will present details of implementations in our benchmarks and further experiments of the cross-machine evaluation, social interaction evaluation, and cross-sensor evaluation. In Section E, we delve into and validate unique challenges in urban micromobility. In Section F, we provide the datasheet of MetaUrban and MetaUrban-12K dataset. In Section G, and H, we evaluate the performance and robustness of models trained on MetaUrban. In Section I, we discuss the impacts, limitations, real-world deployment support, multi-agent learning support, sustainable ecosystem building, and future work of MetaUrban.

A MetaUrban Visualization **17**
- A.1 Static Scene Samples . . . 17
- A.2 Dynamic Scene Samples . . . 20
- A.3 Static Asset Samples . . . 20
- A.4 Dynamic Asset Samples . . . 24

B MetaUrban Simulator **26**
- B.1 Layout Generation . . . 26
- B.2 Object Retrieval . . . 26
- B.3 Cohabitant Populating . . . 30
- B.4 Scene Customization . . . 30
- B.5 User Interface . . . 30
- B.6 Simulator Comparison . . . 31

C MetaUrban-12K Dataset **31**

D Experiment Details **32**
- D.1 PointNav Experiments . . . 32
- D.2 SocialNav Experiments . . . 35
- D.3 Evaluation across Mobile Machines . . . 36
- D.4 Evaluation on Social Interactions . . . 36
- D.5 Evaluation on Different Sensors . . . 37

E Unique Challenges in Urban Micromobility **37**

F Datasheet **40**

G Performance **43**

H Robustness **43**

I Discussion **43**

A METAURBAN VISUALIZATION

A.1 STATIC SCENE SAMPLES

Street blocks. We design five typical street block categories – straight, curve, intersection, T-junction, and roundabout. In the simulator, to form a large map with several blocks, we can sample the category, number, order, lane number, and other related parameters of the blocks. We use the algorithm Block Incremental Generation (BIG) proposed in MetaDrive (Li et al., 2022b) to generate the target road network defined by users. Figure 8 provides demonstrations of generated street maps composed of different numbers of blocks.

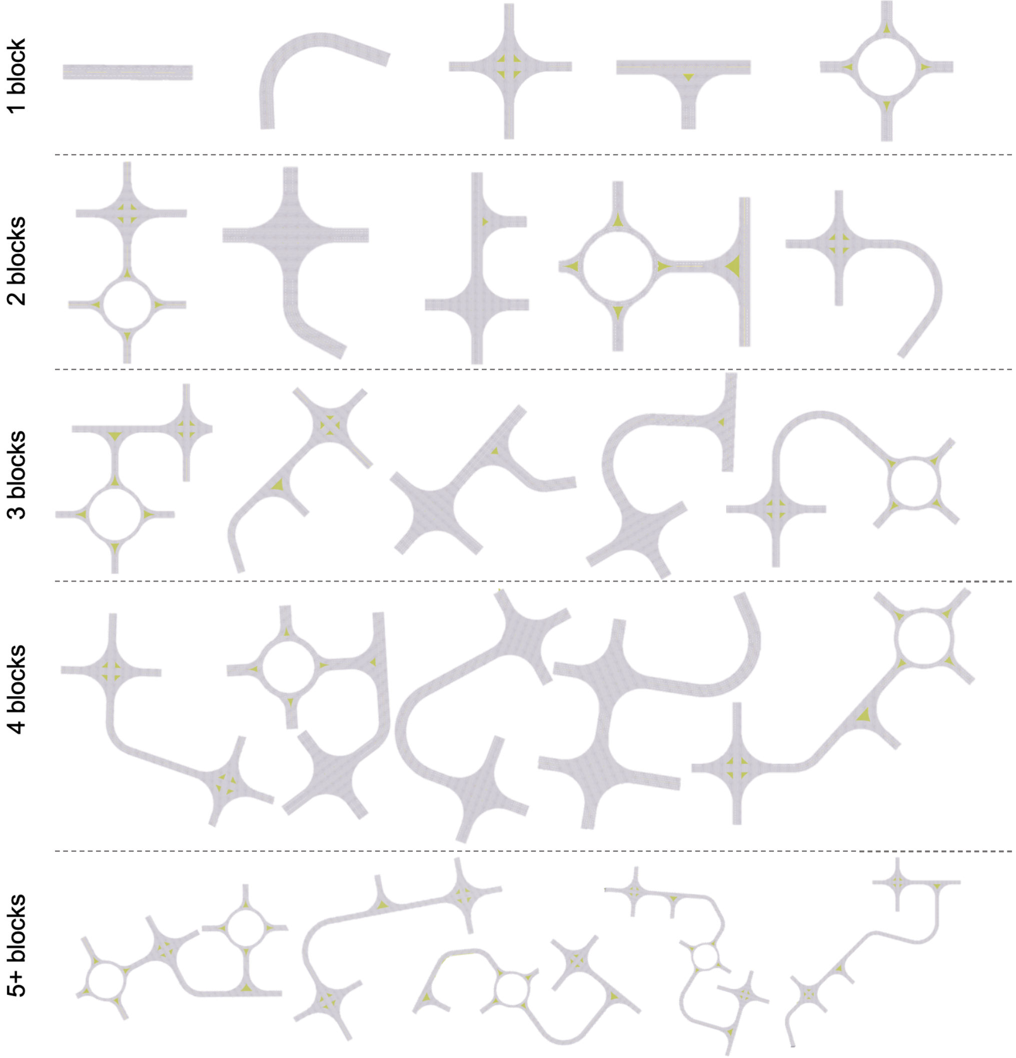


Figure 8: **Examples of block maps.** Generated block maps with a different number of street blocks.

Ground layouts. We construct seven typical templates for sidewalks, more details about the design and the generation process are given in the Section B.1.

As shown in Figure 9, different types of sidewalks can be sampled on the same street block; each type has its unique division and specification of functional zones. Figure 10 further shows several block maps with a different type of sidewalks.

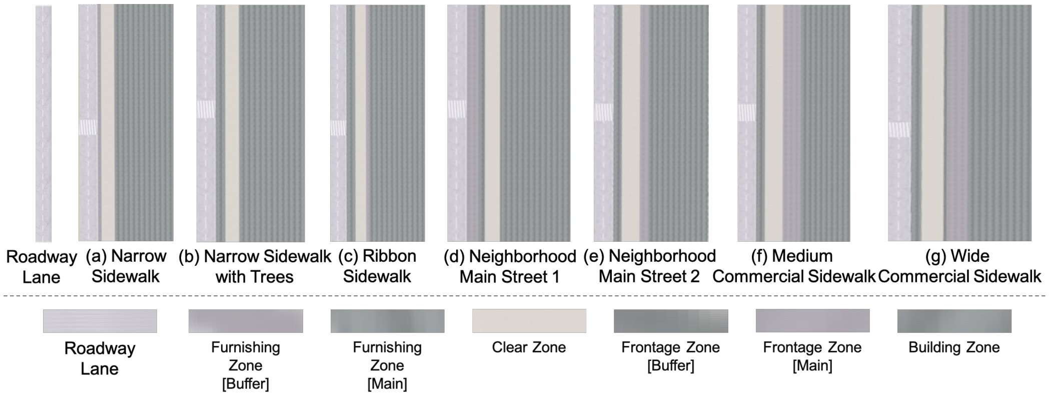


Figure 9: **Examples of sidewalks.** Generated sidewalks with seven templates (a) to (g).

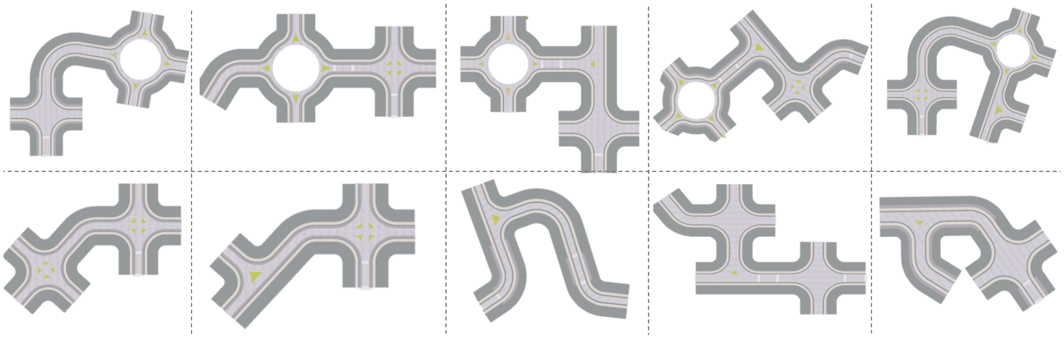


Figure 10: **Examples of block maps with sidewalks.** Generated block maps with a different type of sidewalks.

Static objects. To generate static objects, we build the object placement distribution conditioned on geometric zones of sidewalks, which will be discussed in Section B.1. To better distinguish between the difficulty of scenes on the same road network, we use the object density ρ_s to control the crowding level on the sidewalk. This indicates the ratio of the minimum distance to the default distance between objects. Figure 11 shows block maps with different object densities. We can observe that when the density increases, the walkable region will become more and more crowded. Figure 12 further gives ego-view results by randomly sampling viewpoints on block maps.

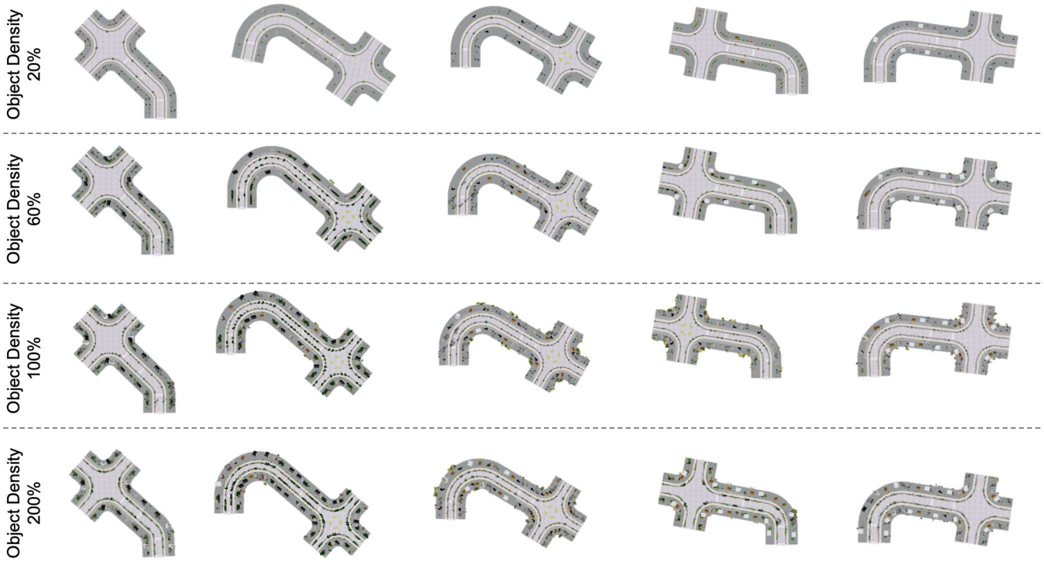


Figure 11: **Examples of block maps with different object densities.** Each row is 5 randomly sampled block maps with one object density, from 20% to 200%.

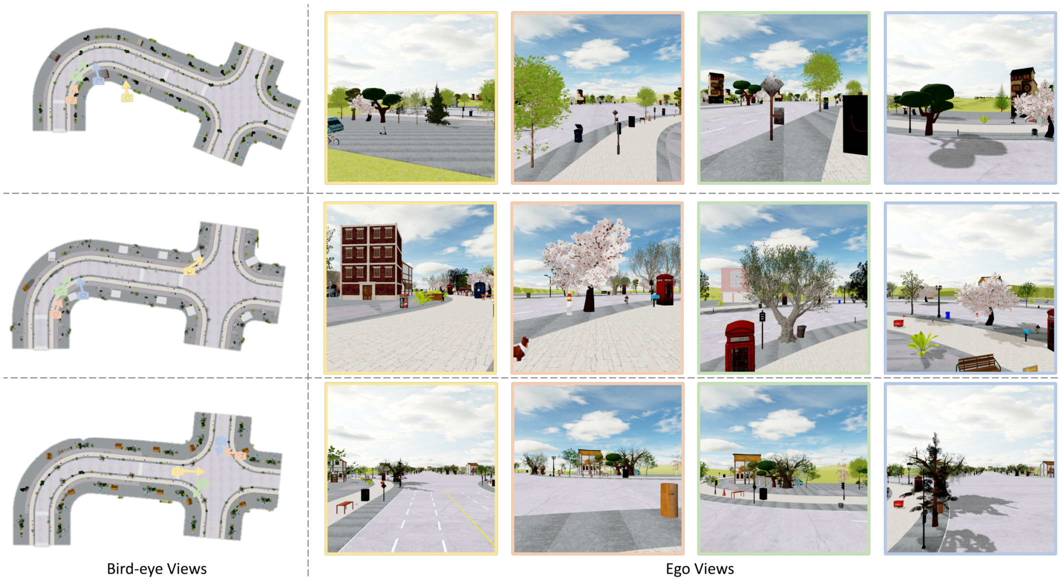


Figure 12: **Examples of ego-view results in static scenes.** Each row is a different object placement with the same object density (60%). For each row, we sample 4 viewpoints to show ego-view results.

A.2 DYNAMIC SCENE SAMPLES

Dynamic agents such as pedestrians, vulnerable road users like bikers (skateboarders, scooter riders), mobile machines (delivery bots, electric wheelchairs, robot dogs, and humanoid robots), and vehicles will be present in the environment. The density of dynamic agents can be controlled with dynamic density ratio ρ_d. Figure 13 shows ego-view results by randomly sampling viewpoints on block maps. The urban spaces are well populated with different agents.

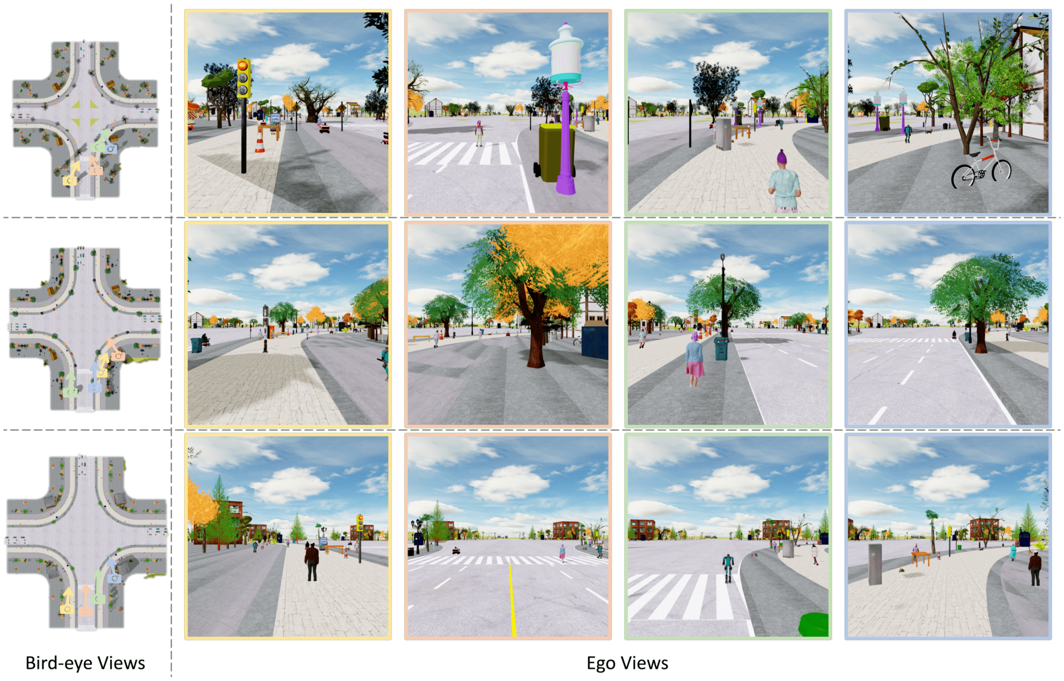


Figure 13: **Examples of ego-view results in dynamic scenes.** Each row is a different specification of dynamics (appearances, movements, and trajectories) with the same dynamic density (100%). For each row, we sample 4 viewpoints to show ego-view results.

A.3 STATIC ASSET SAMPLES

We provide 10,000 high-quality static object assets. The roadside infrastructure is divided into three categories: 1) Standard infrastructure, including poles, trees, and signs, is placed at regular intervals along the road. 2) Non-standard infrastructure, such as buildings, bonsai, and trash bins, is placed randomly within designated zones. 3) Clutter, such as drink cans, bags, and bicycles, is scattered randomly across all functional zones. Figure 14 15 and 16 show examples of these three categories respectively.

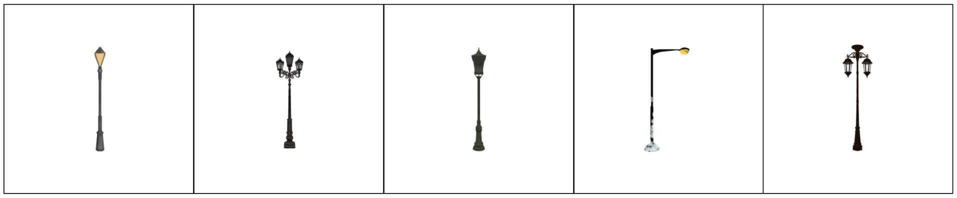
STOP
SPEED LIMIT 40

Traffic Sign

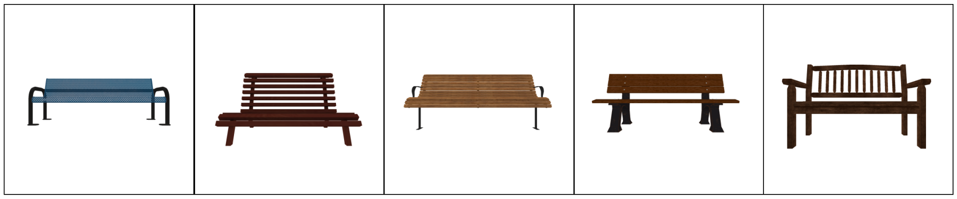
Barrier Pole

Traffic Cone

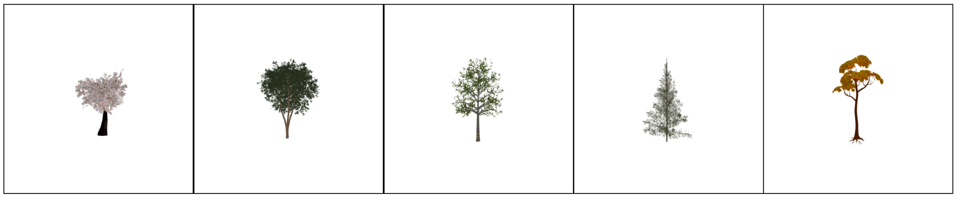
Traffic Light



Lamppost

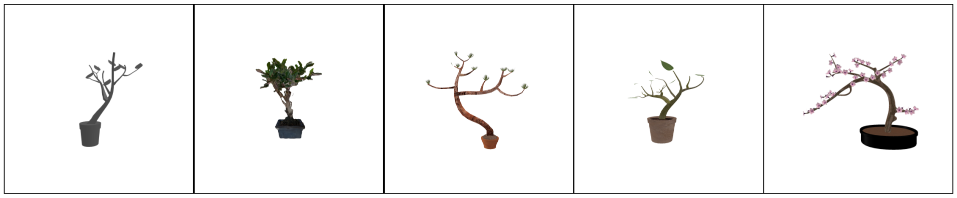


Bench

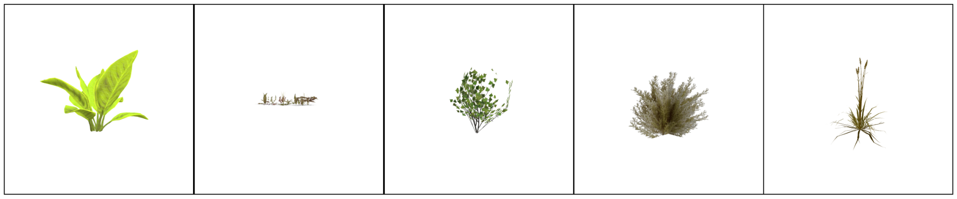


Tree

Figure 14: **Examples of static objects – standard infrastructure.**

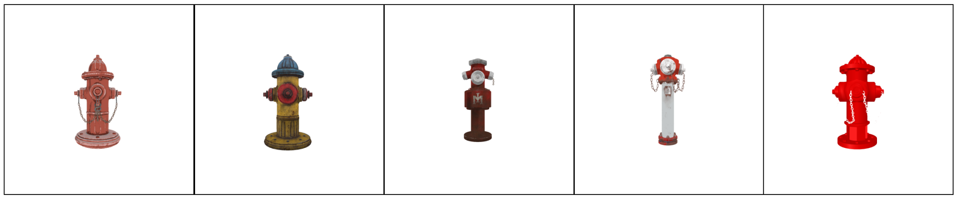


Bonsai



Bush

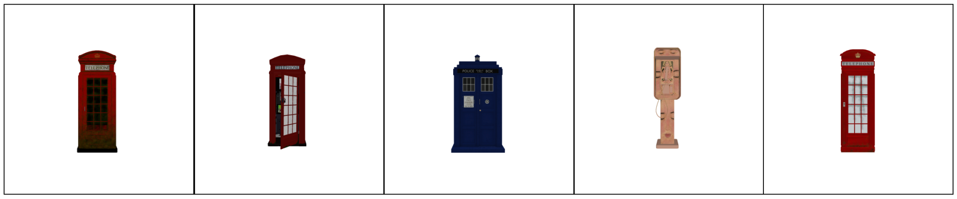
Trash Bin



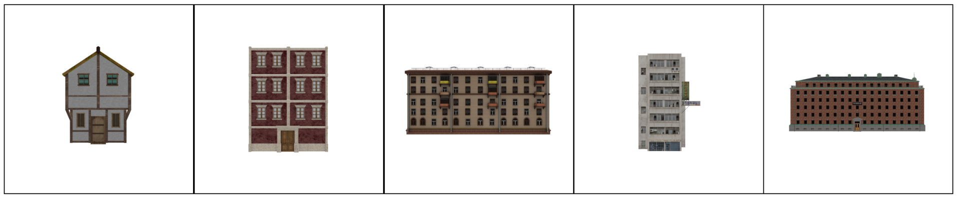
Fire Hydrant

Merry Christmas
PEMEX

Advertising Board



Telephone Booth



Building

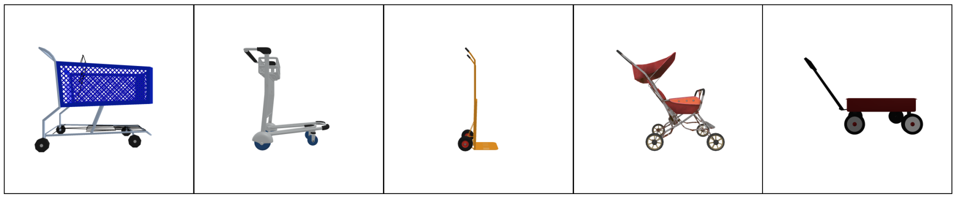
Figure 15: **Examples of static objects – non-standard infrastructure.**

Can

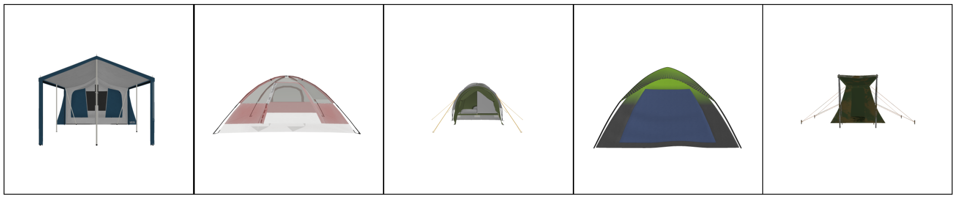
Bag

Bike

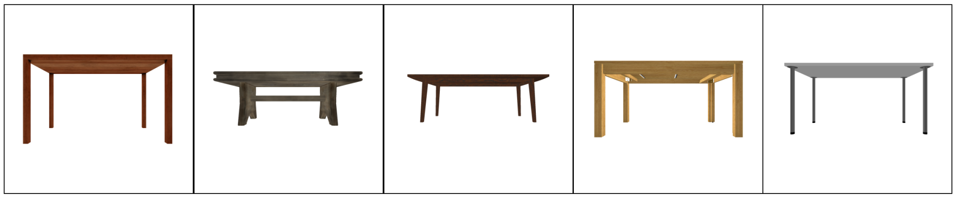
Scooter



Shopping Trolley



Tent



Desk

Figure 16: **Examples of static objects – clutter.**

A.4 DYNAMIC ASSET SAMPLES

Human assets. MetaUrban provides 1,100 rigged 3D human models, sampled from 68 garments, 32 hairs, 13 beards, 46 accessories, and 1,038 cloth and skin textures from SynBody (Yang et al., 2023) dataset. Figure 17 shows 35 randomly sampled humans, demonstrating their diversity with large variations.

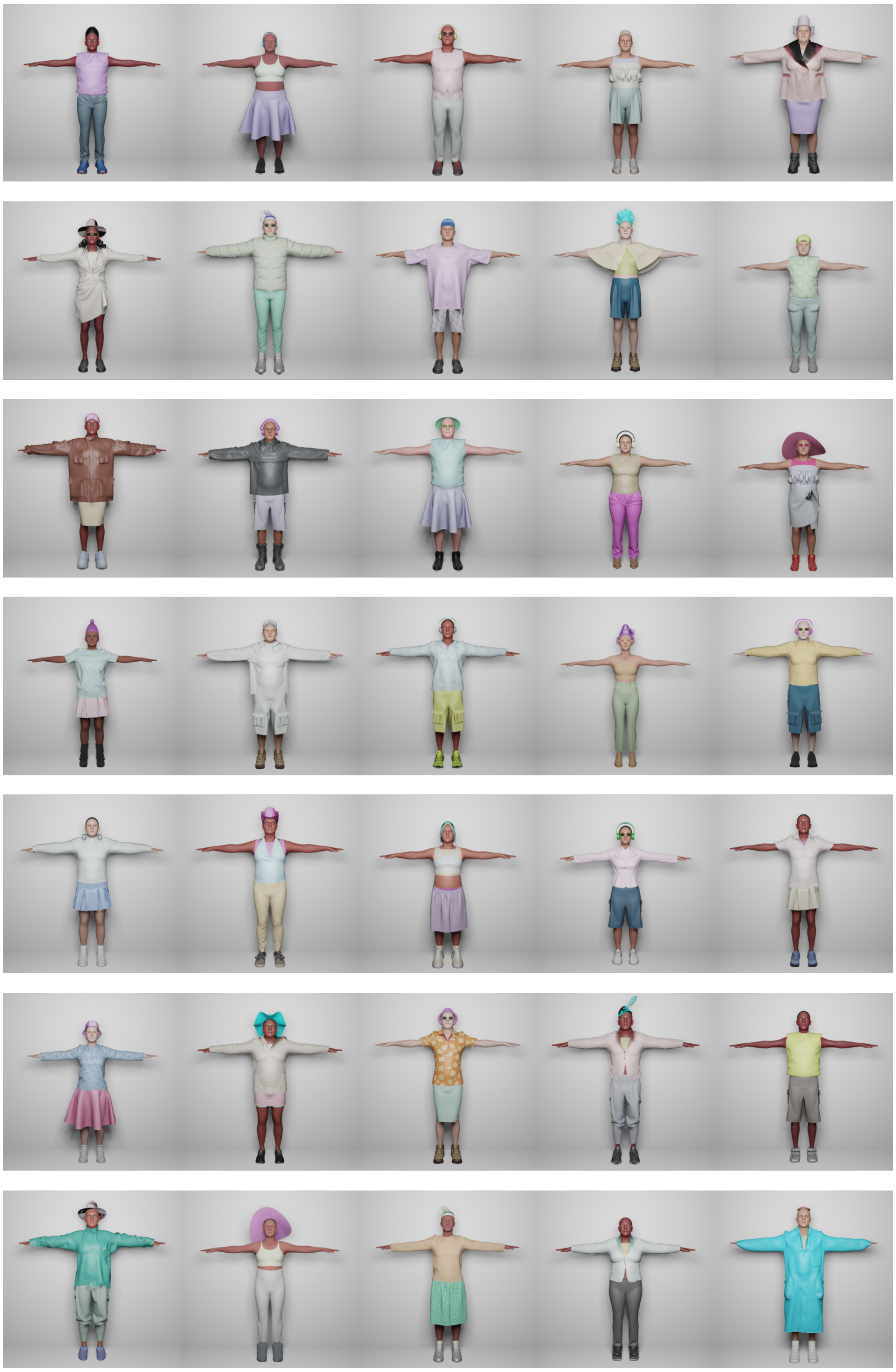


Figure 17: **Examples of dynamics – rigged humans.**

Vulnerable road user assets. MetaUrban provides 5 kinds of vulnerable road users to form safe-critical scenarios. They are bikers, skateboarders, scooter riders, and electric wheelchair users, as shown in the first row of Figure 18. Note that electric wheelchairs, as a human-AI shared control system, can also be seen as mobile machines, not only vulnerable road users. As shown in the second row of Figure 18, we can use different human subjects to obtain various variations for electric wheelchair users.

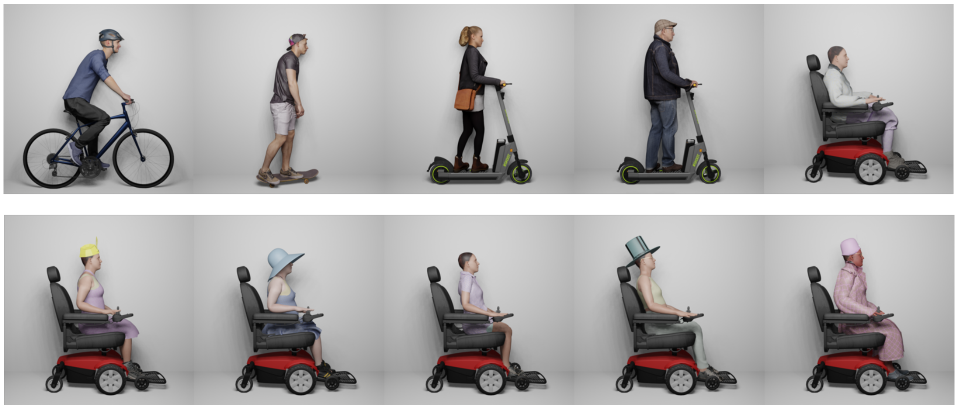


Figure 18: **Examples of dynamics – vulnerable road users.**

Mobile machine assets. MetaUrban provides 6 kinds of mobile machines: Starship, Yandex Rover, and COCO Robotics' delivery bots, Boston Dynamic's robot dog, Agility Robotics' humanoid robot, and Drive Medical's electric wheelchair. Figure 19 shows the first 5 assets, while the electric wheelchair, as a cross-category asset (vulnerable road user and mobile machine), is shown in Figure 18.

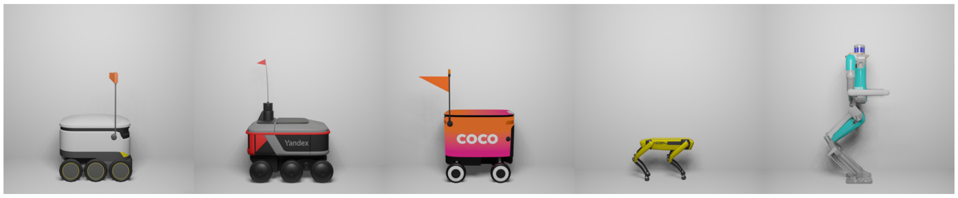


Figure 19: **Examples of dynamics – mobile machines.**

Vehicle assets. MetaUrban provides 37 kinds of vehicles, covering different body types, sizes, and appearances. Figure 20 shows 10 randomly sample vehicles.

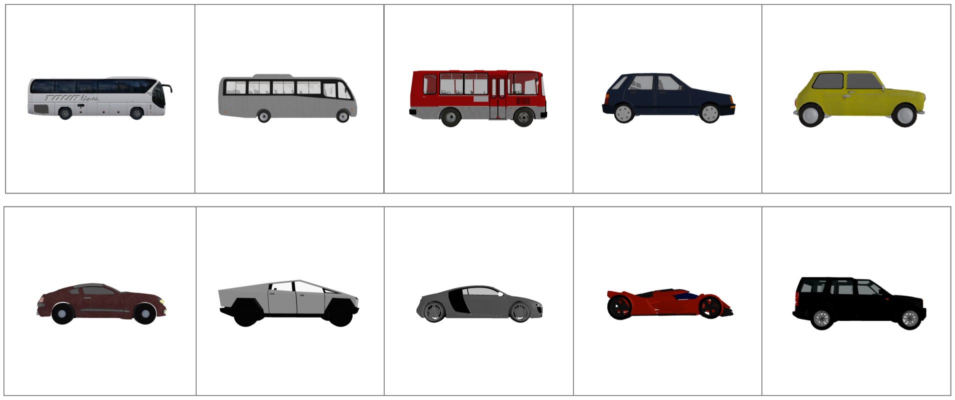


Figure 20: **Examples of dynamics – vehicles.**

B METAURBAN SIMULATOR

B.1 LAYOUT GENERATION

This section gives details about the process we developed to procedurally generate scenes with sidewalks and crosswalks, as well as sample and place static objects on the sidewalk.

Ground plan. As shown in Figure 21 (Top), we define 4 functional zones 🟦 and 6 geometric zones 🟫 for sampling the type of sidewalks and choosing the distribution of parameters for each sidewalk component. As shown in Figure 21 (Bottom), we construct 7 typical templates for sidewalks; each type of them has its unique distribution of geometric zones. To match the distribution with the real world, we set the distribution of the zone width to a uniform distribution for each geometric zone; the maximum and minimum values of the uniform distribution are set according to the Global Street Design Guide (Initiative & of City Transportation Officials, 2016) provided by the Global Designing Cities Initiative.

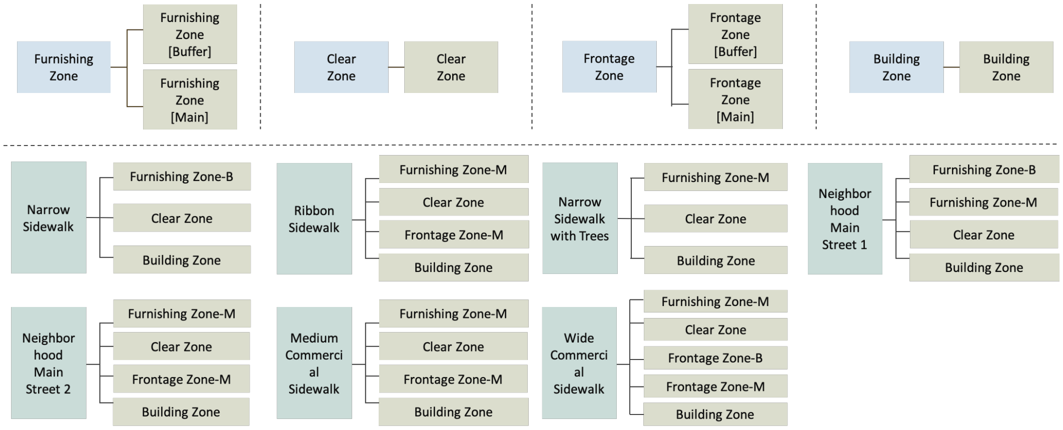


Figure 21: **Architecture of ground layouts.** (Top) The mapping from functional zones to geometric zones. (Bottom) Specifications of geometric zones for 7 sidewalk templates.

To generate a scene, we will first sample the template of the sidewalk z from its distribution $z \sim \mathcal{Z}_{\mathcal{T}}, \quad \mathcal{Z}_{\mathcal{T}} = \{z_1, z_2, ..., z_7\}$, followed by the sampling of widths of each geometric zone $w_i \sim f_w(z, i), \quad \forall i \in \{1, 2, ..., 7\}$, where $f_w(z, i)$ is the width distribution of the ith geometric zone under the sidewalk template z.

Crosswalks are crucial for the connectivity of scenes. MetaUrban provides candidates at the beginning and end of each roadway of a block. Then, locations of the crosswalk can be controlled by a crosswalk density parameter or be specified by users directly.

Object placement. Figure 22 illustrates the iterative process of placing objects in the scene. First, we convert the polygon of the geometric zone of the sidewalk into rectangles. We will place objects on each functional zone or geometric zone independently. At each iteration of generating on the specific zone, we can obtain rectangles that are not occupied. Then we check from the starting region to the ending region for the current retrieved object class. We place it if possible, or we start to place the next class. In the simulator, we use rectangle bounding boxes to represent all objects physically to adopt this object placement method.

B.2 OBJECT RETRIEVAL

Distribution extraction. Distinguished from the recent indoor simulation platform, there are no ready-to-use high-quality asset datasets for urban spaces. Urban spaces have their unique data distribution, such as the infrastructure built by the urban planning administration (“fire hydrants” and “bus stops”) and clutters placed by people (“scooters” and “advertising boards”). Thus, we design a real-world distribution extraction method to get a description pool depicting what objects are frequently shown in urban spaces.

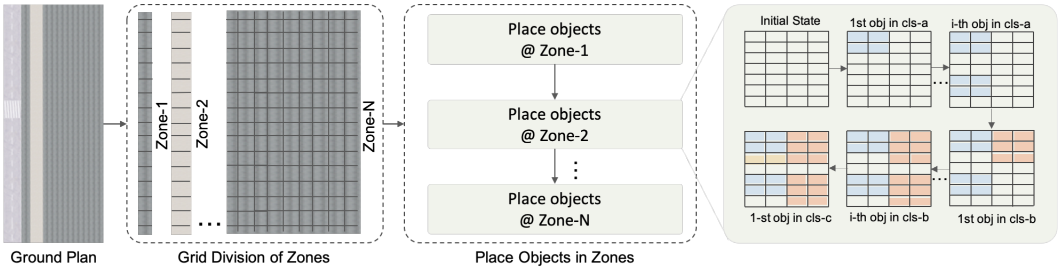


Figure 22: **Iterative object placement.**

We first leverage off-the-shelf scene understanding datasets – Mapillary Vistas (Neuhold et al., 2017) and CityScape (Cordts et al., 2016). Using the provided annotation polygon, we find the overlapping object with the sidewalk and get a list of 90 objects that are with high frequency t1o be put in the urban space (such as “tree” and “bench”). However, the number of objects is limited because of the closed-set definitions in the image datasets. To get broader object distribution from the real world, we introduce two open-set sources – worldwide Google Street data and urban planning description data.

For the Google Street data, we collect 25,000 urban space images from 50 countries across six continents. The selection of image locations was performed by randomly sampling points along the major roads of cities using OpenStreetMap’s (OpenStreetMap contributors, 2017) road network. Image orientation was determined based on road gradient to enhance the relevance of captured scenes. For object detection in these images, we initially employed GPT-4o (OpenAI, 2024) to generate a list of candidate objects. This was followed by the application of Grounded-Dino (Liu et al., 2023c) to obtain bounding boxes for these objects. We refined these boxes using non-maximum suppression (NMS) to ensure the accuracy of object identification.

Further refinement was achieved through the use of the Grounded-SAM model (Ren et al., 2024), an open-set segmentation approach, which filtered the bounding boxes to identify objects specifically located in public urban spaces. A key part of our method involves determining overlaps between identified objects and sidewalks. For each object detected, we calculate its spatial intersection with sidewalk regions derived from the datasets. This overlap analysis helps in curating a list of objects that are relevant to public urban spaces.

To address the diverse descriptions generated by GPT-4o (OpenAI, 2024) and ensure semantic uniformity, we cluster the embeddings of descriptions using DBSCAN (Ester et al., 1996), which result in 1,075 distinct object clusters with unique descriptors, such as "a gray trash bin" and "potted cactus". We use “all-mpnet-base-v2” model from SentenceTransformers (Reimers & Gurevych, 2019) to embed each description.

For the urban planning description data, we get a list of 50 essential objects in public urban spaces (such as “drinking fountains” and “bike racks”) through a thorough survey of urban design handbooks. Finally, by combining these three data sources, we can get an object description pool with 1,215 items of descriptions that can form the real-world object category distribution.

Figure 23 illustrates the distribution of objects in urban space extracted from all of the worldwide collected data. Houses, gates, and trees emerge as universal elements, dominating the urban landscapes across all depicted countries, reflecting their fundamental role in both urban and rural settings.

Figure 24 illustrates the object distribution of example countries from 6 continents, showcasing distinct environmental and cultural characteristics through object prevalence. The data also highlights notable regional distinctions: Japan, for instance, features a higher incidence of poles and road cones, hinting at unique aspects of its urban infrastructure. In contrast, Brazil’s considerable frequency of gates and metal gates suggests prominent architectural and security preferences. Such variances not only reveal the diverse urban aesthetics and functional priorities across different regions but also enhance our understanding of how specific objects can define the character and utility of public spaces globally. This comparative analysis of object distributions contributes significantly to constructing region-specific sidewalks’ simulation environments.

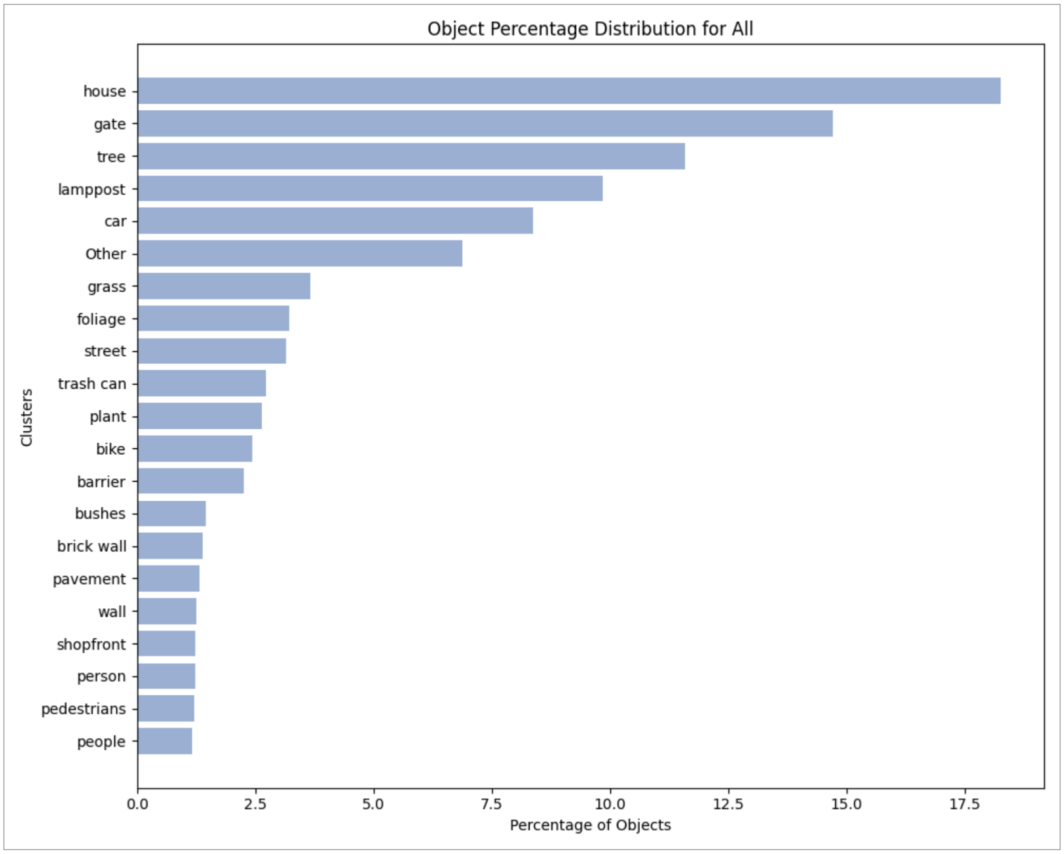
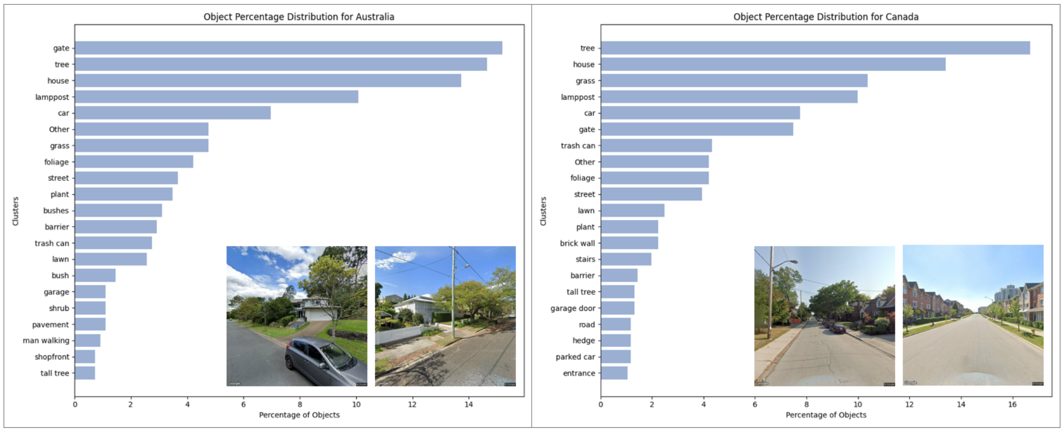
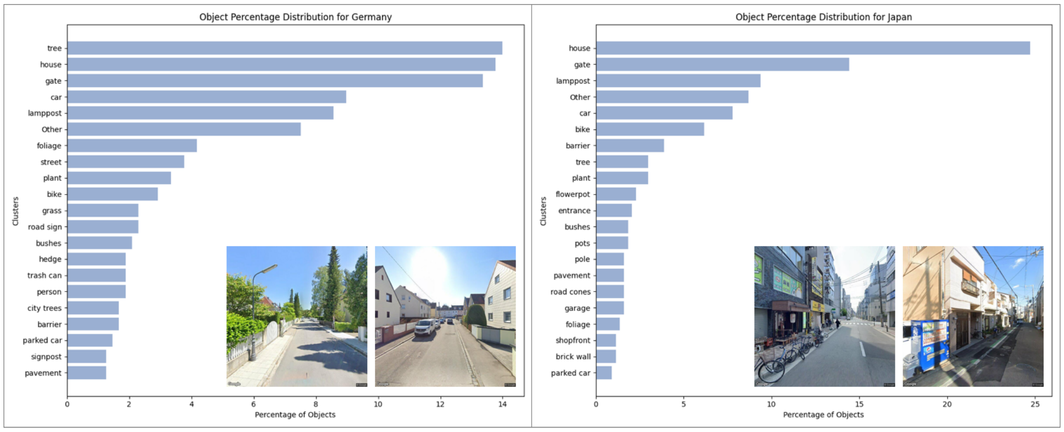


Figure 23: **Distribution of objects in urban spaces for all collected data worldwide.**



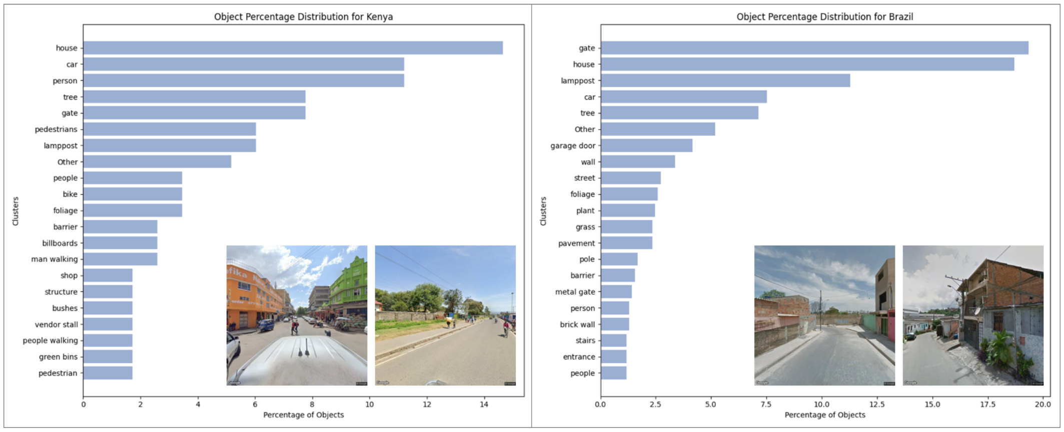
Australia

Canada



Germany

Japan



Kenya

Brazil

Figure 24: **Distribution of objects in urban spaces across different countries.** Two example images are shown together with each distribution figure, demonstrating large variations among different countries.

Open-vocabulary search. To effectively retrieve digital assets corresponding to the object description pool, we developed a robust pipeline utilizing the Objaverse (Deitke et al., 2023c) and ObjaverseXL (Deitke et al., 2023a) repositories, known for their extensive digital assets. The process begins with the extraction of digital assets using a multi-threaded approach for further processing. Each downloaded asset is then rendered into 20 distinct images, capturing various angles to provide a comprehensive visual representation. Following (Luo et al., 2023; 2024), viewpoints with higher quality are used for the calculation of visual feature embedding.

For the matching process, we leverage the BLIP2 (Li et al., 2023) model, a pre-trained feature extractor, to align visual data with our textual descriptions. This involves processing the images to extract visual features and concurrently transforming textual descriptions into embeddings. These embeddings are compared using cosine similarity to determine the semantic correspondence between text and images, allowing us to identify and collect the digital assets that best match the descriptions.

Once the assets are collected, a meticulous review process is initiated for each category. We manually inspect each asset, filtering out those that are of low resolution, lack realism or do not meet our quality standards. The selected assets are then uploaded into MetaUrban to adjust asset characteristics such as size, position, and orientation. This meticulous curation ensures that only high-quality digital assets are incorporated into our static object dataset.

Object repository extension. MetaDrive provides an interface for including objects enabled by recent advances in 3D content generation, such as 3D object reconstruction (Liu et al., 2023b; Kerbl et al., 2023) and generation (Poole et al., 2023; Chen et al., 2023). Thus, one can easily further extend the object repository with generated contents. Also, this function can work together with scene customization (Section B.4) to get customized scenes with specific objects.

B.3 COHABITANT POPULATING

Appearances. We include 1,100 3D human models, 5 kinds of vulnerable road users – bikers, skateboarders, scooter riders, and electric wheelchair users, and 6 kinds of mobile machines as cohabitants in the MetaUrban simulator. The number of dynamic agents in a scene can be set by the parameters respectively. The environment initialization time and RAM usage are only proportional to the number of individual agents. For example, 100 same agents will take the same initialization time and RAM usage as one. This schema can be used to significantly increase the maximum number of spawned agents for a specific hardware.

Movements. We include 3 daily movements – idle, walking, and running, as well as 2,311 unique movements from the BEDLAM (Black et al., 2023) dataset. All of the motion sequences are trimmed and checked by designers one by one to ensure their quality. With the same skeletal binding, all of the unique movements can be transferred to all of the 3D human models directly. Thus, we can get 1,100 $\times$ 2,311 numbers of human-motion pairs.

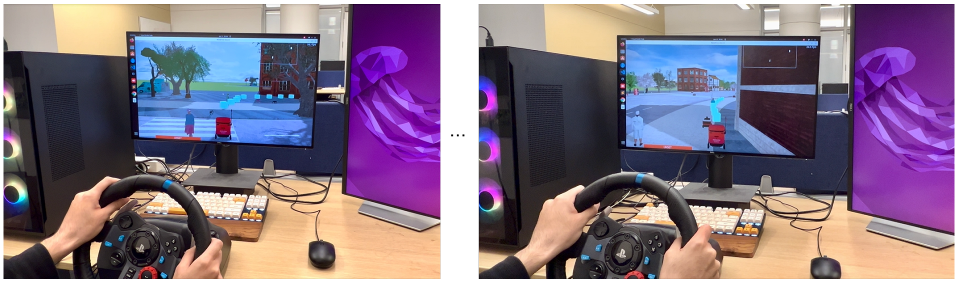
Trajectories. We harness ORCA (Van Den Berg et al., 2011) and Push and Rotate (P&R) algorithm (De Wilde et al., 2014) to get the trajectories of all dynamic agents. First, we build the 0-1 mask that indicates whether the grid is a walkable region or not. Then, we sample the start and ending points for each agent randomly, followed by generating their 2D trajectories by using the model of ORCA (Van Den Berg et al., 2011) and P&R (De Wilde et al., 2014). The trajectory plan process is efficient, running within 5s for 100 agents on a Core i9 CPU processor. Vehicles will also be added in dynamic scenes. All traffic vehicles will follow IDM policies, as MetaDrive (Li et al., 2022b) does.

B.4 SCENE CUSTOMIZATION

MetaUrban supplies various compositional elements, such as street blocks, objects, pedestrians, vulnerable road users, and other mobile agents' appearances and dynamics. With just a few simple lines of specification, it is easy to create customized urban spaces of interest, such as street corners, plazas, and parks.

B.5 USER INTERFACE

MetaUrban provides user interfaces for two purposes: 1) Demonstration data collection for Offline RL and IL. 2) Object labeling and scene customization. For demonstration data collection, MetaUrban provides interfaces for mouse, keyboard, joystick, and racing wheel. We can easily collect human expert demonstrations as shown in Figure 25. In addition, MetaUrban provides tools for object labeling – size, orientation, and attributes, and scene customization – assigning the locations of the selected objects.



Demonstration 1 ... Demonstration N

Figure 25: **Demonstration data collection with the user interface.**

B.6 SIMULATOR COMPARISON

We will compare MetaUrban with other simulators below in Table 3, through the scale, sensor, and feature dimensions. For the scale, MetaUrban can generate infinite scenes with a procedural generation pipeline. It provides the largest number of humans (1,100) and movements (2,314) among all simulation environments. For objects, so far, we have provided 10,000. Compared to other simulators, all of the objects from MetaUrban are urban-specific. Also, we provide an interface to extend object data to any size easily with recent advances in 3D content generation (Section B.2). For the sensor, MetaUrban provides RGBD, semantic, and lidar. For the feature, different from other simulators, MetaUrban provides real-world distribution of the object's categories and uses a more sophisticated path plan algorithm to get the natural agent's trajectories. It also provides flexible user interfaces – mouse, keyboard, joystick, and racing wheel, which vastly ease the collection of human expert demonstration data. MetaUrban uses PyBullet as its physical engine, which is open-source and highly accurate in physics simulation, providing a cost-effective and flexible solution for future developments. MetaUrban uses Panda3D (Goslin & Mine, 2004) for rendering, which is a lightweight, open-source game engine with seamless Python integration, providing a flexible and accessible development environment.

Table 3: **Comparison of Embodied AI simulators.** We compare MetaUrban to simulators specialized for three environments – indoor, driving, and social navigation environments.

	Scale				**Sensor**				**Feature**				
Simulator	# of Scenes	# of Objects	# of Rigged Humans	# of Human Motions	RGBD	Semantic	LiDAR	Acoustic	Object Category Distribution	Env. Agent Trajectory	User Interface	Physics Engine	Scenario
HuNavSim (Pérez-Higueras et al., 2023)	5	✗	5	6	✗	✗	✗	✗	✗	Social Force	✗	Gazebo	Social
SEAN 2.0 (Tsoi et al., 2022)	3	34	<100	1	✓	✗	✗	✗	Manual	Social Force	✗	Unity	Social
SocNavBench (Biswas et al., 2022)	4	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	Social
SUMO (Krajzewicz et al., 2002)	∞	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	Driving
CARLA (Dosovitskiy et al., 2017)	15	66,599	49	1	✓	✓	✓	✓	Manual	Rule-based	Keyboard, Joystick	Unreal4	Driving
MetaDrive (Li et al., 2022b)	∞	5	1	1	✓	✓	✓	✓	Manual	Rule-based	✓	PyBullet	Driving
AI2-THOR (Kolve et al., 2017)	120	609	✗	✗	✓	✓	✗	✓	Manual	✗	Mouse	Unity	Indoor
ThreeDWorld (Gan et al., 2021)	15	200	✗	✗	✓	✓	✗	✓	Manual	✗	VR	Flex	Indoor
iGibson 2.0 (Li et al., 2021)	15	1,217	✗	✗	✓	✓	✗	✗	Manual	✗	Mouse, VR	PyBullet	Indoor
ProcTHOR (Deitke et al., 2022b)	∞	1,547	✗	✗	✓	✓	✗	✓	Manual	✗	✗	Unity	Indoor
OmniGibson (Li et al., 2024)	306	5,215	✗	✗	✓	✓	✓	✗	Manual	✗	✗	PhysX	Indoor
Habitat 3.0 (Puig et al., 2023b)	211	18,656	12	3	✓	✓	✗	✗	Manual	Rule-based	Mouse, Keyboard, VR	Bullet	Indoor
MetaUrban	∞	**10,000**	**1,100**	**2,314**	✓	✓	✓	✗	**Real-world**	**ORCA +P&R**	**Mouse, Keyboard Joystick, Racing Wheel**	**PyBullet**	**Urban**

C METAURBAN-12K DATASET

Data. Based on the MetaUrban simulator, we construct the MetaUrban-12K dataset, including 12,800 interactive urban scenes for training (MetaUrban-train) and 1,000 scenes for testing (MetaUrban-test). For the train and test sets, we sample randomly from the 6 templates (a-f) of sidewalks shown in Figure 4 (right) with the same distributions of objects and dynamics. We further construct an unseen test set (MetaUrban-unseen) with 100 scenes for zero-shot experiments, in which we sample from the unseen template (g) – Wide Commercial Sidewalk, unseen objects, trajectories of agents with further designers' manual adjustments according to real-world scenes. In addition, to enable the fine-tuning experiments, we construct a training set of 1,000 scenes with the same distribution of MetaUrban-unseen, termed MetaUrban-finetune. 12K scenes can be generated in 12 hours on a local workstation. Notably, our MetaUrban platform can easily extend the scale of urban scenes from a multi-block level to a whole city level. To enable the Offline RL and IL training, we collect expert demonstration data from a well-trained RL agent and human operators, forming 30,000 steps of high-quality demonstration data. The success rate of the demonstration data is 60%, which can be taken as a reference for the experiments of Offline RL and IL.

Statistics. Figure 26 shows distributions of the number of objects (left), areas occupied by objects (middle), and episode length (right). As shown in the distribution of object numbers, there are lots of objects in each scenario with a minimal value of 300. As shown in the distribution of objects' areas, objects in the dataset cover large areas, which complies with a normal distribution centered at $5250m^2$. As shown in the distribution of episode length, more than 20% of them are more than 800 steps. From these distributions, we can observe that scenes are significantly challenging in MetaUrban-12K for agents to navigate through, which are crowded and with long horizons.

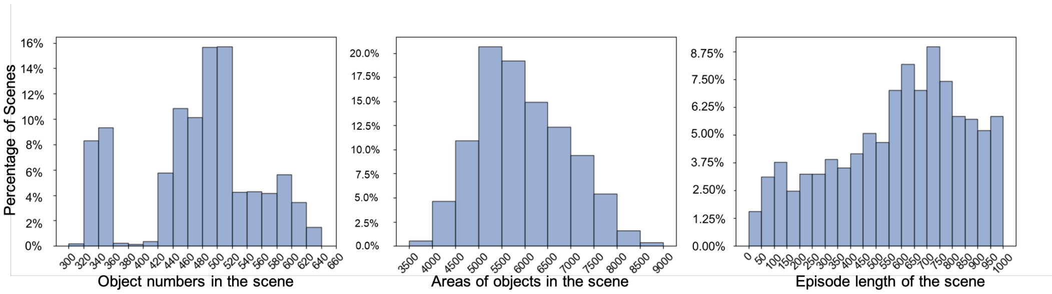


Figure 26: **MetaUrban-12K statistics.** (Left) Distribution of object numbers in the scene. (Middle) Distribution of areas occupied by objects in the scene. (Right) Distribution of episode length in the scene.

D EXPERIMENT DETAILS

This section discusses the settings of environments, action spaces, observation spaces, evaluation metrics, training details for methods, as well as the reward and cost in the benchmarks of Point Navigation (PointNav) and Social Navigation (SocialNav), respectively.

D.1 POINTNAV EXPERIMENTS

Environments. For PointNav experiments, there are only static objects besides the ego agent in the environment. To evaluate the trained policy, we split seven types of sidewalks into six types for training and validation with one for test. The one used for the test is the Wide Commercial Sidewalk, in which the frontage zone buffer will be, as well as some unseen objects.

We use delivery bots as the ego agent in our experiments. The task of agents in PointNav experiments is following the trajectory in the environment that navigates from start points to ending points, ensuring that it does not collide with other objects. To generate such a task, we harness ORCA (Van Den Berg et al., 2011) and Push and Rotate (P&R) algorithm (De Wilde et al., 2014) to get the trajectory of the ego agent after placing objects. The process is the same as discussed in Section B.3. Notably, there may be some trajectories with small moving distances, we set a threshold of $5m$ to filter out scenarios with small moving distances for testing to evaluate different methods more effectively.

Action spaces. We use the same continuous action space as MetaDrive (Li et al., 2022b), which is a 2-dimensional vector that normalized to $[-1.0, 1.0]$ indicating the acceleration and steering rate of the agent. Considering that the dynamics of a delivery bot is different from a vehicle, we change some core parameters like maximum velocity, maximum acceleration, maximum steering rate.

Observation spaces. Multi-modal observations are provided by MetaUrban, including RGB, Depth, Semantic Map, and LidAR. We use LidAR in all of our experiments for its 3D information of the surrounding environment, which provides distance and direction of the nearest object within a $50m$ maximum detecting distance centering at the ego.

Evaluation metrics. For PointNav, an episode is considered successful if the agent issues the DONE action, defined as completing 95% of the set route within 1,000 maximum steps. The agent is evaluated using the Success Rate (SR) and Success weighted by Path Length (SPL) (Anderson et al., 2018; Batra et al., 2020) metrics, which measure the effectiveness and efficiency of the path taken by the agent. Additionally, to measure the safety performance of the trained policy, we define the cost function by two events, *i.e.*, crashing with objects on the sidewalk or buildings in the building zone. $+1$ cost is given once those events occur.

Methods. In our study, we employ a diverse set of 7 baseline models to establish comprehensive benchmarks on MetaUrban. These models span various domains, including Reinforcement Learning, Safe Reinforcement Learning, Offline Reinforcement Learning, and Imitation Learning.

Reinforcement learning. In the realm of Reinforcement Learning, we use the Proximal Policy Optimization (PPO) (Schulman et al., 2017) for evaluation. PPO is a widely adopted and effective method that strikes a balance between sample complexity and ease of tuning, and it is easy to scale as it adopts parallel and distributed training well. The agent in this setting is trained to maximize the reward, which we carefully design to encapsulate the desired behavior of the agent in the MetaUrban environment. The specifics of the reward structure will be discussed in the subsequent paragraph. We train the PPO using the same set of hyperparameters with 128 parallel environments, which occupy 128 processes. The total training time is 12 hours, and 5M environment steps for PointNav on a single Nvidia A5000 GPU. The detailed hyperparameters are provided in Table 4.

Table 4: Hyper-parameters of RL and SafeRL for PointNav.

PPO/PPO-Lag/PPO-ET Hyper-parameters	**Value**
Environmental horizon T	1,000
Learning rate	5e-5
Discount factor γ	0.99
GAE parameter λ	0.95
Clip parameter ϵ	0.2
Train batch size	25,600
SGD minibatch size	256
Value loss coefficient	1.0
Entropy loss coefficient	0.0
Cost limit	1

Safe reinforcement learning. As driving in urban spaces is a safety-critical application, it is important to evaluate Safe Reinforcement Learning (SafeRL) algorithms. In the domain of SafeRL, we utilize two approaches: PPO with a Lagrangian constraint (PPO-Lag) (Ray et al., 2019) and PPO with modeling of Early Terminated Markov Decision Processes (PPO-ET) (Sun et al., 2021). Both methods aim to ensure that the learned policies adhere to specific safety constraints while optimizing the reward. PPO-Lag incorporates a Lagrangian term into the objective function to enforce the constraints, while PPO-ET changes the modeling of the Constrained Markov Decision Process (CMDP) to a new unconstrained MDP, the optimal policy that coincidences with the original CMDP.

For PPO-Lag (Ray et al., 2019), it considers the learning objectivate as Equation 1 rather than adding negative cost as rewards.

$$\max_{\theta} \min_{\lambda \geq 0} E_{\tau}[R_{\theta}(\tau) - \lambda(C_{\theta}(\tau) - d)] \tag{1}$$

where R_θ, C_θ, θ, and d are episodic reward, episodic cost, parameters of the policy, and given cost threshold, respectively.

The rule for PPO-ET (Sun et al., 2021) is to stop when the constraint cost exceeds a given value, which can be easily implemented in practice.

We implement both of these SafeRL methods based on OmniSafe (Ji et al., 2023). We train both of them with 50 parallel environments and the training takes 12 hours for PointNav on a single Nvidia A5000 GPU. The detailed hyperparameters are provided in Table 4.

Offline reinforcement learning. For Offline Reinforcement Learning, we employ two prominent methods: Implicit Q-Learning (IQL) (Kostrikov et al., 2021) and Twin Delayed Deep Deterministic Policy Gradient with Behavior Cloning (TD3+BC) (Fujimoto & Gu, 2021). We create the dataset for PointNav by combining 20% human demonstrations with 80% demonstrations from a well-trained PPO policy, consisting of 30,000 samples with approximately 60% success rate. The training is purely offline and takes around 2 hours on a single Nvidia A5000 GPU for 100 epochs. The detailed hyperparameters for IQL and TD3+BC are provided in Table 5 and 6, respectively.

Table 5: Hyper-parameters of IQL.

IQL Hyper-parameters	**Value**
Learning rate	1e-4
Discount factor γ	0.99
Target critic update ratio	5e-3
Inverse temperature β	3.0
Log std range	(-5.0, 2.0)
Expectile	0.7

Table 6: Hyper-parameters of TD3+BC.

TD3+BC Hyper-parameters	**Value**
Learning rate	1e-4
Discount factor γ	0.99
Target critic update ratio	5e-3
Actor update delay	2
BC loss coefficient	2.5

Imitation learning. For Imitation Learning algorithms, we use the same high-quality mixed demonstration used in Offline Reinforcement Learning. In the Imitation Learning setting, the agent learns to mimic the behavior shown in the expert demonstration, and it is differentiated from Offline Reinforcement Learning in the sense that the agent does not have access to the rewards. We employ two well-established methods: Behavior Cloning (BC) (Bain & Sammut, 1995) and Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016). BC is a straightforward approach that trains the agent to directly match the actions of the expert given the observed states. GAIL, on the other hand, formulates the imitation learning problem as a two-player game between the agent and a discriminator, which tries to distinguish between the agent's behavior and the expert's demonstrations. The detailed hyperparameters for IQL and TD3+BC are provided in table 7 and 8, respectively.

Table 7: Hyper-parameters of BC.

BC Hyper-parameters	**Value**
Dataset size	30,000
Learning rate	1e-4
SGD batch size	64
SGD epoch	40

Reward and cost. The reward function is composed as follows:

$$R = R_{term} + c_1 R_{disp} + c_2 R_{lateral} + c_3 R_{steering} + c_4 R_{crash} \tag{2}$$

Specifically,

Table 8: Hyper-parameters of GAIL.

GAIL Hyper-parameters	**Value**
Dataset size	30,000
SGD batch size	64
Sample batch size	12,800
Generator Learning rate	1e-4
Discriminator Learning rate	3e-3
Generator optimization epoch	5
Discriminator optimization epoch	2,000
Clip parameter ϵ	0.2

- Terminal reward R_{term}: a sparse reward set to $+5$ if the vehicle reaches the destination, and -5 for out of route. If given $R_{term} \neq 0$ at any time step t, the episode will be terminated at t immediately.
- Displacement reward R_{disp}: a dense reward defined as $R_{disp} = d_t - d_{t-1}$, wherein the d_t and d_1 denote the longitudinal position of the ego agent in Frenet coordinates of current lane at time t and $t-1$, respectively. We set the weight of R_{disp} as $c_1 = 0.5$.
- Lateral reward $R_{lateral}$: a dense reward defined as $R_{lateral} = -||l_t||$, wherein the l_t denotes the lateral offset of the ego agent in Frenet coordinates of current lane at time t, which is designed to prevent agent driving on non walkable areas. We set the weight of $R_{lateral}$ as $c_2 = 1.0$.
- Steering smoothness reward $R_{steering}$: a dense reward defined as $R_{steering} = -||s_t - s_{t-1}|| \cdot v_t$, wherein the s_t and s_{t-1} denotes the steering of the agent at t and $t-1$, respectively. And v_t denotes the speed of the agent at time t. This reward term is designed as a regularization to prevent the agent changing the steering too frequently. We set the weight of $R_{steering}$ as $c_3 = 0.1$.
- Crash reward R_{crash}: a dense negative reward defined as $-1(c_t)$, wherein the c_t denotes the collision between agents and any other objects at time t and $1(\cdot)$ is the indicator function. It's notable we do not use the termination strategy for collision as in MetaDrive (Li et al., 2022b). We set the weight of R_{crash} as $c_4 = 1.0$.

And for benchmarking Safe RL algorithms, collision to any objects raises a cost $+1$ at each time step.

D.2 SOCIALNAV EXPERIMENTS

Environments. For SocialNav experiments, most settings are the same as the ones in PointNav. The most important difference is that dynamic agents will also be present in the environment. The trajectories of environmental agents are generated together by using the model of ORCA (Van Den Berg et al., 2011) with P&R (De Wilde et al., 2014). Since vehicles are inherited from MetaDrive (Li et al., 2022b), we use the same parameter to control its density, *i.e.*, traffic density 0.05 in our experiments.

Evaluation metrics. For SocialNav, an episode is considered successful if the agent issues the DONE action, defined as completing 95% of the set route within 1,000 maximum steps. The agent is evaluated using the Success Rate (SR) and Social Navigation Score (SNS) (Deitke et al., 2022a), which is the average of Success weighted by Time Length (STL) and Personal Space Compliance (PSC). SNS measures the agent in terms of safety and efficiency.

Methods. We benchmark the same methods as in PointNav experiments with the same hyperparameters. However, due to the involvement of lots of dynamic agents, the training speed of SocialNav is about approximately 1/3 of PointNav on online methods. The cost scheme is defined as raising a cost of $+1$ at each time step if the ego agent crashes with any agents, vehicles, or objects.

D.3 EVALUATION ACROSS MOBILE MACHINES

In this experiment, we evaluate the influence of mechanical structures in *policy execution* on different terrains. We conduct experiments on three types of wheeled mobile machines – a delivery bot, an electric wheelchair, and a mobility scooter, with remarkably different specifications, such as wheelbase, wheel radius, and wheel width. We designed three sufficiently long runways with three kinds of terrains: slopes, stairs, and roughs. From the starting point, each runway has a gradually increasing difficulty. Slopes will have an increasingly steeper angle; stairs will have increasingly higher step heights; roughs will have increasingly larger bumps. We apply the same "moving forward" policy to each mobile machine to test the longest distance and time duration they can travel before termination, and report the metrics of x-displacement (m) and Time to fall (s) (Agarwal et al., 2023) respectively. Terminal conditions are getting stuck, slowing down significantly, and toppling over.

Results are shown in Table 9. The mobility scooter, which has the largest wheelbase, wheel width, and radius, achieves the best performance in the slopes test. It indicates that a larger wheelbase increases stability and reduces the risk of tipping backward on steep inclines, while wheel width and radius help in better traction on slopes. All three machines show similar but poor performance in the stairs test. It indicates the inherent defect of wheeled mobile machines and emphasizes the importance of accessibility in public urban spaces. The delivery bot, which has the smallest wheelbase, wheel width, and radius, achieves poor performance on all three tests. It indicates that although the delivery bot's structure gives it good maneuverability on flat surfaces, it comes at the cost of losing stability on complex terrains.

Table 9: **Evaluation of policy execution across mobile machines.** For each row of different terrains, 🟥 indicates the best performance among the three machines.

Terrain	**x-displacement (m) ↑**			**Time to fall (s) ↑**		
	Delivery Bot	**Wheelchair**	**Mobility Scooter**	**Delivery Bot**	**Wheelchair**	**Mobility Scooter**
Wheelbase (m)	0.45	0.5	0.6	0.45	0.5	0.6
Wheel radius (m)	0.1	0.15	0.2	0.1	0.15	0.2
Wheel width (m)	0.1	0.1	0.15	0.1	0.1	0.15
Slopes	31.90	34.58	38.07	6.3	6.9	7.7
Stairs	38.55	38.94	38.67	7.0	7.8	7.2
Roughs	28.06	31.91	34.17	5.8	6.4	6.7

D.4 EVALUATION ON SOCIAL INTERACTIONS

In this experiment, we evaluate the agent's capability to handle complex social interactions. We follow SEAN 2.0 Tsoi et al. (2022) to design five interaction scenarios: Cross Path, Down Path, Leave Group, Join Group, and Empty. These five scenarios can be defined as below:

- *Cross Path:* A robot is positioned at $\mathbf{p_r}$ with orientation $\mathbf{o_r}$. Nearby, there is an agent located at $\mathbf{p_a}$, moving with velocity $\mathbf{v_a}$ and orientation $\mathbf{o_a}$, where $\mathbf{o_a}$ is perpendicular to $\mathbf{o_r}$.
- *Down Path:* A robot is positioned at $\mathbf{p_r}$ with orientation $\mathbf{o_r}$. Nearby, there is an agent at $\mathbf{p_a}$, moving with velocity $\mathbf{v_a}$ and orientation $\mathbf{o_a}$, where $\mathbf{o_a}$ is parallel to $\mathbf{o_r}$.
- *Leave Group:* A robot currently at position $\mathbf{p_r}$ originated from a starting position $\mathbf{p_r'}$, which made it a member of a group centered at $\mathbf{c_g}$. The robot is near an agent at $\mathbf{p_a}$, who is still part of the same group.
- *Join Group:* A robot positioned at $\mathbf{p_r}$ has a navigation target $\mathbf{p_r''}$, which will place it within a group centered at $\mathbf{c_g}$. The robot is also near an agent at $\mathbf{p_a}$, who is already a member of the group.
- *Empty:* A robot located at $\mathbf{p_r}$ has no other agents in its vicinity.

We evaluate the model trained on the SocialNav task, which has encountered diverse, randomly generated behaviors. We then test it on unseen interaction scenarios to assess its generalizability in social interactions. The results are shown in Table 10. We can draw three critical insights from the results.

1) *Superior performance in simple scenarios.* The model achieves the best performance in the "Empty" scenario, with the highest Success Rate (70%), perfect Social Navigation Score (1.00), and lowest Cost (0.30). This highlights the agent's strength in non-social or low-interaction environments but also underscores its limitations in handling complex social dynamics.

2) *Difficulty in coordinating parallel movements.* The "Down Path" scenario exhibits the lowest Success Rate (50%) and Route Completion (66.72%), indicating that parallel movement with nearby agents poses significant challenges. This suggests the need for improved adaptability to dynamic, aligned trajectories.

3) *Task completion vs. safety trade-off.* The "Leave Group" scenario achieves the highest Route Completion (82.90%), showcasing strong task-oriented behavior. However, it also incurs the highest Cost (12.90), reflecting a trade-off between completing tasks and maintaining safety in socially dense situations.

Table 10: **Evaluation on social interaction scenarios.** For each row of different matrics, 🟥 and ▨ indicates the best and worst performance among five scenarios.

Scenario	**Cross Path**	**Down Path**	**Leave Group**	**Join Group**	**Empty**
SR ↑	60%	50%	60%	60%	70%
Route Completion (%) ↑	80.46%	66.72%	82.90%	74.31%	79.05%
SNS ↑	0.96	0.95	0.92	0.99	1.00
Cost ↓	4.30	7.80	12.90	1.70	0.30

D.5 EVALUATION ON DIFFERENT SENSORS

In this experiment, we assess performance changes using different sensors: LiDAR, Depth, and RGB. We perform experiments on the PointNav task and train RL models using the PPO (Schulman et al., 2017) algorithm. To evaluate the models' ability to generalize to unseen scenarios, we test them on the MetaUrban-unseen dataset.

The results in Table 11 highlight the performance differences among LiDAR, Depth, and RGB sensors. LiDAR demonstrates superior performance across all metrics. This indicates that LiDAR's precise spatial information enables more efficient and safe navigation in unseen scenarios. Depth sensors show moderate performance, reflecting challenges in extracting accurate spatial features in complex urban environments. RGB sensors perform the worst across all metrics, likely due to the lack of 3D information and higher sensitivity to environmental variations. These results underscore the importance of robust spatial sensing, with LiDAR offering the most reliable input for generalizable navigation in unseen scenarios.

Table 11: **Evaluation on different sensors.** For each row of different matrics, 🟥 indicates the best performance among the three sensors.

Sensor	**LiDAR**	**Depth**	**RGB**
SR (%) ↑	87.77%	60.00%	54.00%
Route Completion (%) ↑	92.26%	69.36%	57.28%
SPL ↑	0.54	0.31	0.23
Cost ↓	1.03	1.40	3.20

E UNIQUE CHALLENGES IN URBAN MICROMOBILITY

In this section, we delve into and validate four unique challenges a mobile machine will encounter in public urban space, which is the stage of urban micromobility tasks, distinct from previous indoor and driving environments, *i.e.*, long horizon tasks in large-scale scenes, multifarious terrains, diverse obstacles, and dense pedestrians.

Long horizon tasks in large-scale scenes. In urban spaces, mobile machines need to perform long-horizon tasks with large-scale scenes connected with several street blocks, which are several orders of magnitude larger than indoor environments. We compare the performance of two models trained with different episode length settings. Setting-1: mean length with $10m$ (a common length following the indoor environment ProcThor (Deitke et al., 2022b)). Setting-2: mean length with $410m$ (a common length in urban micromobility tasks). Both models are then tested on the MetaUrban-test dataset. As shown in Table 12 (Left), the model trained with Setting-1 achieves poor performances when testing on the urban environments. However, when trained in an urban environment with a longer episode length (Setting-2), the model's performance improves dramatically. It indicates that long-horizon tasks in large-scale scenes bring a unique challenge to mobile machines, and validate the necessity of procedural generation of large-scale urban scenes in MetaUrban.

Table 12: **Unique challenges validation.** (Left) Long horizon tasks in large-scale scenes. (Right) Dense pedestrians.

	Setting-1	**Setting-2**
SR↑	10%	41%
SPL↑	0.08	0.38

	Setting-1	**Setting-2**	**Setting-3**	**Setting-4**
SR↑	24%	16%	10%	8%
Cost↓	0.51	0.75	0.74	0.68

Multifarious terrains. In indoor environments, most of the grounds are smooth and even, while in driving environments, roadway surfaces can only have slight cracks and damage. Yet, in public urban spaces, mobile machines will encounter multifarious complex terrains, such as slopes, stairs, and roughs. As shown in Table 9, different terrains will deeply influence the performance of mobile machines with different mechanical structures, such as wheelbase, wheel radius, and width, *etc.*. Results show that along with the increasing difficulty of terrains, all of the machines will fail because of getting stuck, barely moving, or toppling over. It indicates that multifarious terrains bring a unique challenge to mobile machines, and validate the necessity of the terrain generation system designed in MetaUrban.

Diverse obstacles. In indoor environments, even though there are many objects, the distribution has a large variation compared to urban spaces. In driving environments, there are only a few obstacles on roadways, such as cones and barriers. As shown in Figure 27, we compare the distribution of object category in MetaUrban with that in ProcThor (Deitke et al., 2022b), a state-of-the-art indoor simulation environment. We can observe a significant variation in object category distributions between public urban spaces and indoor spaces, although these two scenarios both accommodate a lot of objects. The diversity, particularity, and concentration of obstacles in urban spaces present a unique challenge for mobile machines. The statistical results validate the necessity of the pipeline of scalable obstacle filling in MetaUrban.

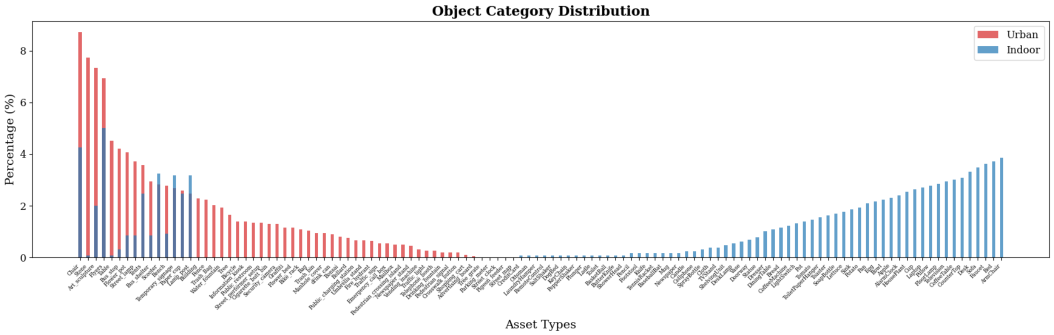


Figure 27: **Object category distributions.** Red: Distribution of object category in urban spaces. Blue: Distribution of object category in indoor spaces.

Dense pedestrians. In indoor environments, humans will share walkable spaces with humans; however, there are only 3-5 people in one room in common (as shown in Habitat 3.0 (Puig et al., 2023b)). In driving environments, except the intersections, there exist barely any shared spaces for pedestrians and vehicles. In contrast, in public urban spaces, almost all of the spaces for mobile machines are shared with pedestrians. We compare models trained with different pedestrian densities and locations. Setting-1: 10 pedestrians per 100-meter episode (a common scenario in an urban environment). Setting-2: 5 pedestrians per 100-meter episode (a common scenario in an indoor environment). Setting-3: 10 pedestrians per 100-meter episode but only shown in intersections (a common scenario in a driving environment). All the three models are then tested on the MetaUrban-test dataset. Setting 4: using the model trained in Setting-1 and testing on a density of 30 pedestrians per 100-meter episode (a crowd scenario in an urban environment).

Results are shown in Table 12 (Right). We take Setting-1 as a reference. With fewer pedestrians in Setting-2, the success rate will decrease, and the cost will increase significantly, indicating a higher frequency of bumping into pedestrians. With different distribution but the same pedestrian density in Setting-3, both success rate and cost degrade dramatically, indicating the unique challenge of sharing walkable regions with pedestrians. In Setting 4, we further increase the pedestrian's density and see a huge degradation in success rate but a moderate degradation in Cost. It indicates that having more pedestrians poses a significant challenge for the agent to reach the goal point. Interestingly, the agent still attempts to avoid pedestrians due to its effective training in public urban spaces. Results demonstrate that the high pedestrian density and interaction frequency in public urban spaces will place a unique challenge for mobile machines. It also validates the importance of the Cohabitant Populating module in MetaUrban.

F DATASHEET

Motivation	
For what purpose was the dataset created?	The dataset was created to enable agents training on diverse scenes and facilitate AI-driven urban micromobility research.
Who created and funded the dataset?	This work was created and funded by the MetaUrban team at the University of California, Los Angeles.
Composition	
What do the instances that comprise the dataset represent?	Each instance is a JSON file including the configuration of our MetaUrban environment and a specific seed.
How many instances are there in total (of each type, if appropriate)?	There are 12,800 urban scenes released in the MetaUrban-12K dataset, along with the code to sample substantially more.
Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?	We offer 12,800 urban scenes, with the ability to generate more using procedural generation scripts.
What data does each instance consist of?	Each scene is specified as a JSON file including the configuration of our MetaUrban environment and a specific seed.
Is there a label or target associated with each instance?	No.
Is any information missing from individual instances?	No.
Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?	Each urban scene is created independently, so there are no connections between the scenes.
Are there recommended data splits?	Yes. See Section 4 in the main paper.
Are there any errors, sources of noise, or redundancies in the dataset?	No.
Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?	The dataset is self-contained.
Does the dataset contain data that might be considered confidential?	No.
Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?	No.
Collection Process	
How was the data associated with each instance acquired?	Each scene was procedurally generated.

If the dataset is a sample from a larger set, what was the sampling strategy?	The dataset consists of 12,800 scenes, each by sampling the parameters of its composed elements.
Who was involved in the data collection process?	The authors were the sole individuals responsible for creating the dataset.
Over what timeframe was the data collected?	Data was collected in Sept. 2024.
Were any ethical review processes conducted?	No.
Preprocessing/Cleaning/Labeling	
Was any preprocessing/cleaning/labeling of the data done?	We label each object's location area and pivots to make them spawn in target functional zones and face a natural direction. We use VLMs to automatically label 2D images of cities worldwide, which enables the extraction of real-world category distribution of objects in urban spaces.
Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data?	There is no raw data.
Is the software that was used to preprocess/clean/label the data available?	The code related to preprocessing, cleaning, and labeling the data will be made available.
Uses	
Has the dataset been used for any tasks already?	Yes. See Section 4 of the main paper.
What (other) tasks could the dataset be used for?	The scenes can be used in a wide variety of tasks in urban micromobility, embodied AI, computer vision, and urban planning.
Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?	No.
Are there tasks for which the dataset should not be used?	Our dataset can be used for both commercial and non-commercial purposes.
Distribution	
Will the dataset be distributed to third parties outside of the entity on behalf of which the dataset was created?	Yes. We plan to make the entirety of the work open-source, including the code used to generate scenes and train agents, the scripts to get the MetaUrban-12K dataset, and the asset repositories.
How will the dataset be distributed?	The scene files will be distributed with a custom Python package. The code, asset, and repositories will be distributed on GitHub.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?	The scene dataset, 3D asset repository, and code will be released under the Apache 2.0 license.
Have any third parties imposed IP-based or other restrictions on the data associated with the instances?	For 3D human assets, we use Synbody (Yang et al., 2023). Its license is CC BY-NC-SA 4.0. For movement sequences, we use BEDLAM (Black et al., 2023). See `https://bedlam.is.tue.mpg.de/license.html` for its license.
Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?	No.
Maintenance	
Who will be supporting/hosting/maintaining the dataset?	The authors will be providing support, hosting, and maintaining the dataset.
How can the owner/curator/manager of the dataset be contacted?	For inquiries, email <metaurban_team@gmail.com>.
Is there an erratum?	We will use GitHub issues to track issues with the dataset.
Will the dataset be updated?	We will continue adding support for new features to make the urban scenes more diverse and realistic. We also intend to support new tasks in the future.
If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were the individuals in question told that their data would be retained for a fixed period of time and then deleted)?	The dataset does not relate to people.
Will older versions of the dataset continue to be supported/hosted/maintained?	Yes. Revision history will be available for older versions of the dataset.
If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?	Yes. The work will be open-sourced, and we intend to offer support to assist others in using and building upon the dataset.

Table 13: A datasheet (Gebru et al., 2021) for MetaUrban and MetaUrban-12K.

G PERFORMANCE

Single environment performance. We measure the single-environment performance of MetaUrban under varying street blocks, different densities of static objects, and dynamic agents in the scene. All experiments are conducted on a single Nvidia V100 GPU and in a single process. For the environment, there are approximately 200 objects covering $1500m^2$ on average. We sample 1,000 steps for actions and run 10 times to report the average and standard error results of FPS. For the RGB and depth image, we use the 128×128 resolution. On average, for the RGB, Depth, and LiDAR observation, we achieve 50±15, 60±10, and 120±12 FPS in training, respectively. The current simulation FPS performance has made MetaUrban applicable to many scenarios. It is faster than real-time in a single environment and is efficient for on-screen model testing, demonstration data collection, and human-in-the-loop learning.

The performance is closely dependent on the various enabled simulation features. In experiments, users can choose to turn off some features based on specific experimental needs. Taking the result of RGB sensor as "Baseline", we report the performance of the setting without advanced features in the column "Simplification" of Table 14, such as shadows, sky rendering, and high-precise physical simulation. This can improve the mean performance by 8 FPS. To balance feature customization with performance, we will provide users with flexible interfaces that allow them to switch specific features based on their requirements.

Multiple environments performance. To further enhance the simulation efficiency, we have added support for distributed training, enabling scalable performance on multiple environments across multiple machines, which makes the single environment FPS not a bottleneck of model training. We report the performance changing with different numbers of environments in Table 14. It achieves up to 685 FPS on a single GPU with 32 environments running in parallel. The performance scales consistently with additional environments, ensuring efficiency for large-scale training tasks.

Table 14: **Performance of different settings.**

Settings	**Baseline**	**Simplification**	**8 Envs**	**16 Envs**	**32 Envs**
FPS (mean)	50	58	205	479	685

H ROBUSTNESS

We trained PPO on PointNav with different seeds and found that the variance of the performance across different seeds is small. The success rate of the PPO agents is 0.695 ± 0.014 on the MetaUrban-test set and 0.638 ± 0.060 on the MetaUrban-unseen set.

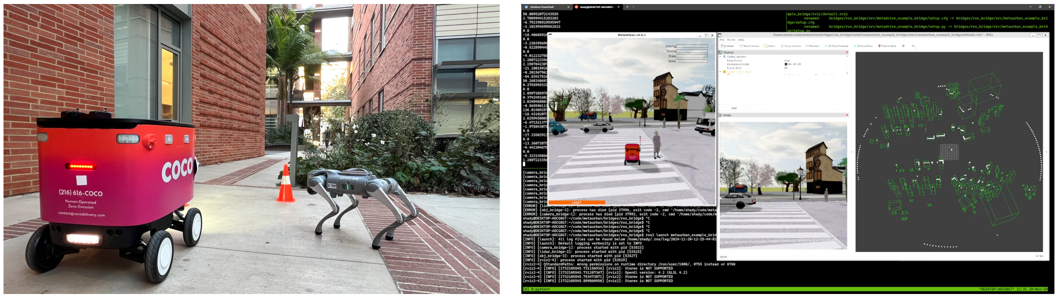
I DISCUSSION

Impact. As the first urban space simulator, MetaUrban could benefit broad areas across Embodied AI, Economy, and Society. 1) *Embodied AI.* MetaUrban contributes to advancing areas such as robot navigation, social robotics, and interactive systems. It could facilitate the development of robust AI systems capable of understanding and navigating complex urban environments. 2) *Economy.* MetaUrban could be used in businesses and services operating in urban environments, such as last-mile food delivery, assistive wheelchairs, and trash-cleaning robots. It could also drive innovation in urban planning and infrastructure development by providing simulation tools and insights into how spaces are utilized, thereby enhancing the economic and societal efficiency of public urban spaces like sidewalks and parks. 3) *Society.* By enabling the safe integration of robots and AI systems in public spaces, MetaUrban could support the development of assistive technologies that can aid in accessibility and public services. Using AI in public spaces might foster new forms of social interaction and community services, making urban spaces more livable and joyful. 4) *Potential negative societal impacts.* The integration of AI and robots in urban environments, while beneficial, raises several concerns. Increased surveillance could infringe on privacy, while automation may lead to job displacement and exacerbate economic inequalities. Societal dependency on technology poses risks of dysfunction during failures, and the presence of robots might alter social norms and interactions. Thus, the environmental impact of manufacturing and operating urban simulators must be carefully managed. Addressing these issues is crucial for ensuring that the benefits of such technologies are realized without detrimental societal consequences.

Limitations. 1) *Real-world scene distribution.* In this work, we extract object category distribution from real-world data of urban spaces. Other than the real-world distribution of object categories, the distribution of object location and scene layout is also important for constructing specialized scenes for agent training. Extraction of such distribution relies on an accurate reconstruction of 3D scenes from real-world videos or even images, and thus is extremely challenging. An interesting direction is extracting real-world scene distribution from in-the-wild videos, including object category, object location, and scene layout. Then, we can build a digital twin of a target scene for the agent's training. It could help to develop scene-specific agents. 2) *Interactive agent behaviors.* In this work, we construct the environmental agents‘ dynamic with deterministic methods, determining their movements and trajectories with rules. However, in the real world, all environmental agents are interactive; their behaviors are affected by each other and the surrounding environments. An interesting research direction is to endow personal traits like job, personality, and purpose to agents and harness the advances of LLMs (Achiam et al., 2023) and LVMs (Liu et al., 2023a) to form social (Puig et al., 2023a) and interactive behaviors (Park et al., 2023) of agents in urban scenes spontaneously. 3) *Robots' additional capability learning.* In urban micromobility, safe navigation through the city is the primary goal for mobile machines. However, additional capabilities, such as locomotion and manipulation, can enable robots to perform more complex tasks in urban spaces. Thus, an important direction is to extend MetaUrban to support additional capabilities learning gradually. It could enable various complex but important services in urban environments. 4) *Efficiency.* In this work, different from indoor scenes and driving simulators, MetaUrban supports generating complex interactive urban scenes with arbitrary scales. However, with the increase in scale, the number of objects and dynamic agents will surge dramatically, which will bring the degradation of the efficiency of physical simulation and rendering. A promising direction is to integrate more sophisticated physical engines and renders.

Real-world deployment support. We position MetaUrban as an Embodied AI simulator that aims to enable fast model training and evaluation before real-world deployment of the physical robots. We followed the standards of existing popular embodied AI simulators, *i.e.*, AI2-THOR (Kolve et al., 2017), Habitat (Savva et al., 2019), and ProcTHOR (Deitke et al., 2022b), so there is no real-world experimentation provided in the current version.

We will support the long-term development and maintenance of MetaUrban to become a sustainable infrastructure for the community, so we fully recognize that real-world experiments are essential for practical applications. Thus, based on the current MetaUrban simulator, we are constructing an end-to-end experimentation component that extends the simulation to the real-world deployment of robots.



Real-World Deployment on Robots

ROS 2 Support

Figure 28: **Sim-to-real support.** (left) Real-world deployment on two typical robots. (Right) The ROS 2 is supported by MeteUrban.

Preliminary experiments with Unitree's Go2 quadruped robot and COCO Robotics' wheeled robot (as shown in Figure 28 (Left)) showed that training robots with abstract observations, such as depth maps, combined with domain randomization, already achieved good transferability to real-world environments. These real-world evaluations offer valuable insights that can inform the future development roadmap of MetaUrban.

To support the real-world deployment for broad users, MetaUrban fully supports ROS 2 (as shown in Figure 28 (Right)). We enabled seamless communication between the simulator and the ROS 2 ecosystem. This support allows users to control robots, pedestrians, and other mobility devices directly through ROS 2 nodes and topics, ensuring smooth integration with existing robotics software pipelines. Currently, we are actively building benchmarks in real-world experiments and will provide a detailed analysis in the final version.

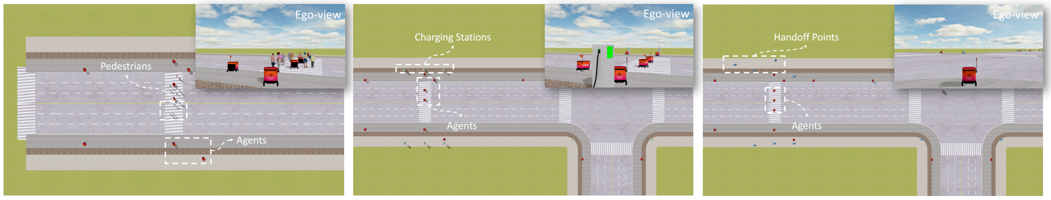
Multi-agent learning support. MetaUrban has supported multi-agent learning. We have defined three multi-agent tasks so far: 1) traffic management at crosswalks, 2) charging queue racing, and 3) package handoff between robots.

1) *Traffic management at crosswalks.* Task definition: N robots navigate through a shared crosswalk while heading to unique target points. They must coordinate to avoid collisions with other robots and dynamic obstacles. Example scenario: Eight robots approach a crosswalk from different directions. Each robot carrying time-sensitive packages must decide its crossing strategy to minimize delays while ensuring safety in a congested urban environment (Figure 29 (a)).

2) *Charging queue racing.* Task definition: N robots compete for access to M charging stations while balancing battery constraints and delivery deadlines. They must decide whether to queue, continue tasks, or seek alternative stations. Example scenario: Twelve robots with varying battery levels approach eight charging stations. One robot with a critically low battery must secure a spot to avoid running out of charge, while others evaluate trade-offs between waiting in a queue and traveling farther to an available station to meet delivery deadlines (Figure 29 (b)).

3) *Package handoff between robots.* Task definition: N robots collaborate to deliver packages via K predefined handoff points. Robots must strategically select handoff points and partners to optimize delivery time and energy usage. Example scenario: A robot carrying a package has a low battery and cannot reach the distant delivery location. It heads to a nearby handoff point, where another robot, fully charged and closer to the destination, takes over the package, ensuring timely delivery without interruptions (Figure 29 (c)).

We will release the code for multi-agent learning on the three tasks described above in the first official version. Additionally, we will continuously integrate new multi-agent micromobility tasks into MetaUrban to enhance community support.



(a) Traffic Management at Crosswalks (b) Charging Queue Racing (c) Package Handoff between Robots

Figure 29: **Multi-agent learning support.**

Sustainable ecosystem building. Building a sustainable ecosystem for MetaUrban requires addressing several critical aspects to ensure its long-term viability, adaptability, and impact. Following Business Model Generation (Osterwalder & Pigneur, 2010), we identify six critical aspects for MetaUrban as delivering clear value to users, establishing strategic partnerships, fostering strong community engagement, ensuring accessibility through effective channels, maintaining a robust financial model, and optimizing operational costs. Together, these elements enable MetaUrban to thrive as an open-source infrastructure that supports research, development, and deployment in micromobility and embodied AI.

Value propositions. Providing meaningful value to users is essential for the growth of an ecosystem. MetaUrban shortens development time and reduces research and development (R&D) costs through efficient simulation. It also accelerates research and equips students with in-demand skills, which enhances their employability. These value propositions attract a diverse range of stakeholders from academia, industry, and government, creating a strong foundation for widespread adoption and collaboration.

Key partners. Strategic partnerships are crucial for the growth and sustainability of MetaUrban. Collaborating with research institutions, technology companies, and government organizations grants access to advanced technologies, expertise, and funding. These partnerships enhance the platform's credibility and ensure that its development aligns with real-world needs, making MetaUrban an essential resource for both research and practical applications.

Customer relationships. Establishing strong relationships with the community is essential for ensuring user retention and encouraging active participation in the ecosystem. MetaUrban engages users through workshops, competitions, and networking events. Additionally, it offers technical support via Slack groups and GitHub issues. These initiatives create a collaborative environment where users can contribute, share knowledge, and influence the platform's development while addressing their specific needs.

Channels. Effective communication and collaboration channels are vital for accessibility and user engagement. MetaUrban utilizes a dedicated website, an open-source repository, a professional document, and a chat group for discussions. These channels allow for the easy sharing of updates, encourage contributions, and provide spaces for collaborative problem-solving, ensuring that users can interact with the platform effortlessly and derive maximum benefit from it.

Revenue streams. Sustainable funding is essential for achieving long-term success. MetaUrban obtains financial support from a variety of sources, including public research programs, government grants, and sponsorships from technology stakeholders. This diverse revenue model guarantees stability and enables ongoing investments in platform enhancements, community events, and operational maintenance.

Cost structure. Effectively managing operational costs enables MetaUrban to allocate resources to areas that create a significant impact. The cost structure emphasizes costs in development labor, server, and infrastructure expenses, and sponsorship of competitions. These expenditures foster innovation, support the community, and uphold the platform's technical quality, ensuring it remains a valuable tool for users.

By addressing these interconnected aspects, MetaUrban will create a sustainable and collaborative open-source ecosystem. This strategic approach ensures the platform continues to support cutting-edge research and real-world applications in micromobility and embodied AI, fostering innovation and creating lasting impact across academia, and industry.

Future work. 1) *Foundation model.* MetaUrban can easily generate infinite urban scenes with a large quantity of semantics and complex interactions, which could facilitate the pre-training of foundation models (like LLMs and LVMs) that can be used for downstream agent learning tasks. 2) *Human-robot cohabitate.* Mobile machines have started emerging in the urban space, which makes it no longer exclusive to humans. We plan to work with urban sociologists to study the influence of robots on human urban life through both simulation and field experiments. 3) *Improve limitations.* The directions discussed in limitations and sim-to-real gaps are also meaningful future work we will conduct. In summary, MetaUrban, as a new urban environment simulator, will bring a lot of new interesting research directions. We are dedicated to maintaining MetaUrban in the long term and supporting the community's efforts to develop it into a sustainable infrastructure.