DGGT: Feedforward 4D Reconstruction of Dynamic Driving Scenes using Unposed Images

Supplementary Material

6. Implementation Details

Appendix A.1 — Data Preprocessing

We construct pseudo dynamic masks based on the LiDAR-based 3D bounding-box annotations of the Waymo Open Dataset [48], which include tracking identifiers. Specifically, we first transform the center of each 3D box from the ego-vehicle coordinate frame to the global coordinate system, and aggregate per-object temporal trajectories using the provided timestamps. Object velocities are then estimated, and category-specific thresholds are applied to determine dynamic instances (pedestrians: $> 0.2\,\text{m/s}$; vehicles: $> 0.5\,\text{m/s}$). For objects identified as dynamic, we project their 3D bounding boxes onto the corresponding camera planes to obtain accurate 2D bounding boxes for each frame. These 2D boxes are subsequently used as prompts for an *off-the-shelf* instance segmentation model (SAM2) [43], combined with temporal information to propagate masks and obtain temporally consistent, object-level instance masks. Static background and sky masks are produced using a semantic segmentation model [64], whose semantic outputs are further employed in downstream scene understanding tasks.

Appendix A.2 — Baseline implementations

Baseline selection. For all per-scene reconstruction methods shown in Tab.1, we selected ten representative baselines. Among them, PVG[6], DeformableGS[75], 3DGS[23], and STORM[73] are reconstruction methods implemented on the Waymo Open Dataset[48]; MVSplat[7] and DepthSplat[65] are feedforward inference networks with strong empirical performance; NoPoSplat[76] is a *pose-free* inference framework that does not require camera pose inputs. In addition, we include EmerNeRF[72], LGM[50], and GS-LRM[83] from STORM's reproduced results. For some baselines, reproduced outputs reported by STORM are used directly.

Task setup. The task in Tab.1 and Fig.3 is short-sequence reconstruction and prediction. To align with STORM's experimental setup, we condition on input frames with IDs 0, 5, 10 and 15 and predict frames 0-19 (20 frames in total) – although our model imposes no restriction on the number or indices of input and output frames. Inference is performed from three camera viewpoints – the forward-facing, front-left and front-right cameras. This represents a basic scene reconstruction and novel view synthesis (NVS) task. Across the ten experiments, methods that require iterative fitting are uniformly trained/fitted for 5,000 iterations; feedforward reconstruction methods are evaluated without this iterative training constraint. All evaluations are performed on the Scene Flow validation split of the Waymo Open Dataset[48], which contains 202 scenes.

Sky mask and depth metrics. Some methods do not explicitly reconstruct the sky; therefore depth-related metrics are computed only over non-sky regions. Sky masks are obtained from Waymo's LiDAR data and filtered to ensure high confidence. For depth evaluation, methods without camera pose input, including NoPoSplat and ours, predict only relative depth, whose scale and offset may differ from ground truth. To ensure fair comparison, we perform a linear alignment of the predictions before computing the error, and then report the aligned depth RMSE (D-RMSE) within the valid mask regions.

Computation. During the training phase, the model was trained on the Waymo Open Dataset[48] using an eight-card H200 GPU configuration. The training process was completed in approximately 24 hours, with convergence achieved at around 5,000 iterations, as indicated by the stabilization of loss values and performance metrics on the validation set. In the experimental phase, to ensure direct comparability with STORM[73] and its reproduced baseline methods, all evaluations in this study were exclusively conducted on NVIDIA A100 GPUs. This hardware alignment guarantees a consistent and fair comparison of computational performance and results across different models.

7. Additional Experimental Results

Appendix B.1 — Comparison on addtional datasets

We evaluate the generalization of our model on the public nuScenes [2] and Argoverse2[60] datasets. To provide a systematic comparison of cross-domain and in-domain performance, we design two complementary experiments: (1) **zero-shot evaluation**, where the model is trained on the Waymo Open Dataset[48] and tested directly on nuScenes/Argoverse2 to assess cross-domain generalization; (2) **target-domain training evaluation**, where the model is trained and evaluated independently on nuScenes and Argoverse2 to measure upper-bound performance within each target domain.

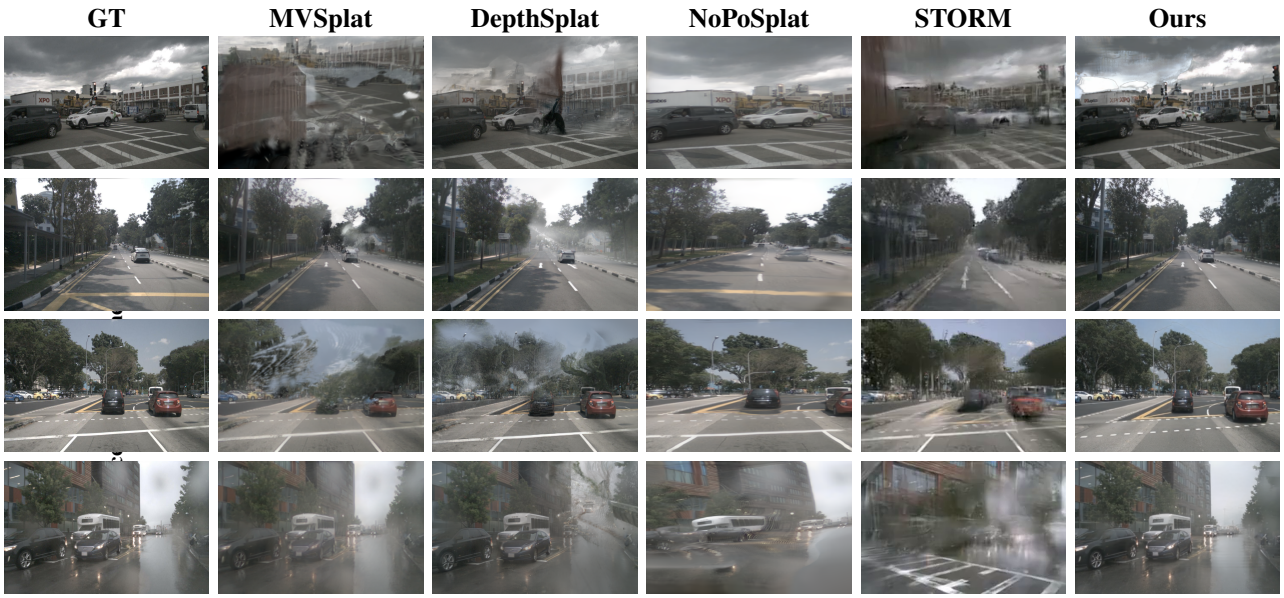

Figure 7. Zero-shot experiment on nuScenes and Argoverse2 datasets.

Figure 8. Lane-change scenes.

Sampling and split details are as follows. For nuScenes (v1.0), the dataset contains approximately 1,000 driving scenes, each lasting roughly 20 s, with camera sampling at 12 Hz. We randomly sample 600 scenes for this study, using the first 500 scenes for training and the remaining 100 scenes for testing. For the Argoverse2 Sensor Dataset, which comprises roughly 1,000 annotated driving sequences and provides multi-modal sensor observations, the camera trigger frequency is approximately 20 Hz; we likewise select 600 sequences, with 500 used for training and 100 for evaluation.

To ensure comparability across datasets, all experiments adopt the same preprocessing and training configuration used for Waymo: camera images are uniformly downsampled/resampled to 518×518, and data augmentation, optimizer settings, and training schedules are kept consistent. Models typically converge after roughly 1,000 epochs. Selected zero-shot inference examples on nuScenes and Argoverse2 are shown in Fig. 7.

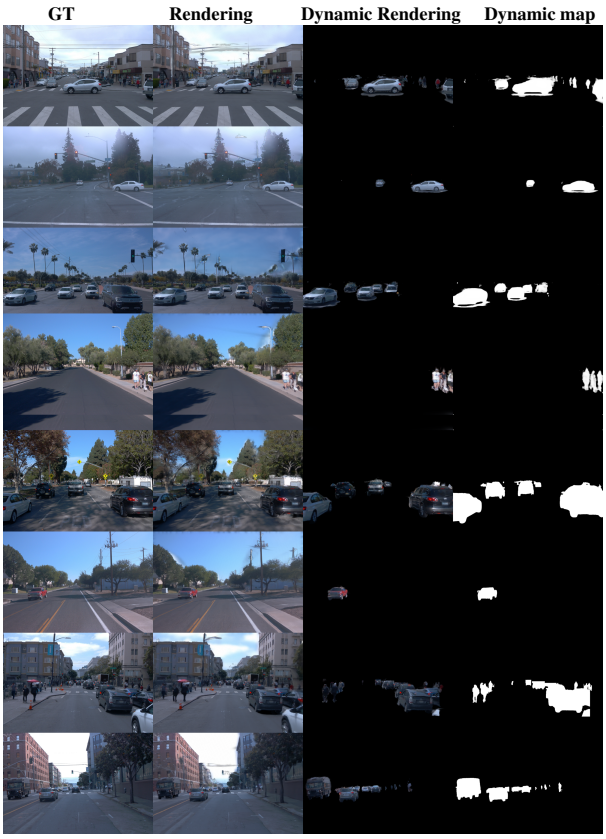

Figure 9. More Qualitative Results.

Appendix B.2 — Lane-change case studies

We selected representative lane-change sequences and present qualitative results in Fig. 8 Each example focuses on modest lateral displacements corresponding to single-lane maneuvers. After these small-magnitude lane shifts,

DGGT maintains overall scene stability: object identity and appearance are preserved across frames, and the rendered frames exhibit strong visual coherence with limited temporal flicker.

Appendix B.3 — more qualitative results

Fig. 9 presents additional qualitative results of our method, showing full-image renderings, dynamic object renderings, and the predicted dynamic masks. Our approach effectively separates dynamic elements, such as vehicles and pedestrians, from the static background across diverse urban driving scenarios. The dynamic renderings align closely with ground-truth object locations, and the dynamic maps provide accurate object masks, demonstrating the effectiveness of our dynamic scene modeling and motion decomposition.