

Supplementary material

A. Urban Scene Reconstruction with NeRFs

NeRFs [37] can be used to model dynamic urban scenes. SUDS [51] uses a single network for dynamic actors, which limits the possibility of altering the behavior of the actors. EmerNeRF [65] follows a similar idea to SUDS by decomposing the scene purely into static and dynamic components. NeuRAD [49] takes advantage of monocular or LiDAR-based 3D bounding box predictions and proposes a joint optimization of object poses during the reconstruction process. Although these methods produce reasonable results, they are still 1) limited to the high training cost and low rendering speed; or 2) do not address extrapolation of future vehicle appearance far beyond the original camera views.

B. Data Collection Details

The initial database contained $\sim 95,000$ car videos of ~ 100 views on average. The first filtering stage includes the filtering of low quality and overly dark images with the CLIP-IQA model [54], discarding frames with a score < 0.2 . Then, we use Qwen-2.5-VL-Instruct (7B) [64] to answer several questions for each frame:

- “Does the image depict a car?”
- “Is the car directly occluded?”
- “Does the image depict the car interior?”
- “Does a hand or finger block the view?”
- “Is the car door open?”
- “Does the image mainly depict the car window?”

Based on the responses, we filter out the corresponding frames or, in some cases, entire car instances. Also, if fewer than 45 valid frames remain for a given instance, the entire instance is discarded.

C. Additional Retrieval Evaluation Results

In this section, we provide an additional illustration of our retrieval algorithm. As shown in our Figure 9, introducing a color-based pre-filter enhances the alignment of vehicle colors between retrieved candidates and the target.

D. Evaluation of Realism for Reconstructed Models

Evaluation of realism using retrieved images. We evaluate the realism quality of the reconstructed 3D car models using the retrieval approach of MADRIVE and the generative image-to-3D methods of Amodal3R [59], TRELLIS [60], and SAM3D [48]. For this, we collected 36 random crops of hold-out cars from the Waymo dataset [18] and retrieved images on them from MAD-CARS using

MADRIVE. For a fair setup, we select retrieved input images of MAD-CARS for generative image-to-3D methods of Amodal3R, TRELLIS, and SAM3D with a roughly horizontal viewpoint and a visible half of the vehicle. For all methods, we rendered reconstructed 3D models from 360 degrees with 60 renderings per reconstructed model. For all image-to-3D models, we also segmented cars in crops using the YOLOv11 instance segmentation model [27]. For reference images, we took 30 hold-out cars from MAD-CARS and collected nearly 2200 images for these cars. To reduce the domain shift, we excluded backgrounds from real images from MAD-CARS with YOLOv11. Finally, we computed FID [23] and KID [5] scores between two sets: 1) the set of all renderings from the reconstructed 3D models with MADRIVE, Amodal3R, TRELLIS, and SAM3D, which we ran on 38 retrieved images of MAD-CARS; 2) the set of real images for 30 test cars from MAD-CARS. We provide visual results in Figure 8 and quantitative results in Table 3.

We found that image-to-3D methods produce substantially higher-quality assets for the input images of high-resolution retrieved MAD-CARS images compared to the results on input images of low-resolution Waymo crops. Nevertheless, their outputs remain less realistic than assets reconstructed from multi-view video and often do not support relighting. Quantitatively, MADRIVE achieves superior FID and KID scores compared to all image-to-3D baselines. We omit reconstruction error metrics, as they would unfairly favor our method: MADRIVE reconstructs appearance from dense multi-view observations, whereas image-to-3D methods must hallucinate unseen geometry and texture from a single image.



Figure 8. Qualitative assessment of car model quality for our method and image-to-3D models.

E. Relighting Evaluation

Generative relighting comparison To illustrate generative relighting approaches, we evaluated publicly available IC-Light [71] and ran GPT-5.2-based image adaptation (Fig. 10). IC-Light preserves the original image structure and fine details, but finds a poor match to the target color distribution in our domain (Tab. 4). In contrast, GPT-5.2 achieves a closer color match, but at higher computational cost and with reduced controllability (e.g., lateral vehicle shifts).

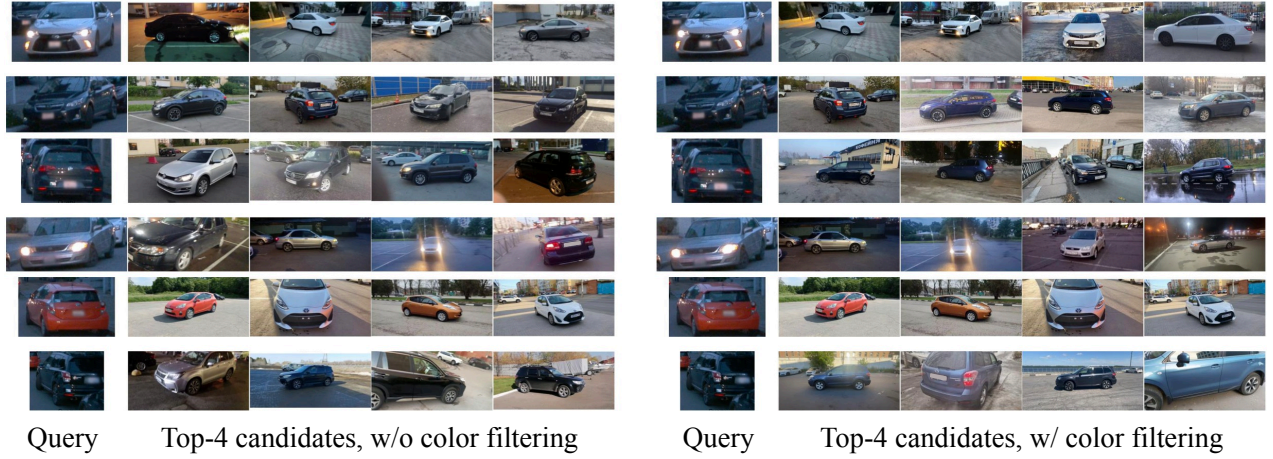


Figure 9. **Retrieval illustration.** Top-4 candidates retrieved using SigLIP 2 without (Left) and with (Right) color filtering.



Figure 10. Relighting approaches compared.

Effect of asset relighting To assess the impact of our relighting module, we analyze the cars shown in Figure 7.

Standard pixel-wise metrics (e.g., LPIPS) failed to capture meaningful differences, largely because geometric mismatches between the inserted asset and the original vehicle dominate these scores. Instead, we evaluate relighting by comparing the color statistics of the integrated assets to those of the real cars.

We extract crops of the inserted vehicles and matching crops from the corresponding ground-truth frames. Each crop is converted to the perceptually uniform CIELAB color space, and we compare their pixel-intensity distributions using the sliced Wasserstein distance [44]. Average distances with standard deviations between different crops are provided in Table 4.

Across all evaluated cars, relighting consistently re-

duces the discrepancy between synthetic and real crops. A more detailed breakdown reveals that the improvement is especially pronounced in the lightness (L^*) channel, while differences in the chromatic channels (a^* , b^*) are smaller—reflecting the fact that lighting primarily affects perceived brightness rather than intrinsic surface color.

	W_1	W_{1,L^*}	$W_{1,(a^*,b^*)}$
W/o relighting	6.74 ± 1.83	10.89 ± 4.78	3.82 ± 3.34
W/ relighting	3.15 ± 0.90	3.54 ± 2.75	2.89 ± 1.30

Table 4. Sliced Wasserstein distances between real and synthetic car crops in CIELAB space. Lower values indicate closer color distribution matches. Relighting consistently reduces the discrepancy, with the largest improvement observed in the lightness (L^*) channel.



Figure 11. **Additional qualitative comparison** of MADrive with non-retrieval-based driving scene reconstruction methods. Reconstruction of the training views (Top). Reconstruction of the hold-out (future) views (Bottom).



Figure 12. **Additional qualitative comparison** of MADRIVE with non-retrieval-based driving scene reconstruction methods. Reconstruction of the training views (Top). Reconstruction of the hold-out (future) views (Bottom).

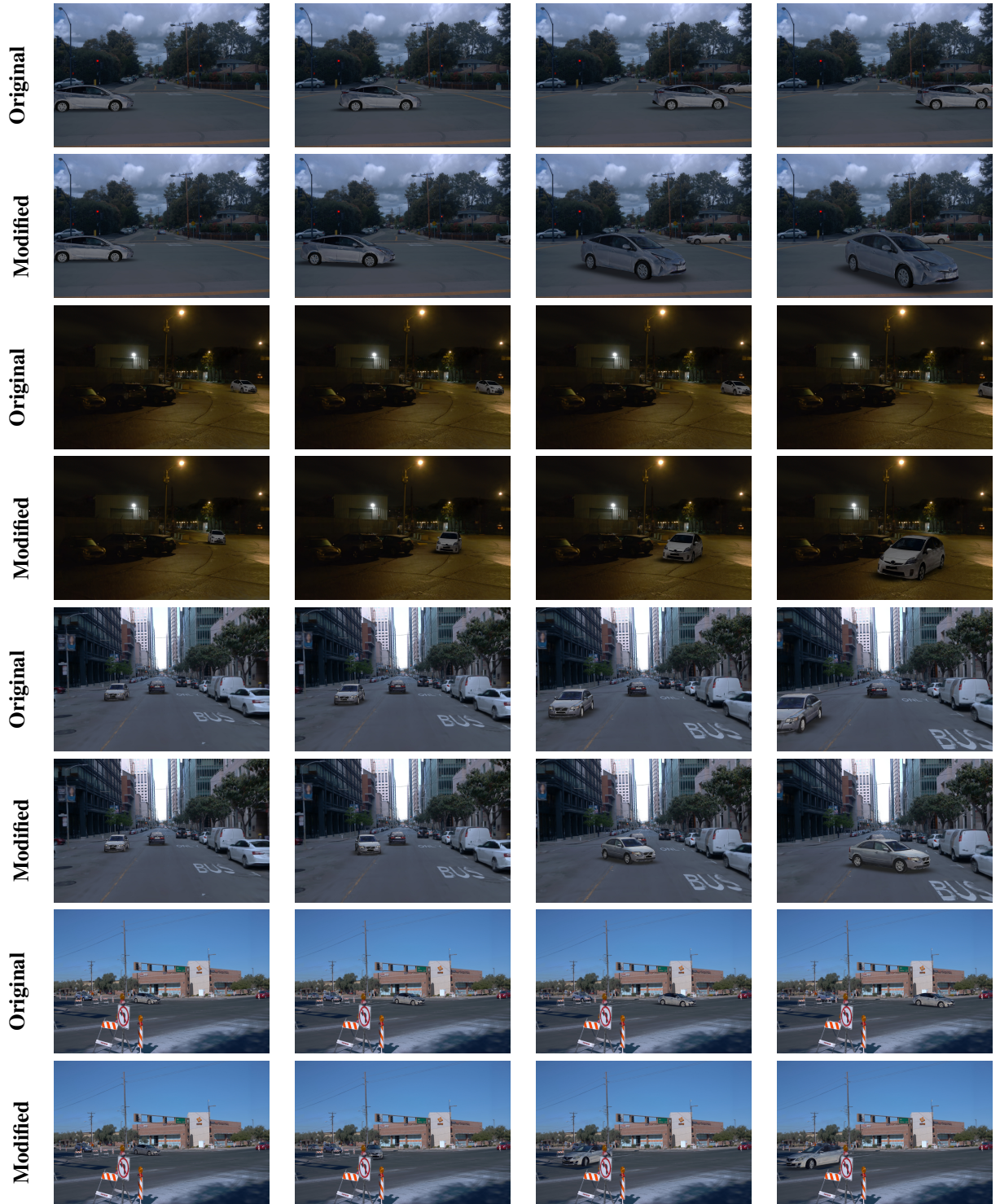


Figure 13. Visualization of original and modified trajectories with MADRIVE. The cars retain high-fidelity appearance even at close distances to the ego camera.

F. Additional Qualitative Comparisons and New Trajectories

We provide additional visual results in Figures 11 and 12. We also demonstrate our method’s capability to render novel views with substantial scene variations. Figure 13 showcases results across four test scenes, where all modifications preserve high image quality.

G. Choice of Reference Masks in the Evaluation

In our validation setup, we used predictions from tracking and segmentation models on ground-truth images as targets, since the Waymo dataset lacks segmentation masks and 2D bounding boxes.

To evaluate whether cars in the synthesized frames are as identifiable as those in the original frames, we applied the same detection algorithm to both. Our method outperforms the baseline, primarily because our system inserts visually coherent cars on test frames by leveraging reconstructed models, whereas baseline approaches result in degraded or incomplete vehicle representations.

However, the inserted cars might be easier to detect. To test this, we conducted an additional experiment. Specifically, we generated a new set of detector targets by projecting the 3D bounding boxes provided in the Waymo dataset onto the image plane. We then evaluated the performance of the detector on both the ground-truth and synthesized (MADrive) frames using the new "ground-truth" annotation.

The results in Table 5 show that the predictions on ground-truth images align slightly better with the projected 3D bounding boxes than those on the synthesized MADrive frames. This indicates that our inserted cars do not artificially simplify detection, supporting the validity of our evaluation.

Table 5. Comparison between detector performance on both the ground-truth and synthesized (MADrive) frames using projected 3D bounding boxes.

Model	MOTA \uparrow	MOTP \downarrow	IDF1 \uparrow
GT frames	0.879	0.270	0.928
MADrive frames	0.861	0.340	0.908

H. Evaluation Setup Details

For scene reconstruction evaluation, we selected 12 scenes from the Waymo Open Dataset [46], with labels listed in Table 6. This table also provides the correspondence between the original scene labels from the [Waymo Cloud Storage](#)

and the short names used in our work. We split each scene into training and testing subsets based on time (Table 7) and camera selection (Table 8). Specifically, frames with indices i^{train} , where $i^{\text{train}} \in [i_{\text{start}}^{\text{train}}, i_{\text{end}}^{\text{train}}]$, were used for training. For evaluation, we used frames $i^{\text{test}} \in [i_{\text{start}}^{\text{test}}, i_{\text{end}}^{\text{test}}]$, with all split indices provided in Table 7.

Table 6. Waymo scenes used for evaluation of scene reconstruction.

Label	Scene name
1231623110026745648_480_000_500_000	123
1432918953215186312_5101_320_5121_320	143
1906113358876584689_1359_560_1379_560	190
10500357041547037089_1474_800_1494_800	105
10940952441434390507_1888_710_1908_710	109
16504318334867223853_480_000_500_000	165
17407069523496279950_4354_900_4374_900	174
18025338595059503802_571_216_591_216	180
14183710428479823719_3140_000_3160_000	141
15834329472172048691_2956_760_2976_760	158
17647858901077503501_1500_000_1520_000	176
7799671367768576481_260_000_280_000	779

Table 7. Train and test frame splits for Waymo scenes over time. All values, except those in the leftmost column, indicate frame indices starting from 0.

Scene name	$i_{\text{start}}^{\text{train}}$	$i_{\text{end}}^{\text{train}}$	$i_{\text{start}}^{\text{test}}$	$i_{\text{end}}^{\text{test}}$
123	106	116	117	175
143	43	53	54	62
190	115	125	126	137
105	164	174	175	196
109	1	16	17	55
165	7	40	41	111
174	34	51	52	72
180	49	55	56	68
141	60	80	81	117
158	44	62	63	100
176	31	42	43	67
779	50	65	66	84

I. Per-Scene Quantitative Evaluation.

In addition to the aggregated results in Table 1, we report per-scene metric values in Table 9, Table 10, Table 11, and Table 12, corresponding to MOTA, MOTP, IDF1, and IoU, respectively. We observe that MADrive consistently outperforms the baselines across most scenes.

J. Baseline Details

Baselines training and evaluation. We trained all baselines (Street-Gaussians, HUGS, and AutoSplat) for 10K

Table 8. Train and test frame splits for Waymo scenes based on camera selection.

Scene name	Train cameras	Test cameras
123	frontal, frontal left	frontal, frontal left
143	frontal, frontal left	frontal, frontal left
190	frontal, frontal left	frontal, frontal left
105	frontal, frontal left	frontal
109	frontal, frontal right	frontal right
165	frontal, frontal left	frontal, frontal left
174	frontal	frontal
180	frontal, frontal right	frontal, frontal right
141	frontal	frontal
158	frontal	frontal
176	frontal	frontal
779	frontal, frontal left, frontal right	frontal, frontal right

Table 9. Mean MOTA \uparrow results on test frames for all Waymo scenes.

Scene name	SG	HUGS	AutoSplat	MADRIIVE
123	0.687	0.685	0.327	0.887
143	0.650	0.513	0.600	0.825
190	0.787	0.795	0.904	0.858
105	0.906	0.656	0.742	0.906
109	0.242	0.448	0.605	0.925
165	0.684	0.461	0.788	0.883
174	0.809	0.886	0.830	0.936
180	0.611	0.528	0.695	0.778
141	0.667	0.607	0.163	0.767
158	0.423	0.233	0.681	0.639
176	0.727	0.562	0.176	0.912
779	0.661	0.296	0.545	0.779

Table 10. Mean MOTP \downarrow results on test frames for all Waymo scenes.

Scene name	SG	HUGS	AutoSplat	MADRIIVE
123	0.073	0.093	0.099	0.079
143	0.114	0.461	0.095	0.203
190	0.088	0.112	0.115	0.144
105	0.073	0.262	0.222	0.118
109	0.093	0.132	0.094	0.122
165	0.125	0.202	0.119	0.149
174	0.075	0.886	0.078	0.093
180	0.150	0.231	0.194	0.195
141	0.119	0.261	0.237	0.179
158	0.087	0.128	0.123	0.119
176	0.093	0.246	0.167	0.072
779	0.176	0.443	0.305	0.180

iterations using the training frames with indices $i \in [i_{\text{start}}^{\text{train}}, i_{\text{end}}^{\text{train}}]$ as specified in Table 7. Additionally, we trained the background models for both the baselines and MADRIIVE for $30K$ iterations using all available frames.

Table 11. Mean IDF1 \uparrow results on test frames for all Waymo scenes.

Scene name	SG	HUGS	AutoSplat	MADRIIVE
123	0.804	0.806	0.475	0.940
143	0.787	0.709	0.750	0.904
190	0.880	0.887	0.950	0.924
105	0.952	0.780	0.877	0.951
109	0.390	0.619	0.754	0.961
165	0.806	0.612	0.894	0.936
174	0.894	0.940	0.907	0.967
180	0.753	0.709	0.820	0.871
141	0.805	0.698	0.278	0.866
158	0.605	0.377	0.829	0.797
176	0.847	0.720	0.316	0.955
779	0.793	0.532	0.739	0.881

Table 12. Mean IoU \uparrow results on test frames for all Waymo scenes.

Scene name	SG	HUGS	AutoSplat	MADRIIVE
123	0.753	0.608	0.500	0.866
143	0.485	0.243	0.510	0.779
190	0.707	0.519	0.740	0.846
105	0.671	0.425	0.439	0.731
109	0.499	0.246	0.419	0.832
165	0.633	0.459	0.647	0.730
174	0.695	0.581	0.655	0.829
180	0.475	0.238	0.582	0.814
141	0.607	0.196	0.226	0.765
158	0.404	0.135	0.498	0.862
176	0.499	0.187	0.263	0.886
779	0.247	0.153	0.273	0.874

These pretrained background models were then used during the rendering of the test frames ($i \in [i_{\text{start}}^{\text{test}}, i_{\text{end}}^{\text{test}}]$), on which we compute the metrics reported in Table 9, Table 10, Table 11, and Table 12.

Street-Gaussians. We used the official implementation available at https://github.com/zju3dv/street_gaussians.

HUGS. We used the official implementation provided at <https://github.com/hy Zhou404/HUGSIM>.

AutoSplat. As no official implementation is publicly available, we re-implemented the core contributions of AutoSplat on top of the Street-Gaussians codebase.

K. Limitations

Reconstruction limitations. To run reconstruction, we estimate camera parameters from the input images. In par-

ticular, we run bundle adjustment starting from the initialization obtained with VGGT. At present, errors in camera estimation remain a primary source of reconstruction failures. We expect that continued advances in foundational vision models will substantially reduce this limitation.

State-of-the-art multiview reconstruction methods continue to struggle with reflective and glossy surfaces like cars even up to this day. Accurate modeling of reflections on metallic surfaces on real datasets demands more precise representations of illumination - beyond what conventional environment maps can provide.

Dataset limitations. We rely on an external dataset sourced from online car sale advertisements, which primarily features passenger vehicles. As a result, other vehicle categories (e.g. buses, trucks, and service vehicles) are underrepresented and cannot yet be reliably replaced by our method.

Nonetheless, our pipeline is fully modular: adding support for additional vehicle types only requires capturing a 360° video of the target vehicle to add it to the retrieval database.

L. Statement on LLM usage

The authors used the large language model (LLM) only to improve the writing and grammar of the text. All the results from the LLM were checked by the authors.