WildCity: A Real-World City-Scale Testbed for Rendering, Simulation, and Spatial Intelligence

Xiangyu Han[1,2], Mengyu Yang[2], Jiaqi Li[2], Bowen Chang[2], Ziyu Chen[4], Hexu Zhao[2], Rahul Kumar Agrawal[1], Anthony Rodriguez[1], Rajani Acharya[1], Fiona Hua[1], Marco Pavone[3,4], Chen Feng[2†], and Yiming Li[2,3†]

[1] May Mobility, Ann Arbor, MI, USA
[2] New York University, New York, NY, USA
[3] NVIDIA, Santa Clara, CA, USA
[4] Stanford University, Stanford, CA, USA
https://han-xiangyu.github.io/Wild-City/

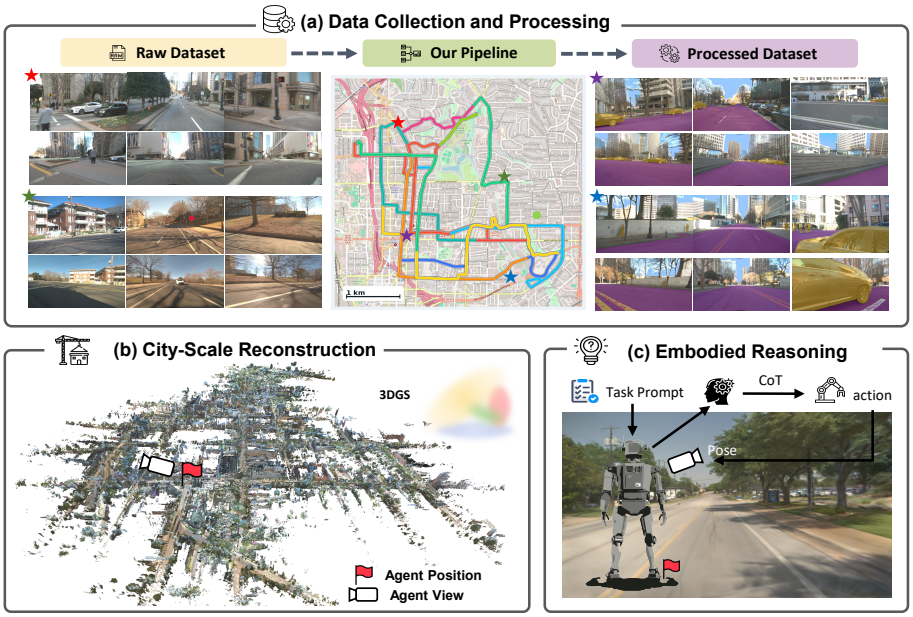


Fig. 1: Overview of *WildCity*. It contains 18 long-horizon trajectories over 1,500 km across six real-world cities. We process raw autonomous-fleet logs into multimodal data for city-scale reconstruction, simulation, and closed-loop embodied reasoning.

Abstract. Humans can navigate an unfamiliar city and gradually form a coherent spatial mental map spanning tens of square kilometers. Can AI build spatial representations at a comparable scale? Although recent foundation models have advanced scene reconstruction and embodied intelligence, scaling to entire cities remains an open challenge, primarily

† Corresponding authors.

due to the lack of city-scale data. To bridge the gap, we introduce **WildCity**, a real-world multimodal dataset collected by autonomous fleets traversing complex urban environments. Our dataset includes **18 trajectories**, each averaging **83.7 kilometers** in length, and preserves the core challenges of in-the-wild perception, *e.g.*, dynamic objects, lighting variations, and imperfect camera poses. We further establish an urban-tailored reconstruction baseline and convert the reconstructed environments into a closed-loop simulator. Beyond the dataset and baseline, we systematically analyze the key challenges on the path to simulation-ready urban digital twins: **scalability**, **extrapolation**, and **uncertainty**. Ultimately, WildCity aims to catalyze progress not only in city-scale rendering, but more broadly in the pursuit of AI that can perceive, remember, and reason across space at a scale comparable to human cognition.

Keywords: Neural Rendering · Dataset and Benchmark · Spatial AI

1 Introduction

Humans can wander for hours through the streets of a city—turning down narrow alleys, crossing wide plazas, catching glimpses of a distant tower between buildings—and still, over time, piece together a coherent *spatial mental map* spanning tens of square kilometers. Such an internal representation of large-scale space can enable self-localization, long-term memory, and efficient planning [1, 7, 23]. This leads to a question: *Can AI, like a human, internalize the structure of an entire city from visual observations and use that knowledge to reason, plan, and act?*

Recent advances in foundation models, *e.g.*, vision-language models (VLMs), have improved embodied intelligence across both virtual and real-world spatial environments—enabling tasks such as visual navigation [13], vision-language navigation [50], and active situated reasoning [47]. Yet these capabilities are typically demonstrated in small-scale scenarios—a single room, a synthetic apartment, or a city block. When scaling up spatial coverage or video duration, current models struggle to maintain spatial coherence and reason over long-range dependencies [44]. *This reveals a critical gap between the effortless scalability of human spatial cognition and the brittle, small-world reasoning of foundation models.*

To bridge this gap, foundation models require access to continuous, long-range visual-spatial observations that capture both the scale and the complexity of real-world environments. Yet existing datasets fall short. Synthetic data offers controllability but suffers from a substantial sim-to-real gap [16], while real-world benchmarks are typically limited to short video clips of isolated urban scenes [2, 15,30]. Without large-scale real-world visual-spatial data, it remains infeasible to build photorealistic city simulators—digital environments that faithfully mirror the complexity of real cities—and, in turn, to use such simulators for training and evaluating embodied agents powered by foundation models at scale.

In this work, we introduce **WildCity**, a real-world dataset and testbed for city-scale spatial intelligence. Collected by autonomous vehicle fleets traversing complex urban environments, WildCity provides continuous, surround-view

multimodal data with city-scale spatial coverage (*over 1,500 km of traversed roads*), diverse urban scenes (*covering distinct functional zones*), and long sensory streams (*averaging 2.5 hours per log*). In addition to its large scale, our dataset features a number of in-the-wild challenges—including dynamic objects, lighting variations, motion blur, and imperfect camera poses. Most importantly, WildCity supports a range of downstream tasks: from reconstructing urban digital twins for photorealistic rendering and closed-loop simulation [12, 20, 31], to studying spatial memory, localization, perception, and reasoning with VLMs across a large city [19, 38]. In this paper, we focus on city-scale rendering as a first concrete instantiation of this broader vision.

Beyond dataset curation, we build a strong baseline tailored to the challenges of large-scale, noisy, and unbounded scenes. We reconstruct urban environments at scale and integrate the resulting models into a closed-loop simulator, demonstrating their utility for downstream embodied tasks. Moreover, we conduct a systematic analysis of the key obstacles in real-world city-scale reconstruction: **performance scalability**, **view extrapolation**, and **data uncertainty**. This reveals fundamental limitations of existing methods and points toward promising directions for building simulation-ready urban digital twins and, more broadly, advancing city-scale spatial intelligence. Our main contributions are threefold:

- **A city-scale real-world dataset.** We introduce WildCity, a large-scale benchmark featuring various in-the-wild challenges—dynamic objects, lighting variations, and imperfect observations—making it a realistic platform for reconstruction, rendering, and embodied AI at city scale.
- **An urban-tailored reconstruction baseline.** We establish a baseline specifically designed for large, noisy, and unbounded street scenes, and integrate the resulting reconstructions into a closed-loop simulator to support downstream embodied tasks such as end-to-end autonomous driving.
- **A systematic analysis of key challenges.** We identify fundamental limitations of existing pipelines, *i.e.*, scalability, extrapolation, and uncertainty, and outline directions toward simulation-ready urban digital twins.

2 Related Works

Large-Scale 3D Reconstruction. Scaling neural rendering from objects and indoor scenes to unbounded outdoor environments remains a significant challenge [20, 31]. Early NeRF-based methods such as Block-NeRF [31] and Mega-NeRF [33] improve scalability by decomposing large scenes into spatial submodels, but their practical utility remains limited by high training and rendering costs. The emergence of 3D Gaussian Splatting (3DGS) [11] has substantially improved the efficiency of large-scale reconstruction. VastGaussian [18], CityGaussian [20, 21], UrbanGS [14], and FlashGS [8] further enhance optimization and rendering efficiency for large scenes, though they primarily target aerial views or scenes below the scale of entire cities [18, 20, 21]. In contrast, real-world street-view city-scale reconstruction involves long and narrow trajectories, limited viewpoint overlap, and substantial sensor noise [10, 43]. More recently, feed-forward

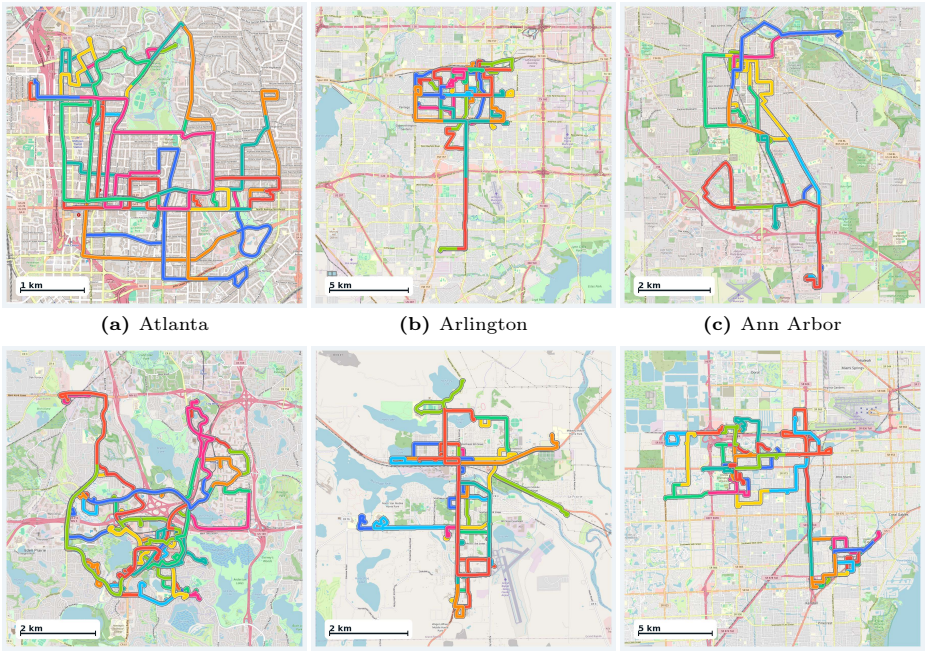
Dataset	Scale	Real	#Cities	Coverage (Avg / Total km)	Surround-view	Continuous Coverage
MatrixCity [16]	City	✗	2	– / 75.3	✓	✓
SS3DM [10]	Route	✗	8	0.48 / 13.4	✓	✓
PandaSet [41]	Clip	✓	1	0.06 / 6.5	✓	✗
Argoverse 2 [39]	Clip	✓	6	0.12 / 120	✓	✗
WayveScenes101 [51]	Clip	✓	1	0.16 / 16.0	✓	✗
nuScenes [2]	Clip	✓	2	0.18 / 100	✓	✗
Waymo Open [30]	Clip	✓	3	0.20 / 100	✓	✗
KITTI-360 [17]	Route	✓	1	6.7 / 74	∼	✓
Oxford RobotCar [22]	Route	✓	1	10.0 / 1010	∼	✓
Block-NeRF [31]	Block	✓	1	1.08 / 1.08	✓	✓
Oxford Spires [32]	Block	✓	1	1.25 / 30.0	✓	✓
H-3DGS [12]	Block	✓	1	3.02 / 9.05	✓	✓
WildCity (Ours)	**City**	✓	**6**	**83.7 / 1507.1**	✓	✓

Table 1: Comparison with existing street-view urban reconstruction datasets. WildCity is the only real-world dataset that jointly provides multi-city coverage, long continuous traversals, surround-view sensing, and city-scale route length. "∼" indicates partial or limited support.

3D reconstruction methods such as DUSt3R [36] and VGGT [34], together with their longer-horizon extensions [4, 6, 35], have emerged as a promising alternative by predicting pixel-aligned geometry from learned 3D priors. However, they remain limited in multi-kilometer urban settings due to long-horizon pose drift and the restricted fidelity of sparse point-based representations [4, 34–36].

City-Scale Street-View Datasets. Existing datasets for urban scene reconstruction can be broadly grouped into synthetic city-scale benchmarks, real-world clip-level driving datasets, and real-world route- or block-level reconstruction datasets (see Tab. 1). Synthetic datasets such as MatrixCity [16] and SS3DM [10] provide broad coverage and clean annotations through controllable simulation, but suffer from sim-to-real gaps [25, 28, 45]. Real-world driving datasets such as PandaSet [41], Argoverse 2 [39], WayveScenes101 [51], nuScenes [2], and Waymo [30] contain surround-view observations in real urban environments, but are primarily organized as short clips for perception tasks and therefore lack the continuous city-scale coverage. Route-level datasets such as KITTI-360 [17] and Oxford RobotCar [22] offer longer traversals, yet remain limited in geographic diversity, sensor coverage, or benchmark design for neural rendering. Reconstruction-oriented datasets such as Block-NeRF [31], Oxford Spires [32], and H-3DGS [12] move closer to large-scale scene reconstruction, but their geographic diversity and spatial coverage remain limited. *Hence, a benchmark that jointly provides real-world data, surround-view sensing, continuous coverage, and street-view imagery across multiple cities is still missing.*

Embodied Spatial Reasoning and Navigation. Researchers have long studied spatial reasoning and long-horizon decision-making for embodied agents. Early benchmarks such as MultiON [38] and Habitat-Web [26] emphasize memory, object-centric exploration, and large-scale embodied interaction in photorealistic environments, primarily indoors. More recent works expand toward richer multimodal and longer-horizon tasks. GOAT-Bench [13] studies lifelong multimodal navigation over sequential open-vocabulary goals, while LH-VLN [29] targets long-horizon vision-language navigation with multi-stage planning. City-



(a) Atlanta **(b)** Arlington **(c)** Ann Arbor

(d) Eden Prairie **(e)** Grand Rapids **(f)** Miami

Fig. 2: Geographic coverage of *WildCity*. We visualize trajectories collected across six U.S. cities, with different colors indicating 5-km segments used for standardized data organization and reconstruction. The routes cover diverse urban layouts, including downtown grids, residential neighborhoods, arterial roads, and suburban corridors.

Walker [19] further moves toward urban embodied navigation by learning from web-scale city walking and driving videos. In autonomous driving, UniDrive-WM [42] couple visual understanding, and future scene generation for decision-making. However, these works still lack real-world, city-scale digital twins for interactive spatial reasoning and evaluation. In contrast, WildCity provides the sensory foundation for constructing such twins, enabling future research on city-scale spatial memory, localization, reasoning, and agent-based evaluation.

3 WildCity Dataset

3.1 Sensor Setup

Our data acquisition platform is equipped with a multi-modal sensor suite comprising a roof-mounted LiDAR, 6 surround-view RGB cameras, an IMU, and a GPS receiver. To ensure comprehensive spatial coverage, the vision system utilizes three narrow-angle cameras for forward-facing views and three wide-angle cameras for lateral and rear views. We provide rigorous calibration, including extrinsic parameters mapping each sensor to the ego-vehicle coordinate frame. Camera intrinsics and lens distortion coefficients are calibrated via AprilCal [27].

Fig. 3: Semantic masks for region-aware reconstruction. We generate ground, sky, and dynamic-object masks to support road regularization, sky modeling, and moving-object filtering, respectively. For movable categories, 3D tracking cuboids distinguish truly dynamic objects from stationary instances: red boxes indicate objects masked out, while green boxes indicate stationary objects preserved for reconstruction.

3.2 Data Collection

Our fleet currently operates in *six U.S. cities: Atlanta, Arlington, Ann Arbor, Eden Prairie, Grand Rapids and Miami*, as shown in Figure 2. These cities exhibit substantial diversity in urban style, geographic layout, and climate conditions. Across the dataset, the fleet traverses a broad range of street-level scenarios, including dense intersections, narrow local streets, multi-lane arterial roads, and high-speed parkways, under varying traffic patterns and scene complexity. The logs are collected on different days during regular daytime fleet operations, naturally introducing real-world variation in illumination, weather, appearance, and traffic dynamics. As a result, the dataset captures the uncertainty and noise inherent in real urban data, including dynamic objects, appearance changes, and imperfect pose estimates, constructing a challenging but valuable benchmark.

3.3 Data Processing

The ego poses are initialized from the onboard SLAM system and further refined using GPS signals to improve global consistency, reduce long-range drift, and enforce loop closure when revisiting previously traversed areas. As the raw sensors operate at different frequencies, we extract their measurements directly from the raw logs and align them using the original timestamps instead of resampling them to a fixed frequency. To further improve accuracy, we apply motion compensation to each LiDAR sweep to obtain more accurately aligned geometry. In addition, we provide semantic masks to support broader downstream usage.

Table 2: Quantitative validation of semantic masks. We evaluate the automatically generated dynamic-object, sky, and ground masks against 100 manually annotated images across four cities. The high overall mIoU indicates that our mask generation pipeline provides reliable region-level supervision for downstream reconstruction.

City	# of Images	Dynamic	Sky	Ground	mIoU
Ann Arbor	20	83.60	96.50	95.31	91.80
Atlanta	30	88.57	84.27	93.86	88.90
Arlington	25	88.95	93.94	94.14	92.34
Eden Prairie	25	90.11	87.62	93.95	90.56
Overall	100	88.48	92.01	94.24	91.58

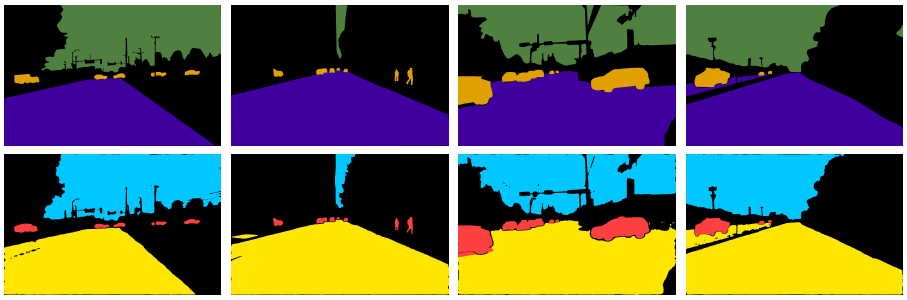


Fig. 4: Qualitative validation of semantic masks. We compare manual annotations (top) with automatically generated masks (bottom) for dynamic objects, sky, and ground regions. The visual agreement complements the quantitative IoU results in Table 2 and supports the use of automatic masks for region-aware reconstruction.

Specifically, we generate masks for ground, sky, and potentially movable objects (e.g., pedestrians, vehicles, and bicycles) using SAM3 [3] with text prompts. We then leverage 3D tracking cuboids from our onboard detection annotations to filter out stationary instances of typically dynamic categories, such as parked vehicles and standing pedestrians. As illustrated in Figure 3, this process produces region masks that support ground modeling, sky separation, and moving-object filtering for reconstruction. We further validate the generated masks quantitatively and qualitatively in Table 2 and Figure 4, showing that they provide reliable region-level annotations for reconstruction and other tasks that benefit from coarse semantic regions.

3.4 Data Organization

To make long-horizon city logs tractable for reconstruction and benchmarking, we adopt a standardized trajectory discretization and segmentation scheme. We first spatially subsample keyframes at a fixed **spacing of 0.5 m** to balance reconstruction fidelity and training efficiency. Each log is then partitioned into contiguous **5 km chunks** for convenient usage and reproducible comparisons. On top of these chunks, we provide sub-trajectories at multiple lengths **[50 m, 250 m, 500 m, 1 km, 2.5 km, and 5 km]** to support evaluation under increasing spatial scale; shorter segments emphasize local consistency while longer segments stress scalability and drift accumulation. Since all poses are aligned

Table 3: Key dataset and reconstruction statistics. We summarize the sensor configuration, total data scale, and computational footprint required for reconstruction at increasing trajectory lengths.

Statistic	Value
Total keyframes	3.01M
RGB cameras / resolution	3 @ 1440×928 and 3 @ 1240×728
LiDAR points per frame	80k original / 15k downsampled
Camera / LiDAR / GPS / IMU rate	10 / 10 / 10 / 100 Hz
Avg. GS @ 0.5km / 1km / 2.5km / 5km	6M / 12M / 30M / 60M
Min. VRAM @ 0.5km / 1km / 2.5km / 5km	1×24G / 1×40G / 1×80G / 2×80G

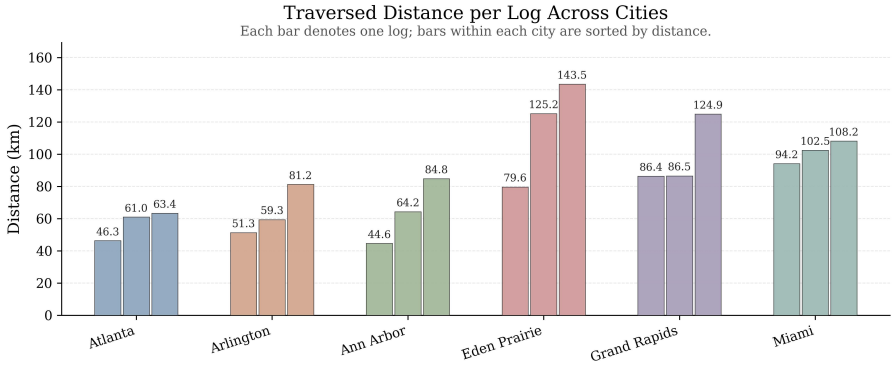


Fig. 5: Log-level route coverage. Each bar represents one continuous driving log, with logs within each city sorted by traversed distance. The distribution highlights the long-horizon, multi-city nature of *WildCity*, in contrast to isolated short-clip datasets.

in a shared city-level coordinate system, segments can be directly concatenated into longer routes without additional registration.

3.5 Data Statistics

As summarized in Table 3, *WildCity* provides **3.01M keyframes** with synchronized surround-view RGB, LiDAR, GPS, and IMU measurements, along with sensor rates, point-cloud density, and reconstruction resource references to facilitate reproducible training and evaluation. At the route level, the spatial coverage is about **40.18 km**^{2} per city, spanning a mixture of downtown regions, residential streets, arterial roads, and other urban corridors. Each driving log lasts about **2.5 hours** and covers **83.7 km** of route length, capturing long continuous traversals rather than isolated short clips. In total, we collect three logs per city, resulting in **18 logs** and **1507.1 km** of recorded driving distance across all cities. As illustrated in Figure 5, the route lengths vary across cities and logs, reflecting the natural diversity of real-world fleet operations rather than an artificially balanced collection process. This organization is particularly important for evaluating methods under realistic city-scale coverage, where both long-range continuity and cross-city diversity matter.

4 WildCity Method

4.1 3D Gaussian Splatting Preliminary

3D Gaussian Splatting (3DGS) [11] represents a scene as a set of anisotropic Gaussians $\mathcal{G} = \{g_i\}_{i=1}^{N}$, where each Gaussian is parameterized by its opacity, mean position, rotation, scale, and view-dependent color. For rendering, the Gaussians are projected to the image plane and alpha-blended in depth order,

$$C = \sum_{i \in \mathcal{N}} c_i \, \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{1}$$

where $\mathcal{N}$ denotes the set of projected Gaussians overlapping the pixel, α_i is the projected opacity of the i-th Gaussian and c_i is the view-dependent color.

4.2 Urban-tailored Large-scale Reconstruction

Rig pose optimization. Accurate poses are essential for high-quality neural reconstruction and rendering, but real-world pose estimates from raw logs are often not globally optimal. Although independent camera pose optimization is widely adopted [5], it ignores the rigid multi-camera setup at each keyframe and can lead to locally improved but globally inconsistent solutions. We therefore decompose pose optimization into two coupled components: *(1) the ego pose at each keyframe (2) the camera extrinsics relative to the ego frame.* Formally, let $\mathbf{T}_t^{\text{ego}} \in SE(3)$ denote the ego pose at keyframe t, and let $\mathbf{T}_c^{\text{rig}} \in SE(3)$ denote the extrinsic pose of camera c in the ego frame. The resulting camera pose is

$$\mathbf{T}_{t,c} = \mathbf{T}_t^{\text{ego}} \mathbf{T}_c^{\text{rig}}. \tag{2}$$

We then jointly optimize the scene parameters θ, ego poses, and rig extrinsics via

$$\min_{\theta, \{\mathbf{T}_t^{\text{ego}}\}, \{\mathbf{T}_c^{\text{rig}}\}} \sum_{t,c} \mathcal{L}_{\text{rend}}(\mathbf{I}_{t,c}, \mathcal{R}(\theta, \mathbf{T}_{t,c}) + \lambda_{pose}(\sum_{t} d(\mathbf{T}_t^{\text{ego}}, \bar{\mathbf{T}}_t^{\text{ego}}) + \sum_{c} d(\mathbf{T}_c^{\text{rig}}, \bar{\mathbf{T}}_c^{\text{rig}})), \tag{3}$$

where $\mathcal{R}$ denotes the renderer, $\mathbf{I}_{t,c}$ is the image from camera c at keyframe t, λ_{pose} represents the pose loss weight, $\bar{\mathbf{T}}_t^{\text{ego}}$ and $\bar{\mathbf{T}}_c^{\text{rig}}$ are the initial poses from localization and calibration, and $d(\cdot, \cdot)$ is a pose-distance metric on $SE(3)$.

Sky model. Following a widely used design in neural rendering, we model the sky separately instead of representing it with 3D Gaussians [5, 43]. Specifically, we use a lightweight view-dependent MLP to predict the sky color image $\mathbf{C}_{\text{sky}}$ and composite it with the rendered Gaussian image $\mathbf{C}_g$ as an infinite background:

$$\mathbf{C} = \mathbf{C}_g + (1 - \mathbf{O}_g) \, \mathbf{C}_{\text{sky}}, \tag{4}$$

where $\mathbf{O}_g$ denotes the rendered Gaussian opacity. This decouples sky appearance from scene geometry and helps reduce spurious far-depth floaters.

Ground regularization. To stabilize underconstrained road geometry, we impose priors on Gaussians identified as ground. We regularize ground height standard deviation by sampling slices along the camera-depth axis and penalizing the variation of the camera-frame vertical coordinate within each slice:

$$\mathcal{L}_{\text{dist}} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \text{Std}\big(\{\mu_{i,y}^{\text{cam}} \mid i \in \mathcal{S}_k\}\big) . \tag{5}$$

where $\mathcal{K}$ is the set of local ground Gaussians, $\mu_{i,y}^{\text{cam}}$ is the position of splats in camera y axis. We further encourage ground Gaussians to be vertically aligned and sufficiently opaque:

$$\mathcal{L}_{\text{align}} = \frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} \left(1 - \left|a_{i,y}^{\text{cam}}\right|\right), \qquad \mathcal{L}_{\text{opa}} = \frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} (1 - o_i)^2 . \tag{6}$$

The final loss is

$$\mathcal{L}_{\text{ground}} = \lambda_{\text{dist}}\mathcal{L}_{\text{dist}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{opa}}\mathcal{L}_{\text{opa}}, \tag{7}$$

where $\mathcal{G}$ is the set of ground Gaussians, $\mathcal{S}_k$ is the k-th sampled depth slice, $a_{i,y}^{\text{cam}}$ the shortest axis component of camera vertical axis and o_i the opacity.

Extrapolated view post-repair. Extrapolated rendering often suffers from broken geometry and floating artifacts [9]. We therefore adopt a progressive render-repair-augment scheme with Difix3D+ [40]. Given Gaussian parameters $\theta^{(m)}$ and a sampled extrapolated pose $\tilde{\mathbf{T}}$, we first render

$$\tilde{\mathbf{I}}^{(m)} = \mathcal{R}(\theta^{(m)}, \tilde{\mathbf{T}}), \tag{8}$$

then repair it with Difix3D+,

$$\hat{\mathbf{I}}^{(m)} = \mathcal{F}_{\text{Difix}}(\tilde{\mathbf{I}}^{(m)}, \mathbf{I}_{\text{ref}}, \textit{prompt}), \tag{9}$$

and add the repaired supervision back to the training set:

$$\mathcal{D}_{m+1} = \mathcal{D}_m \cup \{(\tilde{\mathbf{T}}, \hat{\mathbf{I}}^{(m)})\}. \tag{10}$$

The Gaussians are then re-optimized on the augmented set. Repeating this process progressively repairs geometric artifacts under extrapolated viewpoints.

Multi-GPU training. As scene scale increases, the number of Gaussians grows rapidly and training becomes GPU-memory constrained. Following GrendelGS [49] and GSplat [46], we enable multi-GPU training by sharding Gaussian parameters across devices and synchronizing the necessary statistics and gradients, allowing us to scale to billion-level primitives with bounded per-GPU memory.

4.3 Closed-loop Simulation in City Digital Twin

Beyond reconstruction, *WildCity* enables embodied interaction in a city digital twin, as shown in Figure 6. A Vision-Language-Action model iteratively predicts the next target pose from the current observation and task, and receives the rendered view at that pose. Using Alpamayo 1 [24] as a proof of concept, we observe smooth navigation and plausible multi-step reasoning in the reconstructed urban environment. This highlights the broader value of WildCity as a foundation for studying large-scale spatial intelligence in realistic cities.

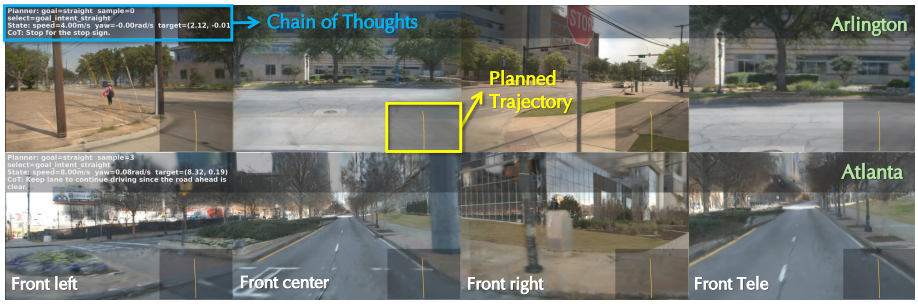


Fig. 6: Agents closed-loop simulation. As proof of concept, Alpamayo [24] is integrated into our closed-loop simulator and able to perceive, reason and act interactively.

5 Experiments

5.1 Evaluation Protocol

We evaluate 3D reconstruction on *WildCity* across different spatial scales to assess state-of-the-art methods under both local and long-horizon settings. We use two representative sequences captured with the same sensor setup: Ann Arbor-0.5k and Atlanta-5k. We report PSNR, SSIM [37], and LPIPS [48] on the static region defined by semantic segmentation masks, and Depth L1 (meters) on pixels with valid LiDAR depth in the same region. For VGGT-based variants, we align the VGGT reconstruction to our metric pointcloud by optimizing a global similarity transform. Each method is trained under identical and sufficiently provisioned H200 hardware, until validation performance stabilizes when feasible.

5.2 Comparison with Baselines

Baselines. We compare our method against the following baselines: classic 3D Gaussian Splatting (3DGS) [11], the scalable hierarchical variant H-3DGS [12], and the city-scale blockwise training approach CityGaussianV2 (CityGS) [21]. We additionally include the feed-forward long-sequence geometry prior VGGT-Long [6] as a baseline for point-cloud and pose reconstruction. We further include VGGT-Long+CityGS, where we replace the default SLAM-initialized poses and sparse point clouds in the CityGS training pipeline with the aligned VGGT-Long reconstruction outputs. This baseline tests whether feed-forward priors can serve as a substitute for sensor-based slam initializations on long sequences.
Quantitative comparison. As shown in Table 4, while baselines achieve strong 2D image metrics on short sequences, they compromise the underlying 3D geometry. For instance, on the short trajectory, methods like H-3DGS and CityGS report high PSNR or SSIM but exhibit massive geometric distortions, with Depth L1 exceeding **17 m**. Our method corrects this misalignment, significantly reducing the depth error to **11.75 m** while maintaining highly competitive perceptual quality. This confirms that relying solely on spatial partitioning strategies without strong structural constraints fails to guarantee physically accurate geometry.

Table 4: Quantitative comparison with baselines on two scene scales. Ann Arbor-Small (0.5k timestamps) and Atlanta-Long (5k timestamps). Best and second-best results in each column are highlighted in green and light green, respectively.

	Ann Arbor-0.5k (0.25km)				Atlanta-5k (2.5km)			
Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**D-L1↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**D-L1↓**
3DGS [11]	20.00	0.829	0.581	32.704	19.39	0.715	0.492	15.447
H-3DGS [12]	27.59	0.849	0.318	18.450	21.40	0.747	0.380	14.368
CityGS [21]	24.41	0.856	0.409	17.053	21.27	0.762	0.536	8.349
VGGT-Long [6]	–	–	–	29.632	–	–	–	19.720
VGGT-Long+CityGS	20.98	0.663	0.535	22.695	13.40	0.666	0.745	20.934
Ours	29.99	0.917	0.240	15.158	23.14	0.799	0.477	6.622

When scaling up to the 2.5 km trajectory, our method's advantage becomes decisive, achieving the best overall performance (23.14 dB PSNR and 6.62 m D-L1). It effectively prevents the structural degradation that plagues baselines under sparse-overlap conditions, proving that optimizing image similarity alone does not yield simulation-ready 3D structures. For completeness, we provide a more comprehensive evaluation across all six cities in the supplementary material (Table 7). Furthermore, we observe that feed-forward priors suffer from compounding long-horizon drift. Attempting to substitute initialization with these drifting priors causes the optimization to collapse, dropping the PSNR to 13.40 dB. This emphasizes that current feed-forward models cannot yet replace globally consistent pose optimization for unconstrained, city-scale reconstruction.

Partitioning and hierarchy are useful for scaling to large scenes, but they do not by themselves resolve the geometric ambiguity of narrow, long street-view trajectories. Reliable city-scale reconstruction requires structural constraints and pose consistency in addition to memory-efficient scene decomposition.

5.3 Qualitative Results

Figure 7 compares renderings on in-trajectory test views and extrapolated views outside the training trajectory. (1) On in-trajectory views, classic 3DGS often over-smooths large street scenes and loses thin structures (e.g., poles and distant façades). Scalable baselines recover richer content but frequently exhibit degraded texture fidelity on ground surfaces and slender objects, often accompanied by geometric inconsistency. *Our method produces sharper textures and cleaner edges in these regions, reducing local artifacts while maintaining overall visual quality.* (2) On extrapolated views, all methods degrade, but baselines show noticeably stronger blur and floaters, leading to unstable structures. *In contrast, our results retain clearer scene layout with fewer floating artifacts, suggesting improved scene geometry and view robustness for simulation.*

(a) Evaluation views sampled *within* the training trajectory.

(b) Extrapolation views sampled *outside* the trajectory (generalization).

Fig. 7: Qualitative comparison with baselines. We compare renderings on in-trajectory test views and extrapolated views sampled at 0.5 m intervals outside the trajectory. For extrapolated views, we additionally present the results refined by diffusion-based repair module [40], demonstrating its effectiveness in further mitigating artifacts and hallucinating coherent structures in unobserved regions.

For closed-loop simulation, in-trajectory rendering quality is not sufficient: agents must query views beyond the recorded path. Simulation-ready urban digital twins therefore require stable geometry and coherent appearance under off-trajectory viewpoints.

6 Discussion

6.1 Performance Scalability

Performance scalability asks whether reconstruction quality can be maintained as scene scale increases. As shown in Figure 8, rendering quality degrades steadily as trajectory length grows, while peak VRAM usage and the number of optimized Gaussians increase accordingly. This indicates that current methods do not scale gracefully to continuous city-scale scenes: longer routes not only require more memory and model capacity, but also accumulate more appearance variation and pose inconsistency over space. The comparison with synthetic data further suggests that this degradation is substantially stronger in real-world settings.

6.2 View Extrapolation

View extrapolation measures whether a model can generalize to unobserved regions under sparse or out-of-distribution viewpoints. This is especially important for interactive city digital twins, where agents must move beyond the exact

Table 5: Component ablation. We evaluate sky modeling, rig pose optimization, and ground regularization on rendering and geometry. Notably, 2D image metrics can misalign with geometric quality: some components may slightly hurt 2D metrics yet markedly improve structural consistency and simulation usefulness (see Figure 9).

Components			Arlington (0.1km)			Atlanta (0.25km)			
Ground	Sky	Pose Opt.	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	D-L1↓
✗	✓	✓	**30.14**	**0.924**	**0.186**	**26.35**	**0.864**	**0.288**	**13.11**
✓	✗	✓	28.69	0.903	0.230	24.72	0.829	0.362	148.54
✓	✓	✗	29.33	0.913	0.199	25.51	0.843	0.320	14.29
✓	✓	✓	29.60	0.910	0.217	25.42	0.815	0.342	13.26

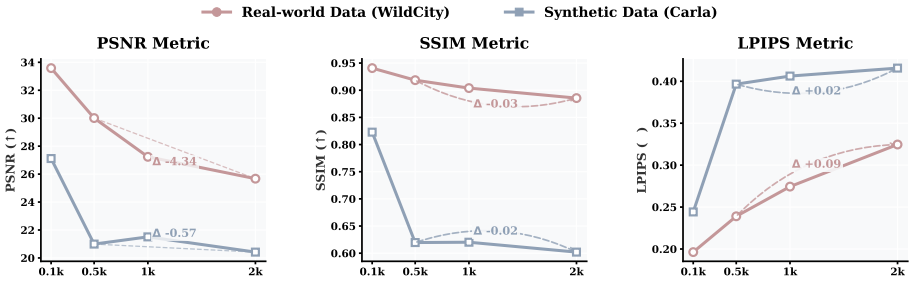


Fig. 8: Quantitative evaluation across data scales. With matched Gaussian budgets at each scale, real-world *WildCity* degrades more severely than synthetic Carla from 0.5k to 2k keyframes, indicating that city-scale reconstruction is constrained not only by capacity but also by accumulated pose, appearance, and uncertainty challenges.

recorded trajectory. As shown in Figure 7, current baselines often fail under extrapolated views, producing broken geometry, floaters, and unstable depth. Our Difix3D+ [40] based post-repair improves visual quality and partially fixes these artifacts, but it remains a costly post-hoc solution and can hallucinate content when the base reconstruction is weak. This suggests that extrapolation needs to be addressed by stronger geometric priors within the reconstruction model itself.

6.3 Data Uncertainty

Data uncertainty arises from uncontrolled factors in real-world urban driving, including dynamic objects, illumination changes, and pose noise. The ablations in Table 5 and qualitative comparisons in Figure 9 confirm that these uncertainties substantially impact reconstruction fidelity and geometric consistency. While our baseline mitigates several dominant sources, *e.g.*, a sky model for unbounded background ambiguity, rig-aware pose optimization for noisy localization, and ground regularization for weakly constrained road surfaces. The remaining errors highlight that handling real-world uncertainty remains a fundamental challenge for city-scale street-view reconstruction.

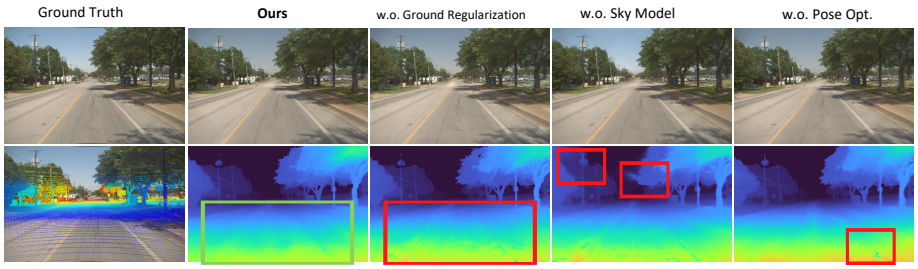


Fig. 9: Qualitative results of different components. This ablation highlights each module's role under real-world uncertainty: removing ground regularization destabilizes road geometry, removing the sky model introduces background artifacts, and removing rig pose optimization causes multi-view misalignment from pose noise.

City-scale reconstruction remains limited by three coupled factors: incomplete 3D priors, long-horizon pose inconsistency, and weak extrapolation under sparse observations. Future evaluation should therefore move beyond 2D fidelity toward geometric correctness and simulation utility.

7 Conclusion

We presented *WildCity*, a real-world city-scale dataset for street-view reconstruction and beyond, built from long-horizon surround-view RGB-LiDAR observations collected in unconstrained urban environments. On top of this dataset, we established an urban-tailored reconstruction baseline and enabled closed-loop interaction in the reconstructed city digital twin. Together, these components provide a practical testbed for studying real-world city-scale reconstruction under realistic noise, uncertainty, and long-range spatial extent.

Limitations. *WildCity* still has limitations. First, our semantic masks are automatically generated; although they achieve 91.58% mIoU on 100 manually annotated images across four cities, boundary errors and rare category mistakes may remain. Second, our GPS-anchored poses avoid unbounded horizontal drift, but residual pose error persists, especially vertically: loop-closure analysis shows sub-centimeter horizontal drift and centimeter-level vertical drift per kilometer. Improving long-horizon urban pose alignment remains important future work.

Beyond rendering. While this paper focuses on explicit reconstruction and simulation, the broader significance of *WildCity* lies in its potential to support implicit spatial understanding over long spatial and temporal horizons. Tasks such as long-form video understanding, memory, localization, and planning continue to be limited by the lack of realistic large-scale urban data. We hope *WildCity* will serve not only as a benchmark for reconstruction, but also as a foundation for future research on city-scale spatial intelligence.

Acknowledgements

We thank the May Mobility operations, data platform, and autonomy teams for their support in real-world data collection, on-site operations, and data processing infrastructure. Chen Feng was supported by NSF grant No. 2238968.

References

1. Bellmund, J.L., Gärdenfors, P., Moser, E.I., Doeller, C.F.: Navigating cognition: Spatial codes for human thinking. Science **362**(6415), eaat6766 (2018)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
3. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., et al.: Sam 3: Segment anything with concepts. arXiv preprint arXiv:2511.16719 (2025)
4. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Ttt3r: 3d reconstruction as test-time training. arXiv preprint arXiv:2509.26645 (2025)
5. Chen, Z., Yang, J., Huang, J., De Lutio, R., Esturo, J.M., Ivanovic, B., Litany, O., Gojcic, Z., Fidler, S., Pavone, M., et al.: Omnire: Omni urban scene reconstruction. arXiv preprint arXiv:2408.16760 (2024)
6. Deng, K., Ti, Z., Xu, J., Yang, J., Xie, J.: Vggt-long: Chunk it, loop it, align it–pushing vggt's limits on kilometer-scale long rgb sequences. arXiv preprint arXiv:2507.16443 (2025)
7. Epstein, R.A., Patai, E.Z., Julian, J.B., Spiers, H.J.: The cognitive map in humans: spatial navigation and beyond. Nature neuroscience **20**(11), 1504–1513 (2017)
8. Feng, G., Chen, S., Fu, R., Liao, Z., Wang, Y., Liu, T., Hu, B., Xu, L., Pei, Z., Li, H., et al.: Flashgs: Efficient 3d gaussian splatting for large-scale and high-resolution rendering. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26652–26662 (2025)
9. Han, X., Jia, Z., Li, B., Wang, Y., Ivanovic, B., You, Y., Liu, L., Wang, Y., Pavone, M., Feng, C., Li, Y.: Extrapolated urban view synthesis benchmark. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 28718–28728 (October 2025)
10. Hu, Y., Wen, K., Zhou, H., Guo, X., Liu, Y.j.: Ss3dm: benchmarking street-view surface reconstruction with a synthetic 3d mesh dataset. Advances in Neural Information Processing Systems **37**, 106649–106666 (2024)
11. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), `https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/`
12. Kerbl, B., Meuleman, A., Kopanas, G., Wimmer, M., Lanvin, A., Drettakis, G.: A hierarchical 3d gaussian representation for real-time rendering of very large datasets. ACM Transactions On Graphics (TOG) **43**(4), 1–15 (2024)
13. Khanna, M., Ramrakhya, R., Chhablani, G., Yenamandra, S., Gervet, T., Chang, M., Kira, Z., Chaplot, D.S., Batra, D., Mottaghi, R.: Goat-bench: A benchmark for multi-modal lifelong navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16373–16383 (June 2024)
14. Li, C., Zhu, H., Chen, H., Liang, X., Chen, T., Shao, S., Yang, L., Tan, H., Zhang, B.: Urbangs: A scalable and efficient architecture for geometrically accurate large-scene reconstruction. arXiv preprint arXiv:2602.02089 (2026)
15. Li, Y., Li, Z., Chen, N., Gong, M., Lyu, Z., Wang, Z., Jiang, P., Feng, C.: Multiagent multitraversal multimodal self-driving: Open mars dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22041–22051 (2024)

16. Li, Y., Jiang, L., Xu, L., Xiangli, Y., Wang, Z., Lin, D., Dai, B.: Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3205–3215 (2023)
17. Liao, Y., Xie, J., Geiger, A.: Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(3), 3292–3310 (2022)
18. Lin, J., Li, Z., Tang, X., Liu, J., Liu, S., Liu, J., Lu, Y., Wu, X., Xu, S., Yan, Y., et al.: Vastgaussian: Vast 3d gaussians for large scene reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5166–5175 (2024)
19. Liu, X., Li, J., Jiang, Y., Sujay, N., Yang, Z., Zhang, J., Abanes, J., Zhang, J., Feng, C.: Citywalker: Learning embodied urban navigation from web-scale videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6875–6885 (June 2025)
20. Liu, Y., Luo, C., Fan, L., Wang, N., Peng, J., Zhang, Z.: Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In: European Conference on Computer Vision. pp. 265–282. Springer (2025)
21. Liu, Y., Luo, C., Mao, Z., Peng, J., Zhang, Z.: Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes. arXiv preprint arXiv:2411.00771 (2024)
22. Maddern, W., Pascoe, G., Gadd, M., Barnes, D., Yeomans, B., Newman, P.: Real-time kinematic ground truth for the oxford robotcar dataset. arXiv preprint arXiv:2002.10152 (2020)
23. Maguire, E.A., Gadian, D.G., Johnsrude, I.S., Good, C.D., Ashburner, J., Frackowiak, R.S., Frith, C.D.: Navigation-related structural change in the hippocampi of taxi drivers. Proceedings of the National Academy of Sciences **97**(8), 4398–4403 (2000)
24. NVIDIA, Wang, Y., Luo, W., Bai, J., Cao, Y., Che, T., Chen, K., Chen, Y., Diamond, J., Ding, Y., Ding, W., Feng, L., Heinrich, G., Huang, J., Karkus, P., Li, B., Li, P., Lin, T.Y., Liu, D., Liu, M.Y., Liu, L., Liu, Z., Lu, J., Mao, Y., Molchanov, P., Pavao, L., Peng, Z., Ranzinger, M., Schmerling, E., Shen, S., Shi, Y., Tariq, S., Tian, R., Wekel, T., Weng, X., Xiao, T., Yang, E., Yang, X., You, Y., Zeng, X., Zhang, W., Ivanovic, B., Pavone, M.: Alpamayo-R1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. arXiv preprint arXiv:2511.00088 (2025)
25. Peng, X., Usman, B., Kaushik, N., Wang, D., Hoffman, J., Saenko, K.: Visda: A synthetic-to-real benchmark for visual domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 2021–2026 (2018)
26. Ramrakhya, R., Undersander, E., Batra, D., Das, A.: Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5173–5183 (2022)
27. Richardson, A., Strom, J., Olson, E.: Aprilcal: Assisted and repeatable camera calibration. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1814–1821 (2013). https://doi.org/10.1109/IROS.2013.6696595
28. Safaei, D., Khastgir, S., Alirezaei, M., Ploeg, J., Tong, S., Cheng, C.H., Zhao, X.: Quantifying fidelity: A decisive feature approach to comparing synthetic and real imagery. arXiv preprint arXiv:2512.16468 (2025)

29. Song, X., Chen, W., Liu, Y., Chen, W., Li, G., Lin, L.: Towards long-horizon vision-language navigation: Platform, benchmark and method. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12078–12088 (2025)
30. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)
31. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8248–8258 (2022)
32. Tao, Y., Muñoz-Bañón, M.Á., Zhang, L., Wang, J., Fu, L.F.T., Fallon, M.: The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods. The International Journal of Robotics Research p. 02783649251369905 (2024)
33. Turki, H., Ramanan, D., Satyanarayanan, M.: Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12922–12931 (2022)
34. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
35. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10510–10522 (2025)
36. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20697–20709 (2024)
37. Wang, Z.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
38. Wani, S., Patel, S., Jain, U., Chang, A., Savva, M.: Multion: Benchmarking semantic map memory using multi-object navigation. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 9700–9712. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/6e01383fd96a17ae51cc3e15447e7533-Paper.pdf
39. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., et al.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. arXiv preprint arXiv:2301.00493 (2023)
40. Wu, J.Z., Zhang, Y., Turki, H., Ren, X., Gao, J., Shou, M.Z., Fidler, S., Gojcic, Z., Ling, H.: Difix3d+: Improving 3d reconstructions with single-step diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26024–26035 (2025)
41. Xiao, P., Shao, Z., Hao, S., Zhang, Z., Chai, X., Jiao, J., Li, Z., Wu, J., Sun, K., Jiang, K., et al.: Pandaset: Advanced sensor suite dataset for autonomous driving. In: 2021 IEEE international intelligent transportation systems conference (ITSC). pp. 3095–3101. IEEE (2021)
42. Xiong, Z., Ye, X., Yaman, B., Cheng, S., Lu, Y., Luo, J., Jacobs, N., Ren, L.: Unidrive-wm: Unified understanding, planning and generation world model for autonomous driving. arXiv preprint arXiv:2601.04453 (2026)

43. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In: European Conference on Computer Vision. pp. 156–173. Springer (2024)
44. Yang, S., Yang, J., Huang, P., Brown, E., Yang, Z., Yu, Y., Tong, S., Zheng, Z., Xu, Y., Wang, M., et al.: Cambrian-s: Towards spatial supersensing in video. arXiv preprint arXiv:2511.04670 (2025)
45. Yao, D., Han, X., Ming, R., Song, Z., Peng, L., Hu, J., Yao, D., Zhang, Y.: A style-based profiling framework for quantifying the synthetic-to-real gap in autonomous driving datasets. arXiv preprint arXiv:2510.10203 (2025)
46. Ye, V., Li, R., Kerr, J., Turkulainen, M., Yi, B., Pan, Z., Seiskari, O., Ye, J., Hu, J., Tancik, M., Kanazawa, A.: gsplat: An open-source library for gaussian splatting. Journal of Machine Learning Research **26**(34), 1–17 (2025)
47. Yu, H., Han, Y., Zhang, X., Yin, B., Chang, B., Han, X., Liu, X., Zhang, J., Pavone, M., Feng, C., Xie, S., Li, Y.: Thinking in 360°: Humanoid visual search in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2026)
48. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
49. Zhao, H., Weng, H., Lu, D., Li, A., Li, J., Panda, A., Xie, S.: On scaling up 3d gaussian splatting training. In: European Conference on Computer Vision. pp. 14–36. Springer (2024)
50. Zheng, D., Huang, S., Zhao, L., Zhong, Y., Wang, L.: Towards learning a generalist model for embodied navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13624–13634 (2024)
51. Zürn, J., Gladkov, P., Dudas, S., Cotter, F., Toteva, S., Shotton, J., Simaiaki, V., Mohan, N.: Wayvescenes101: A dataset and benchmark for novel view synthesis in autonomous driving. arXiv preprint arXiv:2407.08280 (2024)

A Implementation Details

A.1 Data Processing

Timestamp alignment. The raw sensors are asynchronous and operate at different frequencies (e.g., cameras at 10 Hz, LiDAR at 10 Hz, IMU at 100 Hz, and GPS/localization at 2 Hz), so we do not resample all streams to a common clock. Instead, we define each released keyframe by the timestamp of the front camera image,

$$t_k \;=\; t_k^{\text{front}}, \tag{11}$$

and associate the remaining sensor measurements using their original timestamps. For each sensor stream s, we select the temporally closest valid measurement

$$\hat{x}_k^{(s)} \;=\; x^{(s)}_{\arg\min_i \, |t_i^{(s)} - t_k|}. \tag{12}$$

The ego pose is obtained by interpolation from the localization stream,

$$\mathbf{T}_{\text{ego}}(t_k) \;=\; \text{Interp}\big(\{(t_i^{\text{loc}}, \mathbf{T}_i^{\text{loc}})\},\, t_k\big)\,, \tag{13}$$

where $\mathbf{T}_{\text{ego}}(t_k) \in SE(3)$ denotes the ego pose at time t_k. In practice, the timestamp mismatch between the reference image and the associated LiDAR/pose measurements is typically below 50 ms, preserving fine-grained cross-modal consistency without introducing artificial resampling artifacts.

Motion compensation. Each LiDAR sweep is acquired over a finite time interval rather than instantaneously, which introduces geometric distortion when the vehicle moves during scanning. To reduce this effect, we apply sweep-level motion compensation before reconstruction. Let $\mathbf{p}_j$ be a LiDAR point measured at firing time τ_j within the sweep associated with keyframe t_k. After transforming the point to the ego frame, we warp it to the reference time t_k via

$$\tilde{\mathbf{p}}_j \;=\; \mathbf{T}_{\text{ego}}(t_k)^{-1}\, \mathbf{T}_{\text{ego}}(\tau_j)\, \mathbf{p}_j. \tag{14}$$

All LiDAR points in the sweep are deskewed in this manner and expressed at the common reference time t_k. This substantially improves temporal alignment between LiDAR geometry and camera observations, leading to cleaner projected depth and more reliable supervision and evaluation.

Mask pipeline. We provide semantic masks for ground, sky, and potentially movable objects. The initial masks are generated using SAM3 [3] with text prompts, and are further refined using onboard 3D tracking cuboids. To isolate truly dynamic content, we use the velocity estimates from the tracking system and classify an instance n as moving only if

$$\|\mathbf{v}_n\|_2 > 1 \text{ m/s}. \tag{15}$$

The final dynamic mask is formed by projecting only those tracked cuboids satisfying this criterion:

$$\mathcal{M}_{\text{dyn}} \;=\; \bigcup_{n:\, \|\mathbf{v}_n\|_2 > 1} \Pi(\mathcal{B}_n), \tag{16}$$

where $\mathcal{B}_n$ denotes the 3D cuboid of instance n and $\Pi(\cdot)$ its image projection. Tracked instances below this threshold, such as parked vehicles or standing pedestrians, are excluded from the dynamic mask. This design separates static structure from dynamic foregrounds more reliably, making the released masks more suitable for reconstruction, evaluation, and downstream simulation.

Release protocol. To make long-horizon city logs tractable for reconstruction and benchmarking, we spatially subsample keyframes at a fixed interval of 0.5 m. Let $\mathbf{x}_k$ denote the ego translation of keyframe k. We retain keyframes such that consecutive released frames satisfy

$$\|\mathbf{x}_{k_{m+1}} - \mathbf{x}_{k_m}\|_2 \geq 0.5 \text{ m}. \tag{17}$$

Each route is then partitioned into contiguous 5 km chunks according to cumulative traveled distance,

$$s_k \;=\; \sum_{j=1}^{k-1} \|\mathbf{x}_{j+1} - \mathbf{x}_j\|_2, \qquad c_k \;=\; \left\lfloor \frac{s_k}{5000} \right\rfloor . \tag{18}$$

On top of these chunks, we further release sub-trajectories of multiple lengths

$$L \in \{50\,\text{m},\ 250\,\text{m},\ 500\,\text{m},\ 1\,\text{km},\ 2.5\,\text{km},\ 5\,\text{km}\}, \tag{19}$$

so that methods can be evaluated under progressively more challenging spatial scales.

A.2 Our Method Implementation

Our method is built on top of a 3DGS training pipeline with several urban-specific modifications. Unless otherwise specified, all experiments share the same default optimization settings across scenes, while scene-dependent adjustments are limited to the total training steps and the Gaussian capacity budget. The main hyper-parameters of our method are summarized in Table 6.

Sky model. We model the sky separately using a lightweight view-dependent MLP rather than 3D Gaussians. Given a viewing direction $\mathbf{d} \in \mathbb{R}^3$, we first apply a sinusoidal encoder with degree range $[0, 6)$, and then concatenate the encoded direction with a learnable per-image appearance embedding of dimension 16. The resulting feature is passed through a 3-layer MLP with hidden width 64, output dimension 3, and a skip connection at the second layer. Formally, for image i, the sky color is predicted as

$$\mathbf{c}_{\text{sky}} = f_{\text{sky}}([\gamma(\mathbf{d}),\, \mathbf{e}_i])\,,$$

where $\gamma(\cdot)$ is the directional encoder, $\mathbf{e}_i \in \mathbb{R}^{16}$ is the appearance embedding, and f_{sky} is the MLP. A sigmoid activation is applied to the output RGB values. At test time, when image indices are unavailable, we replace $\mathbf{e}_i$ with the mean appearance embedding over the training set.

Table 6: Main hyper-parameters used in our methods. We report the parameters used for Miami 5k-keyframes data as an example. The parameters may change due to the data scale varies, such as training steps, max points and densification steps, but other optimization parameters keep the same.

Hyper-parameter	**Value**
Training steps	600,000
Maximum number of Gaussians	30,000,000
Initial mean learning rate	2×10^{-3}
Final mean LR multiplier	10^{-2}
Depth loss weight λ_{depth}	10^{-1}
Depth mode	disparity
Pose optimization start step	1
Pose optimization learning rate	10^{-3}
Pose optimization regularization	10^{-5}
Pose optimization mode	rig
Densification start step	10,000
Densification stop step	60,000
Densification interval	100
Densify portion	0.005
Ground optimization steps	10,000
Spherical harmonics degree	1
Sky appearance embedding dim	16
Sky MLP layers	3
Sky MLP hidden width	64
Sky skip connection	layer 1

Rig pose optimization. We optimize the ego pose at each keyframe together with the camera extrinsics relative to the ego frame, rather than treating each camera pose independently. Pose optimization is enabled from the beginning of training (`pose_opt_start`=1) in rigid-rig mode, with learning rate 10^{-3} and regularization weight 10^{-5}. This rigid factorization is particularly important for surround-view data, where small localization errors can otherwise accumulate into multi-view inconsistency over long trajectories.

Ground regularization. For Gaussians identified as ground, we apply additional priors to stabilize road geometry, including slice-wise distortion regularization on ground height, shortest-axis alignment, and opacity regularization. These losses are activated only during the early stage of training using a curriculum schedule of 20k steps (`ground_curriculum_steps`=10000), when the geometry is most unstable.

Depth supervision. We apply external depth supervision throughout training using a disparity-based loss with weight 10^{-1} (`depth_mode=disparity` and `depth_lambda`=0.1). This encourages better geometric consistency in large urban scenes, especially in texture-poor regions.

Gaussian optimization and densification. We train for 600k steps with an initial mean learning rate of 2×10^{-3} and a final learning-rate multiplier of 10^{-2}.

The Gaussian capacity is capped at 30M primitives. Densification begins at 10k iterations, ends at 60k iterations, is performed every 100 steps, and adds 0.5% of the current primitive count per refinement step (`densify_portion`=0.005). We use spherical harmonics of degree 1 (`sh_degree`=1).

Multi-GPU training. For long sequences, we follow the distributed design of GrendelGS [49] and shard Gaussian parameters across multiple GPUs. In our default setting, training is performed on two H200 GPUs. This keeps the per-GPU memory footprint bounded and enables training at city scale.

Training protocol. All experiments are trained with dynamic masks enabled and sky modeling activated. Checkpoints are saved at 200k and 400k iterations, and evaluation is performed periodically throughout training, with denser evaluation in the earlier optimization stage and final reporting at 400k steps. We do not use the bilateral grid option in the reported setting.

A.3 Baseline Implementation

We implement all baselines using their official or publicly released codebases whenever available, and adapt only the data loading and preprocessing interfaces necessary to support the WildCity format. All methods use the same camera intrinsics, poses, and train/validation split, and are evaluated under the same masking-aware protocol used in the main paper.

3DGS. We use the standard 3DGS training pipeline [11] as a classic point of reference. Since vanilla 3DGS is not designed for multi-kilometer scenes, it mainly serves as a lower-bound baseline for large-scale street-view reconstruction.

H-3DGS. For H-3DGS [12], we follow the hierarchical Gaussian training procedure and keep its scene decomposition and merging strategy unchanged. This baseline evaluates whether hierarchical Gaussian representations alone are sufficient for continuous street-view trajectories.

CityGS. For CityGaussianV2 [21], we follow the released blockwise training and large-scene rendering setup. This baseline is particularly relevant because it is designed for large-scale Gaussian reconstruction through spatial partitioning.

VGGT-Long and VGGT-Long+CityGS. We use VGGT-Long [6] as a feed-forward long-sequence geometry prior. Since it does not directly produce photo-realistic renderings in our evaluation protocol, we report its depth accuracy after aligning its reconstruction to our metric point-cloud frame via a single global similarity transform. For VGGT-Long+CityGS, we replace the COLMAP initialization in CityGS with the aligned VGGT-Long reconstruction and retain the rest of the CityGS training pipeline unchanged. We also tested VGGT-X as a feed-forward initializer, but on both multi-thousand-view sequences it produced only extremely sparse and unstable point clouds, so we exclude it from the main quantitative table.

Training budget and stopping criterion. We do not enforce a fixed iteration budget across all baselines, because their optimization structures differ substantially. Vanilla 3DGS uses a single-stage end-to-end optimization, H-3DGS uses a base stage followed by post-optimization, and CityGS-based methods use a multi-stage pipeline consisting of coarse optimization, spatial partitioning,

block-wise trimming, merging, and final evaluation. In our experiments, 3DGS is trained for 50k iterations on Ann Arbor-0.5k and 400k iterations on Atlanta-5k, with all urban-specific components disabled, including sky modeling, depth supervision, dynamic masking, and curriculum strategies. H-3DGS follows its released hierarchical pipeline with 50k base iterations plus 15k post-optimization iterations on Ann Arbor-0.5k, and 30k base iterations plus 15k post-optimization iterations on Atlanta-5k. For CityGS and VGGT-Long+CityGS, we convert LiDAR depth to inverse-depth supervision and run the released CityGaussianV2 pipeline. On Ann Arbor-0.5k, both methods use 30k coarse iterations followed by 10k block-wise trim iterations after validation performance saturates. On Atlanta-5k, both methods use 90k coarse iterations followed by 90k block-wise trim iterations, which was the strongest stable budget we completed for the city-scale setting.

For CityGS-based methods, the trim stage uses the released 4×4 spatial partition with block-wise parallel optimization, and we preserve the same downstream optimization schedule when replacing the default COLMAP initialization with the aligned VGGT-Long reconstruction. All runs are conducted on comparable high-memory datacenter GPUs in the H100/H200 class rather than under an artificially fixed device budget. We therefore report each baseline under a strong and stable configuration within its native training paradigm.

B Additional Experiments

B.1 Evaluation Across Cities

To study how reconstruction performance varies across geography, we extend our evaluation to additional cities beyond the main sequences used in the paper. This experiment serves two purposes. First, it measures whether our method can generalize across cities with different street scenes and styles. Second, it reveals how city morphology affects city-scale reconstruction, especially the urban scenes, like Atlanta and rurual scenes, like Grand Rapids.

We consider representative cities with distinct urban characteristics, such as dense downtown regions, residential grids, broad arterial roads, and high-speed corridors. For each city, we evaluate the same method under the same protocol and summarize the results in Table 7. We also include a cross-city generalization setting, where initialization and hyper-parameters are all the same, in order to test the robustness of the pipeline across different urban morphologies.

The results shows that cities with denser intersections, more severe occlusions, and more heterogeneous street layouts pose greater challenges than cities with more regular road structure and cleaner visibility. These results complement the main paper by showing that the difficulty of city-scale reconstruction depends not only on route length, but also on the morphology and operational complexity of the urban environment.

Table 7: Quantitative comparison across different cities (2.5km). We report PSNR, SSIM, and LPIPS on 2.5km sequences. Performance varies across cities due to different urban morphology and scene complexity; for example, Atlanta contains highly textured street scenes and frequent dynamic objects, making reconstruction more challenging. Despite this variation, most cities achieve PSNR above 25 dB, indicating that the reconstruction pipeline remains robust across diverse real-world environments.

City (2.5km)	PSNR ↑	SSIM ↑	LPIPS ↓	D-L1 ↓
Atlanta	23.59	0.8157	0.3648	10.5906
Arlington	24.95	0.8321	0.3421	14.6566
Ann Abour	25.33	0.9040	0.2937	24.1174
Eden Prairie	26.56	0.8771	0.3155	16.9166
Grand Rapids	26.22	0.9165	0.2250	17.0085
Miami	25.07	0.8403	0.3216	14.9335

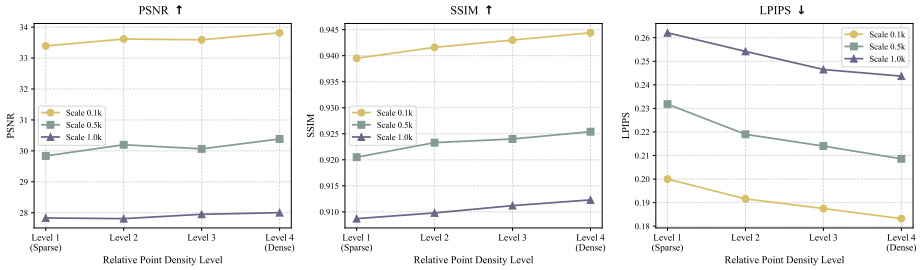


Fig. 10: Quantitative comparison on impact of Gaussian primitive numbers across different scales. Density Levels 1 to 4 correspond to point counts of 3M, 4M, 5M, 6M for Scale 0.1k; 7M, 9M, 11M, 13M for Scale 0.5k; and 15M, 20M, 25M, 30M for Scale 1.0k, respectively.

B.2 Effect of the Number of Gaussian Primitives

We further study the trade-off between reconstruction quality and computational cost under different Gaussian budgets. Specifically, we vary the maximum number of Gaussian primitives while keeping the rest of the training setup fixed, and report the resulting rendering quality, depth accuracy, peak GPU memory, and training time. As illustrated in Figure 10, a higher relative point density level yields an upward trend in the PSNR of the reconstructed scene, alongside a steady increase in SSIM and a steady decrease in LPIPS. This demonstrates that regardless of the scale, adding more Gaussian points enhances the rendering of scene details, thereby improving the overall quantitative metrics.

This experiment helps clarify whether better quality primarily comes from improved modeling design or from simply allocating more Gaussian capacity. It also reveals the practical cost of scaling current pipelines to large urban scenes. In general, increasing the number of primitives improves quality up to a point, but also raises memory usage and optimization cost substantially. This trade-off

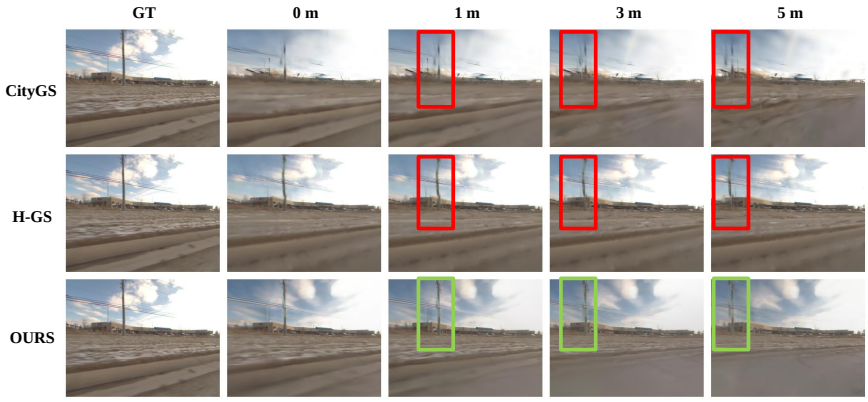


Fig. 11: Qualitative extrapolation under increasing off-trajectory offsets. We render viewpoints with lateral offsets of 1 m, 3 m, and 5 m from the recorded trajectory. This figure shows representative anchor views and compares the recorded-view reference with reconstructions from our method, CityGS, and H-3DGS.

is particularly important for city-scale logs, where Gaussian growth can quickly become a bottleneck.

B.3 Extrapolation Offset Study

To further analyze view extrapolation, we study off-trajectory viewpoints with increasing lateral offsets from the recorded trajectory. Compared with the milder offsets considered in the initial draft, we use a more challenging setting with offsets of $[1, 3, 5]$ m, which makes the degradation trend more visually explicit on long street-view trajectories. Qualitative visualizations are shown in Figure 11–Figure 15. Each row contains five representative anchor views and compares the recorded-view reference together with the renderings produced by our method, CityGS [21], and H-3DGS [12] as the camera progressively departs from the original trajectory.

This experiment complements the qualitative comparison in the main paper by making the extrapolation difficulty more explicit. Across all examples, the visual quality of every method degrades as the viewpoint moves from the recorded pose to 1 m, 3 m, and 5 m off-trajectory, but the failure modes differ substantially. CityGS [21] and H-3DGS [12] increasingly exhibit blurred structures, broken ground surfaces, unstable sky boundaries, and floating artifacts as the offset grows. In contrast, our method remains visibly more stable in both ground and sky regions, which we attribute to the additional geometric supervision imposed on these two structurally important components. The advantage becomes especially clear at larger offsets, where the new viewpoints depart further from the original sampling manifold and therefore stress global geometric consistency rather than local photometric fitting alone.

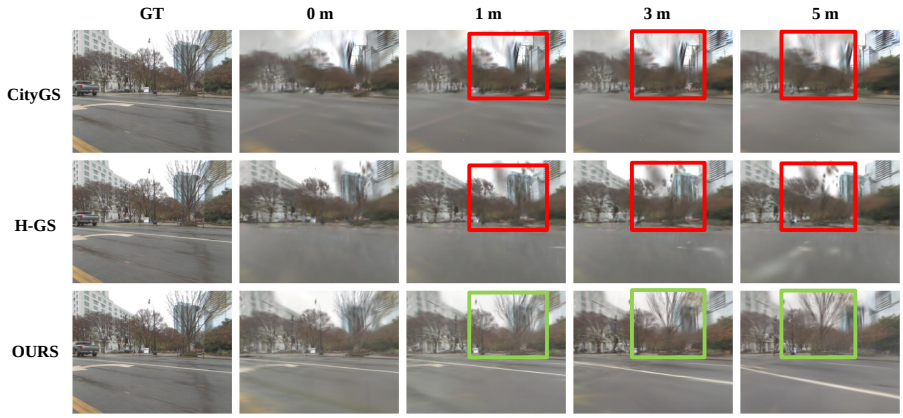


Fig. 12: Additional qualitative extrapolation results. Continued examples under 1 m, 3 m, and 5 m off-trajectory offsets.

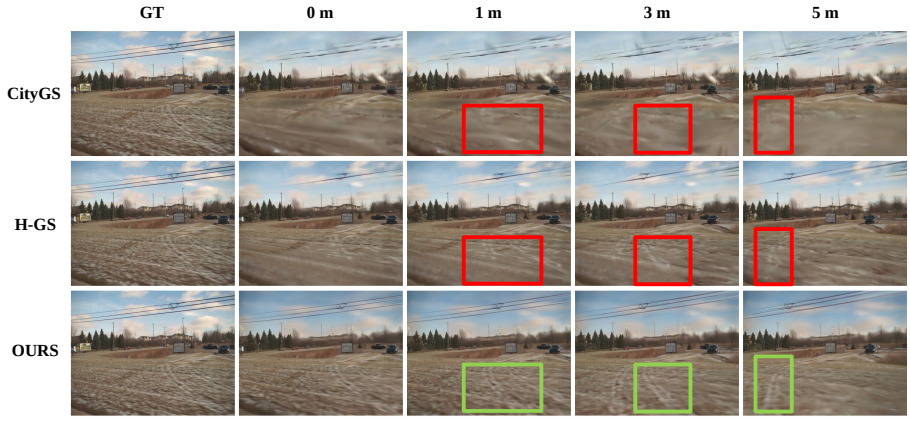


Fig. 13: Additional qualitative extrapolation results. Continued examples under 1 m, 3 m, and 5 m off-trajectory offsets.

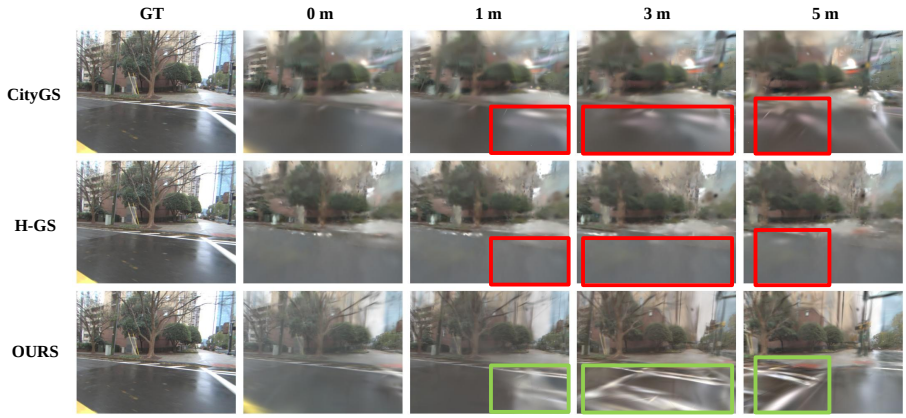


Fig. 14: Additional qualitative extrapolation results. Continued examples under 1 m, 3 m, and 5 m off-trajectory offsets.

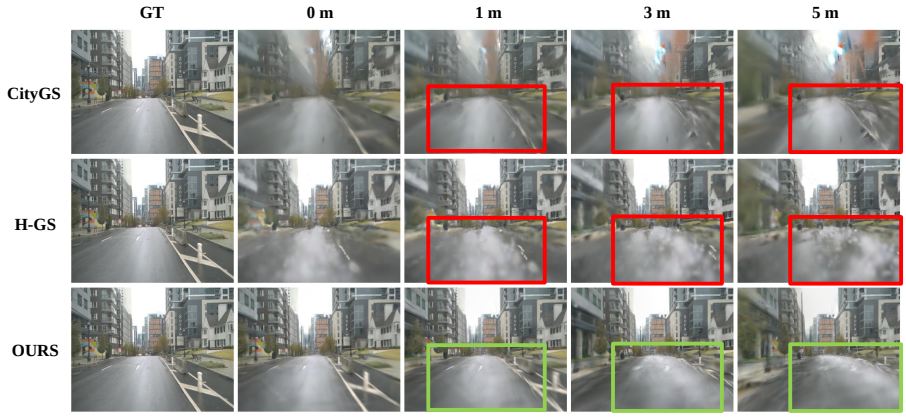


Fig. 15: Additional qualitative extrapolation results. Continued examples under 1 m, 3 m, and 5 m off-trajectory offsets.