2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

UrbanVLA: A Vision-Language-Action Model for Urban Micromobility

Anqi Li[1,2,*] Zhiyong Wang[2,*] Jiazhao Zhang[1,2,*] Minghan Li[2]
Yunpeng Qi[3,4] Zhibo Chen[3] Zhizheng Zhang[2,4,†] He Wang[1,2,4,†]
[1]Peking University [2]Galbot [3]USTC [4]BAAI
Project Page: https://pku-epic.github.io/UrbanVLA-Web/

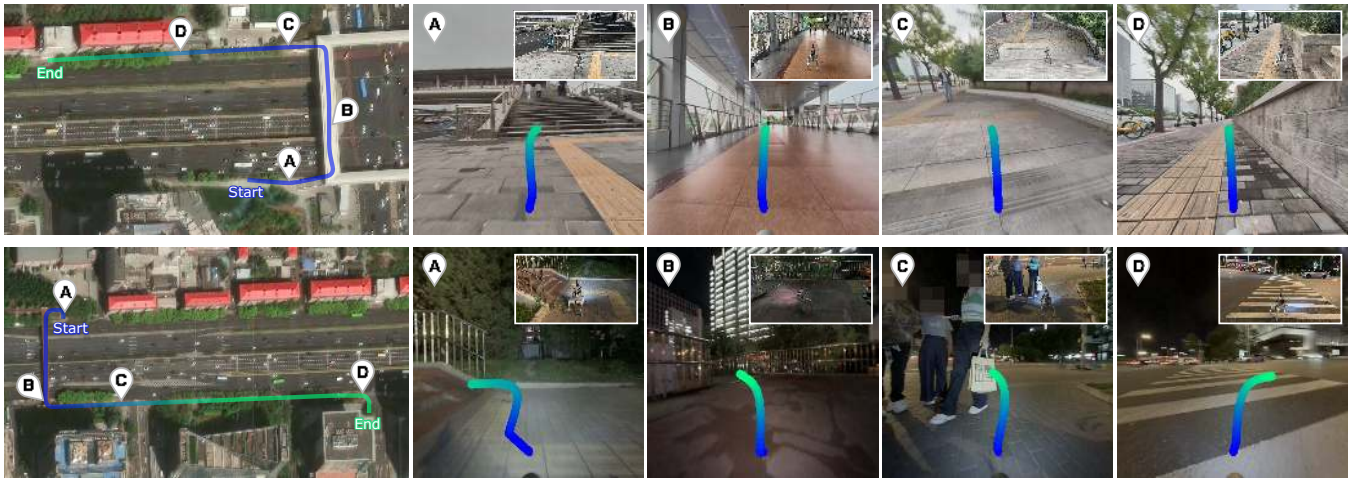

Fig. 1: **Real-world deployments of UrbanVLA** demonstrate zero-shot generalization across diverse environments featuring unseen layouts, dynamic obstacles, and varying illumination, and highlight its ability to perform long-horizon urban micromobility tasks spanning over 500 meters.

***Abstract*— Urban micromobility applications, such as delivery robots, demand reliable navigation across large-scale urban environments while following long-horizon route instructions. This task is particularly challenging due to the dynamic and unstructured nature of real-world city areas, yet most existing navigation methods remain tailored to short-scale and controllable scenarios. Effective urban micromobility requires two complementary levels of navigation skills: low-level capabilities such as point-goal reaching and obstacle avoidance, and high-level capabilities, such as route–visual alignment. To this end, we propose UrbanVLA, a route-conditioned Vision-Language-Action (VLA) framework designed for scalable urban navigation. Our method explicitly aligns noisy route waypoints with visual observations during execution, and subsequently plans trajectories to drive the robot. To enable UrbanVLA to master both levels of navigation, we employ a two-stage training pipeline. The process begins with Supervised Fine-Tuning (SFT) using simulated environments and trajectories parsed from web videos. This is followed by Reinforcement Fine-Tuning (RFT) on a mixture of simulation and real-world data, which enhances the model's safety and adaptability in real-world settings. Experiments demonstrate that UrbanVLA surpasses strong baselines by more than 55% in the SocialNav task in MetaUrban. Furthermore, UrbanVLA achieves reliable real-world navigation, showcasing both scalability to large-scale urban environments and robustness against real-world uncertainties.**

I. Introduction

Urban micromobility [1], [2], encompassing lightweight embodied platforms such as delivery robots, assistive wheelchairs, and guide robots, is emerging as one of the most promising applications of embodied intelligence. To achieve deployment at scale, these systems must operate in dynamic and uncertain environments that, unlike the structured road networks of autonomous driving, unfold in unstructured pedestrian spaces, making reliable and scalable navigation a fundamental challenge.

Urban micromobility is achieved predominantly through visual navigation, serving as the primary paradigm for autonomous operation. Traditional Simultaneous Localization and Mapping (SLAM)-based pipelines [3]–[6] provide reliable navigation in constrained settings but depend heavily on detailed maps such as occupancy grids or HD maps. This reliance severely limits scalability in large and dynamic urban environments, where maintaining accurate maps is costly and often infeasible. To address these limitations, learning-based approaches [7]–[13] formulate urban micromobility as a point-goal navigation task, with foundation model methods such as CityWalker [7] leveraging waypoints from consumer navigation tools (*e.g.*, Google Maps) to provide high-level guidance. However, navigation tools preserve only coarse route-level topological continuity while neglecting geometric accuracy, causing frequent misalignment between waypoints and the physical world.This makes existing point-goal navigation methods difficult to use in large-scale, real-world urban environments.

Recently, navigation-purpose VLA methods have demonstrated both strong performance and generalization [14]–

*Joint first authors †Corresponding authors

[20]. By fine-tuning vision-language models, they align natural language commands with visual observations, enabling robots to ground high-level instructions into low-level actions for reliable navigation in previously unseen environments. Despite this progress, VLAs remain inadequate for navigating urban environments over long distances. They must interpret noisy routes from navigation apps by aligning them with visual cues and commonsense knowledge. Furthermore, they need to adhere to a complex set of rules, including traffic signals, sidewalk etiquette, while simultaneously adapting in real time to dynamic obstacles, varied terrain, and dense pedestrian flows.

To address these challenges, we propose a route-conditioned Vision-Language-Action (VLA) framework for urban micromobility. Our VLA model takes structured route descriptions (roadbooks) as input and directly predicts trajectory waypoints for route following. The VLA learns to align these "roadbooks" with visual observations, generating local trajectories that translate high-level routes into low-level navigation waypoints. This process enables reliable, long-horizon navigation over large areas, thereby allowing our framework to leverage noisy navigation tools as effective guidance for robust and scalable urban micromobility.

To train our route-conditioned VLA, we build upon a pre-trained navigation foundation model [19] and adopt a two-stage training approach. In the first stage, we perform supervised fine-tuning on two complementary tasks: (i) a navigation task using trajectories from the MetaUrban simulator [21] and Sekai web navigation videos [22], where navigation 'roadbooks' are derived from ground-truth paths and paired with visual observations, and (ii) a real-world Video Question Answering (VideoQA) task [22], [23] to enhance real-world scene understanding. In the second stage, we apply offline reinforcement fine-tuning using Implicit Q-Learning (IQL) [24] on a sim-real aggregated dataset. This two-stage training equips our VLA model with the ability to interpret noisy routes, adhere to navigation norms, and adapt to real-world urban complexity.

Experiments demonstrate that our method achieves 55% and 56% performance improvement compared to baselines in the SocialNav task in *MetaUrban-test* and *MetaUrban-unseen* benchmark [21], respectively. Real-world experiments further prove the generalbaility of our method. We summarize our contribution as follows:

- We propose the first route-conditioned VLA designed for urban micromobility, which integrates the guidance of high-level urban navigation tools with vision-language policy learning.
- We develop a simulation-to-reality pipeline, leveraging a route-lifting algorithm to formulate a sim-real aggregated dataset, achieving SOTA performance in simulation and demonstrating real-world generalization.
- We improve safety-critical behaviors by introducing IQL-based reinforcement fine-tuning, improving obstacle avoidance, pedestrian interaction, and traffic compliance.

II. Related Works

Robot Urban Navigation. Urban navigation has long relied on SLAM-based systems [3], [4], [25], [26], typically using pre-built maps (*e.g.*, point clouds or HD maps) for localization and planning. Although effective in structured settings, such methods require costly map construction and careful tuning, limiting scalability. Learning-based alternatives reduce map dependence. OpenNav [27] employs Multimodal Large Language Models (MLLMs) to generate Bird's-Eye-View (BEV) maps and convert free-form instructions into trajectories, while CityWalker [7] learns policies from large-scale urban videos. However, existing methods remain constrained to sensor–action correlations and lack comprehensive scene understanding. To address this, UrbanVLA introduces an end-to-end VLA framework for urban navigation that integrates multimodal perception and semantic reasoning to embed scene semantics into route-conditioned decision-making in dynamic environments.

VLA Models for Navigation. VLA models extend pre-trained VLMs with action generation and show strong generalization in embodied navigation [14]–[20]. By jointly modeling vision, language, and action, they enable instruction-following under real-time observations. NavFoM [19] trains a navigation foundation model on cross-embodiment data for consistent multi-platform performance. However, existing models face challenges in urban navigation, which requires sophisticated spatial reasoning, commonsense adherence, and safety-critical decision-making. To address this, we design a sim-to-real fine-tuning pipeline using the MetaUrban simulator and the Sekai dataset to acquire navigation policies for dynamic urban environments.

Reinforcement Learning for VLA. Reinforcement Learning(RL) has become a key post-training strategy for VLA models [28]–[31], improving robustness over imitation learning. In navigation, VLN-R1 [32] applies temporally-decayed rewards for long-horizon optimization, and OctoNav [33] uses hierarchical rewards with online adaptation. However, VLA models for urban navigation face safety-critical challenges such as pedestrian interactions and traffic rule compliance. To address this, we use Implicit Q-Learning (IQL) [24] for reinforcement fine-tuning to enhance safety-aware decision-making in dynamic urban environments.

III. Method

A. Problem Formulation

Task Definition. We formulate the route-conditioned urban navigation task as follows: at the current time step T, given a high-level target route formulated by a sequence of 2D coordinates $\mathcal{R} = \{r_0, ..., r_n\}$ where $r_i \in \mathbb{R}^2$ is a planar coordinate sampled from the target route under the agent's egocentric frame, and a sequence of RGB image observations $\mathcal{O}_{\text{vis}} = \mathbf{o}_{1:T}^{1:C} \in \mathbb{R}^{W \times H \times 3}$ taken by C different cameras at time steps $\{1, ..., T\}$, the agent is required to learn a navigation policy $\pi(\mathcal{R}, \mathcal{O}_{\text{vis}}) \mapsto \tau$, where $\tau = \{\tau_1, \tau_2, ...\tau_N\}$ is a navigation trajectory formulated by N waypoints with $\tau_i \in \text{SE}(2)$ representing the predicted 2D position and

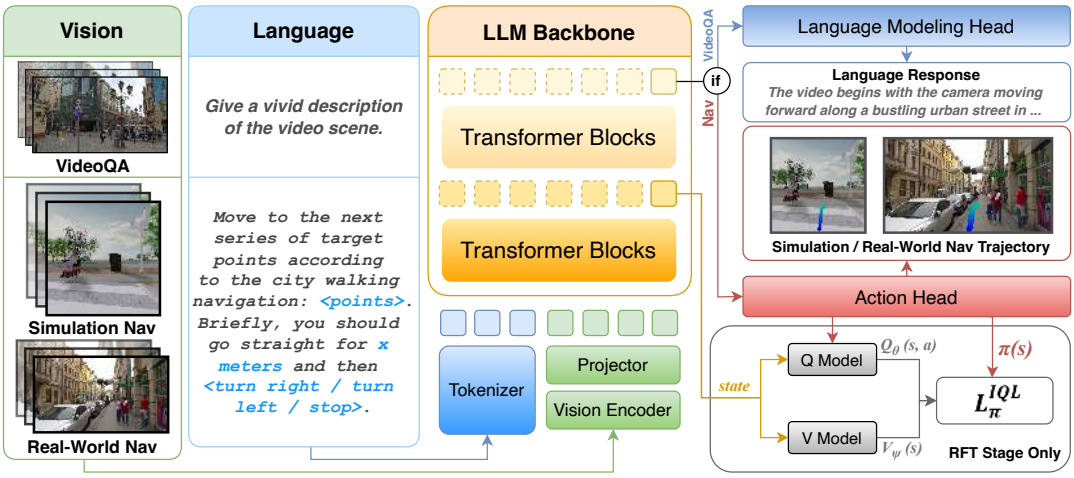

Fig. 2: **Overview of UrbanVLA.** We collect diversified VideoQA data and urban micromobility demonstrations to train the model via a two-stage pipeline. In the SFT stage, UrbanVLA learns essential urban navigation capabilities such as goal-reaching, collision avoidance, and social compliance; in the RFT stage, we refine the model using a sim-real aggregated dataset with IQL, enhancing robustness in real-world scenarios.

orientation under the current egocentric frame that safely drives the agent along the target route toward its destination.

Pipeline Overview. Figure 2 shows the overall pipeline of our approach. We leverage a pre-trained navigation foundation model NavFoM [19] as our base model, and apply a two-stage fine-tuning strategy via supervised fine-tuning (SFT) and reinforcement fine-tuning (RFT), respectively. Specifically, we apply a prompt template to encode the high-level 'roadbook' instructions into a linguistic form $\mathcal{I}$. Following existing VLM approaches [23], [34], [35], we embed $\mathcal{I}$ to obtain language tokens E_L, as well as encode visual observations $\mathcal{O}_{\text{vis}}$ using pre-trained vision encoders to obtain visual tokens $E_{1:T}^{1:C}$. Together, we feed E_L and $E_{1:T}^{1:C}$ into the Large Language Model (LLM) backbone. In the SFT stage, following previous work [18], the dual-branched VLA learns to perform two types of tasks, VideoQA and route-conditioned navigation. We decode the tokens generated using a language head and an action head, respectively, to acquire the linguistic answer and navigation trajectory.

In the RFT stage, we further fine-tune UrbanVLA on a hybrid dataset that combines expert demonstrations collected from both simulation and real-world environments. We adopt Implicit Q-Learning (IQL) [24], a widely used offline RL algorithm, to effectively utilize these limited hybrid data while mitigating overoptimism in out-of-distribution (OOD) samples. To estimate the action-state function Q and value function V in IQL that take current state as input, we encode language instruction $\mathcal{I}$ and visual observation $\mathcal{O}_{\text{vis}}$ into a unified state representation s using the well-tuned vision-language backbone after the SFT stage. The reward function $r(s,a)$ is delicately designed, considering both trajectory efficiency and navigation safety, to allow efficient data collection in the real world, as well as alignment between simulation and real world.

B. UrbanVLA Architecture

High-Level Route Encoding. The high-level route instructions in the urban navigation task should be converted into a form that is interpretable by VLA models and aligned with the data schema of the prevalent urban navigation tools [36] to facilitate large-scale deployment. Consequently, we convert the route instructions into a structured linguistic representation comprising two components. First, a set of waypoints sampled from the high-level route provides the agent with the overall geometry and orientation of the forthcoming path. Second, distance and direction instructions for the next turn (for example, 'turn right in 30 meters') convey essential information to transition between blocks, a critical scenario for successful urban navigation.

Specifically, given a high-level navigation route $\mathcal{R}$, we first resample the upcoming D meters of the route trajectory at a distance of d meters (we set $D = 40$ and $d = 2$, resulting in 20 waypoints), and convert them into the robot frame. Subsequently, when training, we apply a corner detection algorithm (see Section III-D) to segment the route into blocks and then derive block-level distance and direction cues from these segments; while in a real-world setting, this information can be directly acquired from the API of the city navigation tool. Finally, we formulate the above information into an instruction template to obtain the navigation instruction $\mathcal{I}$.

VLA Model Forwarding. Given multi-view RGB observations $\mathcal{O}_{\text{vis}} = \mathbf{o}_{1:T}^{1:C} \in \mathbb{R}^{W\times H\times 3}$, for route-conditioned urban navigation tasks, we apply a visual sliding window to retain the nearest k frames $\mathcal{O}_{\text{retain}} = \mathbf{o}_{T-k+1:T}^{1:C}$. Following recent advanced VLM works [23], [37], [38], we encode visual information using two pre-trained vision encoders (DINOv2 [39] and SigLIP [40]) and concatenate the visual features obtained in the channel dimension to formulate the final visual features $v_1^{1:C}, \ldots v_{T-1}^{1:C}, v_T^{1:C}$. Subsequently, we down-sample the features with grid pooling strategy, and use

a cross-modal projector (double-layer MLP) [34] to project the visual features into the embedding space of the LLM backbone and acquire the visual tokens $E_{1:T}^{1:C}$. We then embed the navigation instruction $\mathcal{I}$ in language tokens E_L. Together, we feed all tokens into an LLM backbone (Qwen2 [41]). The model generates tokens in two manners: for navigation tasks, we capture the generated action token E_T^A at the current time step and decode it through an MLP-based action model to obtain the navigation trajectory τ:

$$\begin{aligned} E_T^A &= \text{LLM}(E_L, E_{1:T}^{1:C}), \\ \tau &= \text{ActionModel}(E_T^A); \end{aligned} \tag{1}$$

while for VideoQA tasks, the model autogressively generates a set of language tokens, which are then decoded through the language model head, as shown in Figure 2.

C. Training Strategy

Supervised Fine-Tuning. We first apply Supervised Fine-tuning (SFT) to the base model NavFoM [19]. In this stage, the model learns from urban navigation demonstrations generated by a PPO expert in simulation [21], as well as web-scale urban travel data that capture human navigation behavior in real-world environments, using a Mean Squared Error (MSE) loss. The SFT stage is designed to instill basic goal-reaching capabilities while exposing the model to the diversity and complexity of urban navigation tasks, thus enhancing its generalization to real-world scenarios.

A key challenge in leveraging such demonstration data is the lack of 'roadbook' instructions. Real-world demonstrations typically provide only the ground-truth trajectory, while simulators often offer perfect route information generated by collision-free planning algorithms such as ORCA [42] and P&R [43]. Using such idealized routes directly as conditions may cause the model to overfit to the input trajectory, thus compromising its generalizability in real-world scenarios.

To address this problem, we introduce *Heuristic Trajectory Lifting (HTL)*, a heuristic algorithm that lifts high-level route information from the raw trajectory in urban navigation data, encouraging the model to learn from visual cues rather than relying solely on idealized route inputs. The raw trajectory is first preprocessed: a Savitzky–Golay filter [44] is applied to denoise web trajectories, while simulator-generated smooth trajectories are used directly. The self-intersecting or otherwise low-quality paths are then removed. Next, significant turning points are detected (see Section III-D) to split the trajectory into several segments. To capture the ambiguity of real-world navigation, each segment is perturbed with Gaussian positional noise, reflecting that high-level directives (*e.g.*, 'go straight') correspond to a corridor of feasible paths rather than a single curve. Finally, noisy segments are smoothly merged and resampled in a fixed spatial step, resulting in the abstracted route $\mathcal{R}$. In summary, this pipeline allows us to generate a large-scale, unified dataset for the SFT stage from both simulated and real-world sources with automatically anotated 'roadbook' instructions.

Reinforcement Fine-Tuning. Building on the capabilities acquired during SFT, UrbanVLA demonstrates strong performance in navigating diverse urban environments, such as intersections, turns, and varying street layouts. To further improve its skills, particularly in collision avoidance and handling ambiguous cues, we adopt an offline RL approach based on Implicit Q-Learning (IQL) [24], which suits well for offline data and mitigates out-of-distribution action issues.

We formulate the route-conditioned urban navigation task as a Partially Observable Markov Decision Process (POMDP) $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, r, \gamma)$, where $\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, r$, and γ denote the state space, action space, observation space, transition model, reward function, and discount factor, respectively. For IQL, we first train value networks $Q_\theta(s,a)$ and $V_\psi(s)$ for advantage estimation. To properly represent the input state $s \in \mathcal{S}$, we employ the VLA model after the SFT stage (denoted as UrbanVLA-SFT) as the encoder. Specifically, at current timestep T, the observation $o_T = (\mathcal{O}_{\text{vis}}, \mathcal{I}) \in \mathcal{O}$ is fed into UrbanVLA-SFT, and the hidden state from the n-th transformer layer of the LLM backbone is extracted to serve as the state embedding $s = H_T^{(n)} \in \mathbb{R}^{\text{dim}}$ (as shown in Figure 2), where dim is the hidden dimension of the LLM. We empirically set $n = 17$, as we observe that mid-layer hidden states yield more robust value estimation than top-layer states. This is potentially due to the latter being overly specialized for predicting action logits. We find this representation effectively encodes the visual and language context, offering task-aware embeddings that stabilize Q-learning [30]. For the input action, we simply vectorize the predicted trajectory: $a := \text{vec}(\tau) \in \mathbb{R}^{3N}$, with τ and N defined in Section III-A.

Based on the above formulation, IQL learns a value function $V_\psi(s)$ and a Q-function $Q_\theta(s,a)$ from the offline dataset $\mathcal{D}$, and the policy $\pi(s)$ is updated via an advantage-weighted regression (AWR) objective

$$\mathcal{L}_\pi^{\text{IQL}} = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[\exp\big(\beta A(s,a)\big)||a - \pi(s)||^2\Big], \tag{2}$$

where $A(s,a) = Q_\theta(s,a) - V_\psi(s)$ is the advantage estimation, and β is an inverse temperature parameter that trades between imitation and performance improvement [24].

The reward function $r(s,a)$ is designed for: (i) ease of acquisition during teleoperation to minimize post-processing, and (ii) sim-real consistency to provide a unified learning objective and enhance data efficiency. We formulate the reward function as

$$r(s,a) = \lambda_{\text{comp}}\, l_{\text{completion}} - \lambda_{\text{coll}}\, \mathbb{1}_{\text{collision}} - \lambda_{\text{dev}}\, \mathbb{1}_{\text{deviation}}, \tag{3}$$

where $l_{\text{completion}}$ denotes the increment of trajectory completion aligned with the ground-truth route, while $\mathbb{1}_{\text{collision}}$ and $\mathbb{1}_{\text{deviation}}$ indicate whether a collision or an excessive deviation from the route corridor occurs, respectively. The weighting coefficients $\lambda_{\text{comp}}, \lambda_{\text{coll}}, \lambda_{\text{dev}}$ are adjusted to balance efficiency and safety and are specified in Section III-D. This formulation relies only on readily available or easily measurable quantities (see the user interface in Figure 3), and enables the seamless transfer of the learned policy to real environments.

Based on this design, we collect a sim-real aggregated dataset for the RFT stage, comprising 2,400 episodes (approximately 40 hours) in the MetaUrban simulator using a PPO expert, along with roughly 8 hours of real-world demonstrations via human teleoperation. Please refer to Section III-D for details. In summary, during the RFT stage, the large scale simulation data facilitate rapid convergence of the Q-learning, and the teleoperation data is efficiently learnt to strengthen the model's real-world safety performance and corner case decisions.

D. Implementation Details

Our model is trained on a cluster server equipped with 8 NVIDIA H100 GPUs for approximately 12 hours, resulting in 96 GPU hours in total. The VideoQA dataset is collected from LongVU [23] and Sekai [22]. Unlike the sliding-window mechanism introduced in the navigation task, we retain all visual frames before feeding them into the model. We supervise the result using cross-entropy loss.

For the corner detection algorithm mentioned in Section III-B, specifically, we adopt a window-based detection algorithm: for each point, we compute the turning angle between the vectors formed by its neighbors in a window of k points. Points with angles above a threshold are marked as candidates. The subsequent candidates are merged by taking the middle point, and a greedy selection step enforces a minimum arc-length spacing to remove redundant corners.

In the RFT stage, the real-world data was collected in the city area of Beijing. For this dataset, we collected visual observations, navigation route instructions obtained by querying navigation tool APIs in real time, and reward signals either annotated through the user interface or generated using records from a real-world LiDAR-Odometry system. We intentionally collected several scenes in which the navigation information is inconsistent with real-world conditions. Notably, a small amount of sub-optimal real-world data (that yields negative reward terms) was also intentionally collected to ensure the stability of Q-learning [24], [45]. The weighting coefficients λ_{comp}, λ_{coll}, and λ_{dev} in the reward function are set to 0.5, 1, and 1, respectively.

IV. Experiments

A. Experiment Setup

Simulation Setup. We conduct our experiments based on the Point Navigation (PointNav) and Social Navigation (SocialNav) benchmarks built on the MetaUrban simulator [21]. These two tasks provide an insightful examination on our model's capabilities of route following, collision avoidance, social compliance, and generalizability to diverse scenery layout. To allow fair comparison, our model is trained on 12,800 scenes in the *MetaUrban-train* dataset, and is tested on 1,000 scenes in the *MetaUrban-test* and 100 scenes in the *MetaUrban-unseen*.

In our evaluation, we choose a wheelchair embodiment. While the baselines output continuous steering and acceleration [21], UrbanVLA focuses on high-level spatial planning by predicting 2D trajectory waypoints. To execute this, the agent is directly displaced to the predicted waypoints. We bound the maximum step distance to the radius of the embodiment (1.5 meters) to ensure accurate collision detection.

Fig. 3: **Real-world deployment of UrbanVLA.** Our system consists of a quadruped robot equipped with GPS, Wi-Fi, a camera, and an onboard computing unit, along with a mobile-deployable console for real-time monitoring, sending navigation targets, visualizing maps and model predictions, and annotating teleoperation data used for reinforcement learning.

Real-World Setup. We conduct extensive real-world experiments across multiple urban blocks, covering diverse scenarios such as overpasses, pedestrian crossings, and environments with dense static and dynamic obstacles. Following NavFoM [19], our model runs on a remote server equipped with an NVIDIA RTX 4090 GPU, communicating with a Unitree Go2 quadruped via Web-ADK. The robot collects visual observations from four cameras mounted on its front, left, right, and rear, and relies on an off-the-shelf local planner to execute the trajectories generated by our model. High-level route instructions are obtained from Amap [36] via its Web API, while a GNSS module provides the real-time position and orientation of the quadruped to convert these instructions into the robot's local frame. Figure 3 illustrates our real-world deployment setup. The system operates at 2 Hz, which is sufficient for smooth and responsive navigation. During the experiments, a human safety operator monitored the model in real time and manually intervened to halt the system whenever unsafe behaviors occurred.

Baselines and Metrics. We select the established RL/IL models from MetaUrban [21] as our primary baselines, as they represent the standard state-of-the-art on the two benchmarks. We evaluate the models using the following metrics: Success Rate (SR) and Success weighted by Path Length (SPL) [52] to evaluate both the effectiveness and efficiency of the agent's navigation; the Cumulative Cost (Cost) [53] evaluates the agent's ability to avoid collisions, while the Social Navigation Score (SNS) [54] quantifies the model's compliance with social navigation standards.

Method	Observation	PointNav						SocialNav					
		Test			Unseen			Test			Unseen		
		SR↑	SPL↑	Cost↓	SR↑	SPL↑	Cost↓	SR↑	SNS↑	Cost↓	SR↑	SNS↑	Cost↓
PPO [46]	LiDAR	66%	0.64	0.51	49%	0.45	0.78	34%	0.64	0.66	24%	0.57	0.51
PPO-Lag [47]	LiDAR	60%	0.58	**0.41**	60%	0.57	**0.53**	17%	0.51	0.33	8%	0.47	**0.50**
PPO-ET [48]	LiDAR	57%	0.53	0.47	53%	0.49	0.65	5%	0.52	**0.26**	2%	0.50	0.62
IQL [24]	LiDAR	36%	0.33	0.49	30%	0.27	0.63	36%	0.67	0.39	27%	0.62	3.05
TD3+BC [49]	LiDAR	29%	0.28	0.77	20%	0.20	1.16	26%	0.61	0.62	32%	0.64	1.53
BC [50]	LiDAR	36%	0.28	0.83	32%	0.26	1.15	28%	0.56	1.23	18%	0.54	0.58
GAIL [51]	LiDAR	47%	0.36	1.05	40%	0.32	1.46	34%	0.63	0.71	28%	0.61	0.67
Ours	RGB	**94%**	**0.91**	0.94	**97%**	**0.95**	0.84	**91%**	**0.87**	0.81	**88%**	**0.85**	0.82

TABLE I: **Benchmarks.** The benchmark of PointNav and SocialNav tasks on the MetaUrban-12K dataset. We compare our method with seven strong baselines with LiDAR observation. The **best** and the second best results are denoted by **bold** and underline, respectively.

B. Quantitative Experiments

As shown in Table I, our method significantly outperforms all baseline methods on both PointNav and SocialNav tasks in terms of SR and SPL/SNS. Specifically, our model achieves 94% SR and 0.91 SPL on the PointNav test set, and 97% SR and 0.95 SPL on the unseen set, surpassing baselines by more than 25% in SR while demonstrating strong generalization capability in unseen urban environments. The superior performance in unseen scenarios with unusual object placement and sidewalk layout indicates that our approach captures systematic structural and geometric priors of urban navigation, enabling highly effective and efficient goal-reaching behavior.

In the SocialNav task, UrbanVLA attains a high Social Navigation Score (SNS) of 0.87 on the test set and 0.85 on the unseen set, significantly exceeding all LiDAR-based baselines. This suggests that our model not only avoids collisions effectively, but also adheres to social norms, such as maintaining comfortable distances and yielding to pedestrians, even though relying solely on RGB inputs.

Although our method yields a relatively higher Cost compared to some baselines (*e.g.* PPO-Lag), this is reasonable given the substantially higher success rates, which indicate longer traveling distance and higher probability to encounter obstacles. Moreover, learning obstacle avoidance using only the RGB input is considerably more challenging than using LiDAR. Thus, achieving such high success rates while maintaining a bounded cost reflects a strong inherent capability to avoid obstacles and navigate socially.

In summary, our method exhibits superior performance in navigation efficiency, success rate, generalization, and social compliance, validating its effectiveness for complex urban navigation tasks.

C. Qualitative Results

We evaluate UrbanVLA in several representative and challenging real-world scenarios, including overpass crossing, pedestrian interaction, street turning, and obstacle avoidance on trajectories greater than 500 meters. As illustrated in Figure 4, the results highlight UrbanVLA 's ability to operate in large-scale, dynamic, and unstructured urban environments. In outdoor scenes, the system is able to produce stable navigation trajectories and follow designated routes while adapting to variations in illumination, weather, and even nighttime conditions. Moreover, the examples suggest that UrbanVLA can effectively align high-level navigation instructions with visual observations, enabling correct turns at intersections, successful navigation across overpasses, and adaptation to diverse route structures. In addition, we observe that UrbanVLA is capable of avoiding static and dynamic obstacles in these scenarios, allowing the maintenance of reasonable distances to pedestrians, and bypassing unexpected objects. Together, these qualitative results indicate that UrbanVLA can integrate high-level route priors with perception to achieve reliable and socially compliant navigation in complex urban settings in the real world.

HTL Configuration	Test		Real World
	SR↑	Cost↓	RC↑
Ours-SFT w/o HTL	**93%**	**0.81**	42%
Ours-SFT w/ HTL	89%	0.86	**100%**

TABLE II: **Comparison of HTL configurations.** We compare the performance of UrbanVLA after the SFT stage (denoted as Ours-SFT), with or without HTL. For simulation results, we report the model's performance on SocialNav in MetaUrban benchmark. For real-world, we report results on the C-shaped sidewalk, averaged over ten trials.

D. Ablation Study

Effectiveness of HTL algorithm. We evaluate the effectiveness of the HTL algorithm in both simulation and real world settings. In simulation, we evaluate using the SocialNav test set. For real-world evaluation, we design a 60 meters C-shaped sidewalk (as demonstrated in Figure 4) with a width of nearly 8 meters, where the robot must gradually turn right and then proceed straight. The route information from the navigation tools represents this curve roughly as a polyline and exhibits an error of approximately 10 meters from the center line of the route in the real world. We use Route Completion (RC) [55] to evaluate real-world performance, which refers to the proportion of the completed route out of the whole route, and we consider the scenario to terminate when the agent deviates from the route corridor (*i.e.* crashes into non-walkable area aside the route).

Fig. 4: **Visualization of qualitative experiment results in real-world scenarios.** We show four critical scenarios in real-world evaluation of UrbanVLA: overpass crossing, street turning, pedestrian crossing, and obstacle avoidance. UrbanVLA generates executive and reasonable trajectories shown by the blue trajectories plotted in the first person view (FPV).

Training stage	Test		Unseen	
	SR↑	Cost↓	SR↑	Cost↓
Ours-SFT	89%	0.86	82%	0.98
Ours	**91%**	**0.80**	**88%**	**0.82**

TABLE III: **Comparison across two training stages.** We compare the result of UrbanVLA before and after the RFT stage on SocialNav in MetaUrban benchmark.

The results in Table II show that ablating HTL leads to a slight improvement in simulation; however, RC drops dramatically in real-world experiments. We observe that most failure cases are caused by the agent repeatedly attempting to reach the shifted goal point, resulting in route deviation. This suggests that without HTL, UrbanVLA tends to overfit the route instructions, significantly reducing its robustness to noisy route information in real-world scenarios and limiting its scalability.

Effectiveness of Reinforcement Learning. We evaluate UrbanVLA under two training strategies, SFT alone and SFT combined with RFT, on the SocialNav benchmark of *MetaUrban-test* and *MetaUrban-unseen*. The results in Table III show that after the RFT stage, UrbanVLA consistently outperforms the SFT-only model, with improved SR and reduced Cost. In particular, the performance gain is larger on the unseen benchmark, with a $+6\%$ increase in SR and a -0.16 decrease in Cost, indicating the enhanced safety and generalizability of UrbanVLA after the RFT stage. We attribute this improvement to the safety-centric reward design, and the increased diversity introduced by real-world teleoperation data.

V. Conclusion

We present a route-conditioned vision-language-action framework UrbanVLA for urban micromobility, which integrates the noisy navigation-tool routes with on-board vision to enable scalable and reliable long-horizon navigation. The model is trained through SFT on simulation-based and web video–parsed trajectories, followed by RFT with a small real aggregated dataset. The proposed approach not only improves obstacle avoidance and social compliance, but also establishes a practical framework for the deployment of embodied agents in open urban areas. Future work will explore broader multimodal cues and further improve adaptability to diverse and dynamic urban contexts.

References

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

[2] G. Oeschger, P. Carroll, and B. Caulfield, "Micromobility and public transport integration: The current state of knowledge," *Transportation Research Part D: Transport and Environment*, vol. 89, p. 102628, 2020. 1

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016. 1, 2

[4] R. W. Wolcott and R. M. Eustice, "Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 292–319, 2017. 1, 2

[5] A. Y. Hata and D. F. Wolf, "Road marking detection using lidar reflective intensity data and its application to vehicle localization," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 18–29, 2016. 1

[6] A. Schlichting and C. Brenner, "Localization using automotive laser scanners and local pattern matching," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, no. 1, 2016, pp. 143–150. 1

[7] X. Liu, J. Li, Y. Jiang, N. Sujay, Z. Yang, J. Zhang, J. Abanes, J. Zhang, and C. Feng, "Citywalker: Learning embodied urban navigation from web-scale videos," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 6875–6885. 1, 2

[8] P. Roth, J. Nubert, F. Yang, M. Mittal, and M. Hutter, "Viplanner: Visual semantic imperative learning for local navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5243–5249. 1

[9] F. Yang, C. Wang, C. Cadena, and M. Hutter, "iplanner: Imperative path planning," *ArXiv*, vol. abs/2302.11434, 2023. 1

[10] J. Zhang, L. Dai, F. Meng, Q. Fan, X. Chen, K. Xu, and H. Wang, "3d-aware object goal navigation via simultaneous exploration and identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6672–6682. 1

[11] H. He, Y. Ma, W. Wu, and B. Zhou, "From seeing to experiencing: Scaling navigation foundation models with reinforcement learning," *arXiv preprint arXiv:2507.22028*, 2025. 1

[12] A. Zhang, H. Sikchi, A. Zhang, and J. Biswas, "Creste: Scalable mapless navigation with internet scale priors and counterfactual guidance," 2025. 1

[13] W. Wu, H. He, C. Zhang, J. He, S. Z. Zhao, R. Gong, Q. Li, and B. Zhou, "Towards autonomous micromobility through scalable urban simulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 1

[14] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "Navid: Video-based vlm plans the next step for vision-and-language navigation," *Robotics: Science and Systems*, 2024. 1, 2

[15] A.-C. Cheng, Y. Ji, Z. Yang, X. Zou, J. Kautz, E. Biyik, H. Yin, S. Liu, and X. Wang, "Navila: Legged robot vision-language-action model for navigation," in *RSS*, 2025. 1, 2

[16] J. Zhang, K. Wang, S. Wang, M. Li, H. Liu, S. Wei, Z. Wang, Z. Zhang, and H. Wang, "Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks," *Robotics: Science and Systems*, 2025. 1, 2

[17] M. Wei, C. Wan, X. Yu, T. Wang, Y. Yang, X. Mao, C. Zhu, W. Cai, H. Wang, Y. Chen *et al.*, "Streamvln: Streaming vision-and-language navigation via slowfast context modeling," *arXiv preprint arXiv:2507.05240*, 2025. 1, 2

[18] S. Wang, J. Zhang, M. Li, J. Liu, A. Li, K. Wu, F. Zhong, J. Yu, Z. Zhang, and H. Wang, "Trackvla: Embodied visual tracking in the wild," *arXiv pre-print*, 2025. 1, 2, 3

[19] J. Zhang, A. Li, Y. Qi, M. Li, J. Liu, S. Wang, H. Liu, G. Zhou, Y. Wu, X. Li, Y. Fan, W. Li, Z. Chen, F. Gao, Q. Wu, Z. Zhang, and H. Wang, "Embodied navigation foundation model," 2025. 1, 2, 3, 4, 5

[20] N. Hirose, C. Glossop, D. Shah, and S. Levine, "Omnivla: An omni-modal vision-language-action model for robot navigation," 2025. 1, 2

[21] W. Wu, H. He, J. He, Y. Wang, C. Duan, Z. Liu, Q. Li, and B. Zhou, "Metaurban: An embodied ai simulation platform for urban micromobility," *ICLR*, 2025. 2, 4, 5

[22] Z. Li, C. Li, X. Mao, S. Lin, M. Li, S. Zhao, Z. Xu, X. Li, Y. Feng, J. Sun, Z. Li, F. Zhang, J. Ai, Z. Wang, Y. Wu, T. He, J. Pang, Y. Qiao, Y. Jia, and K. Zhang, "Sekai: A video dataset towards world exploration," *arXiv preprint arXiv:2506.15675*, 2025. 2, 5

[23] X. Shen, Y. Xiong, C. Zhao, L. Wu, J. Chen, C. Zhu, Z. Liu, F. Xiao, B. Varadarajan, F. Bordes *et al.*, "Longvu: Spatiotemporal adaptive compression for long video-language understanding," *arXiv preprint arXiv:2410.17434*, 2024. 2, 3, 5

[24] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *International Conference on Learning Representations*, 2022. 2, 3, 4, 5, 6

[25] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Proc. Robotics: Science and Systems (RSS)*, 2014. 2

[26] M. Schreiber, C. Knöppel, and C. Stiller, "Laneloc: Lane marking based localization using highly accurate maps," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 449–454. 2

[27] Y. Qiao, W. Lyu, H. Wang, Z. Wang, Z. Li, Y. Zhang, M. Tan, and Q. Wu, "Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 6710–6717. 2

[28] Y. Chen, S. Tian, S. Liu, Y. Zhou, H. Li, and D. Zhao, "Conrft: A reinforced fine-tuning method for vla models via consistency policy," *arXiv preprint arXiv:2502.05450*, 2025. 2

[29] J. Liu, F. Gao, B. Wei, X. Chen, Q. Liao, Y. Wu, C. Yu, and Y. Wang, "What can rl bring to vla generalization? an empirical study," *arXiv preprint arXiv:2505.19789*, 2025. 2

[30] D. Huang, Z. Fang, T. Zhang, Y. Li, L. Zhao, and C. Xia, "Co-rft: Efficient fine-tuning of vision-language-action models through chunked offline reinforcement learning," *arXiv preprint arXiv:2508.02219*, 2025. 2, 4

[31] H. Zhang, S. Zhang, J. Jin, Q. Zeng, Y. Qiao, H. Lu, and D. Wang, "Balancing signal and variance: Adaptive offline rl post-training for vla flow models," *arXiv preprint arXiv:2509.04063*, 2025. 2

[32] Z. Qi, Z. Zhang, Y. Yu, J. Wang, and H. Zhao, "Vln-r1: Vision-language navigation via reinforcement fine-tuning," *arXiv preprint arXiv:2506.17221*, 2025. 2

[33] C. Gao, L. Jin, X. Peng, J. Zhang, Y. Deng, A. Li, H. Wang, and S. Liu, "Octonav: Towards generalist embodied navigation," *arXiv preprint arXiv:2506.09839*, 2025. 2

[34] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *NeurIPS*, 2023. 3, 4

[35] Y. Li, C. Wang, and J. Jia, "Llama-vid: An image is worth 2 tokens in large language models," *arXiv preprint arXiv:2311.17043*, 2023. 3

[36] AutoNavi, "Amap (gaode) maps," [Online]. Available: https://lbs.amap.com, 2023, accessed: Sep. 15, 2025. 3, 5

[37] S. Tong, E. Brown, P. Wu, S. Woo, M. Middepogu, S. C. Akula, J. Yang, S. Yang, A. Iyer, X. Pan, A. Wang, R. Fergus, Y. LeCun, and S. Xie, "Cambrian-1: A fully open, vision-centric exploration of multimodal llms," 2024. 3

[38] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024. 3

[39] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023. 3

[40] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 11 975–11 986. 3

[41] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, and Z. Fan, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024. 4

[42] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," *Robotics research*, pp. 3–19, 2011. 4

[43] B. de Wilde, A. ter Mors, and C. Witteveen, "Push and rotate: a complete multi-agent pathfinding algorithm," *J. Artif. Intell. Res.*, vol. 51, pp. 443–492, 2014. 4

[44] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964. 4

[45] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," 2020. 5

[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. 6

[47] S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau, "Benchmarking batch deep reinforcement learning algorithms," 2019. 6

[48] H. Sun, Z. Xu, M. Fang, Z. Peng, J. Guo, B. Dai, and B. Zhou, "Safe exploration by solving early terminated mdp," 2021. 6

[49] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," 2021. 6

[50] M. Bain and C. Sammut, "A framework for behavioural cloning," in *Machine Intelligence 15*, 1995. 6

[51] J. Ho and S. Ermon, "Generative adversarial imitation learning," 2016. 6

[52] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018. 5

[53] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 3461–3475, 2022. 5

[54] M. Deitke, D. Batra, Y. Bisk, T. Campari, A. X. Chang, D. S. Chaplot, C. Chen, C. P. D'Arpino, K. Ehsani, A. Farhadi *et al.*, "Retrospectives on the embodied ai workshop," *arXiv preprint arXiv:2210.06849*, 2022. 5

[55] H. Zhou, W. Cao, A. Sui, and Z. Bing, "What matters to enhance traffic rule compliance of imitation learning for end-to-end autonomous driving," 2024. 6