SceneEdited: A City-Scale Benchmark for 3D HD Map Updating via Image-Guided Change Detection

Supplementary Material

A. Supplemental: Static and Dynamic Labels

The label definitions used in this work are directly inherited from the Argoverse dataset [1, 5], which provides a comprehensive taxonomy of object categories commonly encountered in urban driving scenarios. We further reorganize these categories into two groups: *dynamic* and *static* to create *SceneEdited*. The full lists of static and dynamic labels are listed in the following paragraph:

Static Labels (9 total)

- BOLLARD
- CONSTRUCTION_BARREL
- CONSTRUCTION_CONE
- MESSAGE_BOARD_TRAILER
- MOBILE_PEDESTRIAN_CROSSING_SIGN
- OFFICIAL_SIGNALER
- SIGN
- STOP_SIGN
- TRAFFIC_LIGHT_TRAILER

Dynamic Labels (21 total)

- ANIMAL
- ARTICULATED_BUS
- BICYCLE
- BICYCLIST
- BOX_TRUCK
- BUS
- DOG
- LARGE_VEHICLE
- MOTORCYCLE
- MOTORCYCLIST
- PEDESTRIAN
- RAILED_VEHICLE
- REGULAR_VEHICLE
- SCHOOL_BUS
- STROLLER
- TRUCK
- TRUCK_CAB
- VEHICULAR_TRAILER
- WHEELCHAIR
- WHEELED_DEVICE
- WHEELED_RIDER

B. Supplemental: Changes Objects in *SceneEdited*

To better characterize the changes introduced in *SceneEdited*, we report the distribution of edited objects across both deletions and additions from 2,235 out-of-date scenes. Fig. 1 summarizes the objects that were removed from the original static scenes P^*_{upd}. These deletions include both static objects annotated from Argoverse [1, 5] and our newly annotated static objects (e.g., buildings, trees, tunnels, overpasses). Notably, our newly introduced annotations aim to bring important and more challenging change objects into the *SceneEdited* dataset. In particular, these manually added labels correspond to large-scale or occlusion-heavy structures that are difficult to recover from a single image. For instance, the geometry of a tunnel or overpass is rarely visible in its entirety from any single viewpoint, and buildings often extend beyond the field of view of an individual frame. As a result, recovering these categories requires aggregating evidence across multiple images during reconstruction, which makes them especially valuable for benchmarking the robustness of change-detection and map-updating pipelines.

Fig. 2 illustrates the objects that were added into the outdated scenes. These additions are drawn from our curated patch database, which contains pre-segmented object patches that can be reinserted into novel locations. Each object patch in the database has been manually inspected and cleaned to remove artifacts, incomplete geometry, and noisy boundaries. This ensures that when new objects are inserted, they integrate more cleanly into the voxelized point clouds and better reflect realistic scene changes. The distribution is dominated by frequently encountered road furniture such as stop signs, signs, and bollards, while less common categories like vehicular trailers or construction-related objects appear in smaller quantities. Together, these additions highlight common urban changes while maintaining high-quality insertions thanks to the manual refinement of the patch database.

C. Supplemental: Change Maps

While the sparse change map directly reflects the projected 3D change points, it is often too discontinuous to serve as reliable supervision for image-level tasks. Sparse detections make it difficult to delineate object boundaries and may leave large gaps that reduce label quality. By refining the sparse map into a dense change mask, we obtain

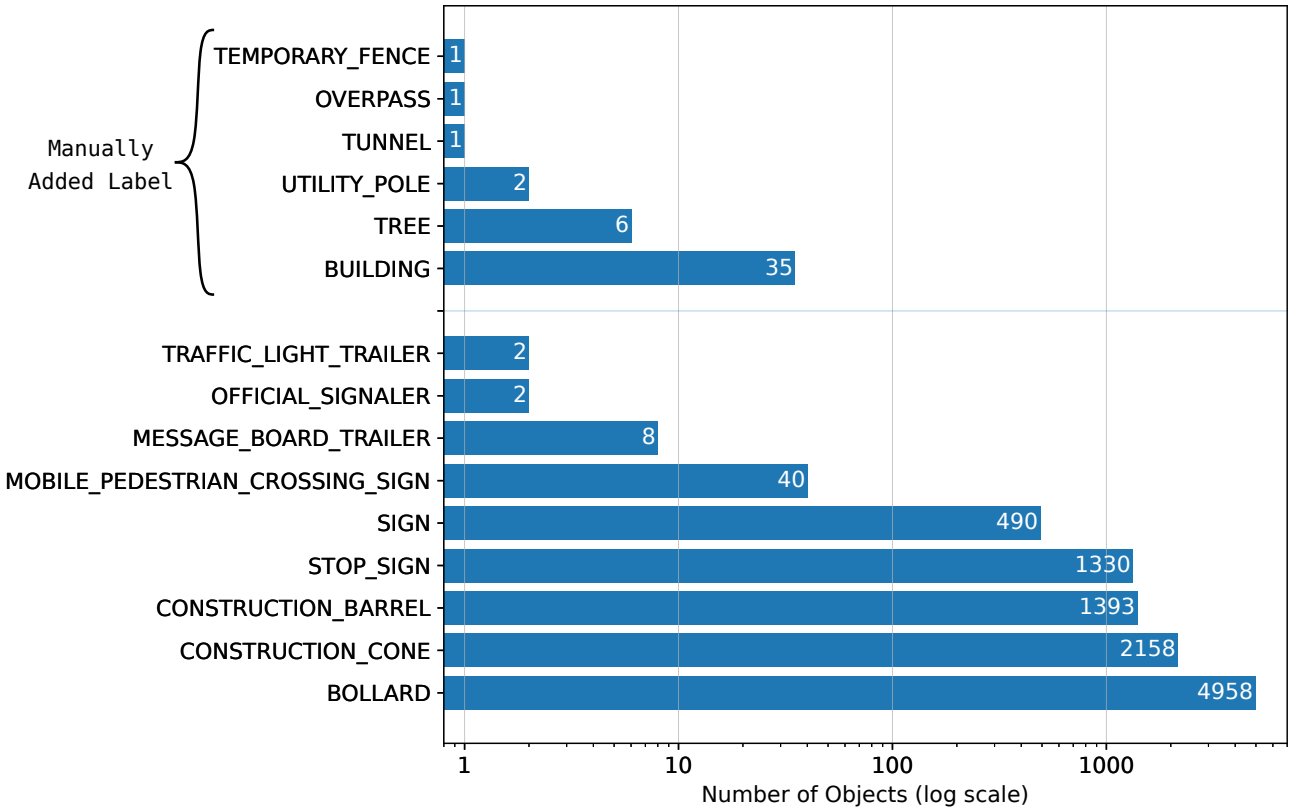

Figure 1. Statistic of missing objects from P_{out} in *SceneEdited* dataset. Each object will remove a number of voxels (points) from P^*_{upd}.

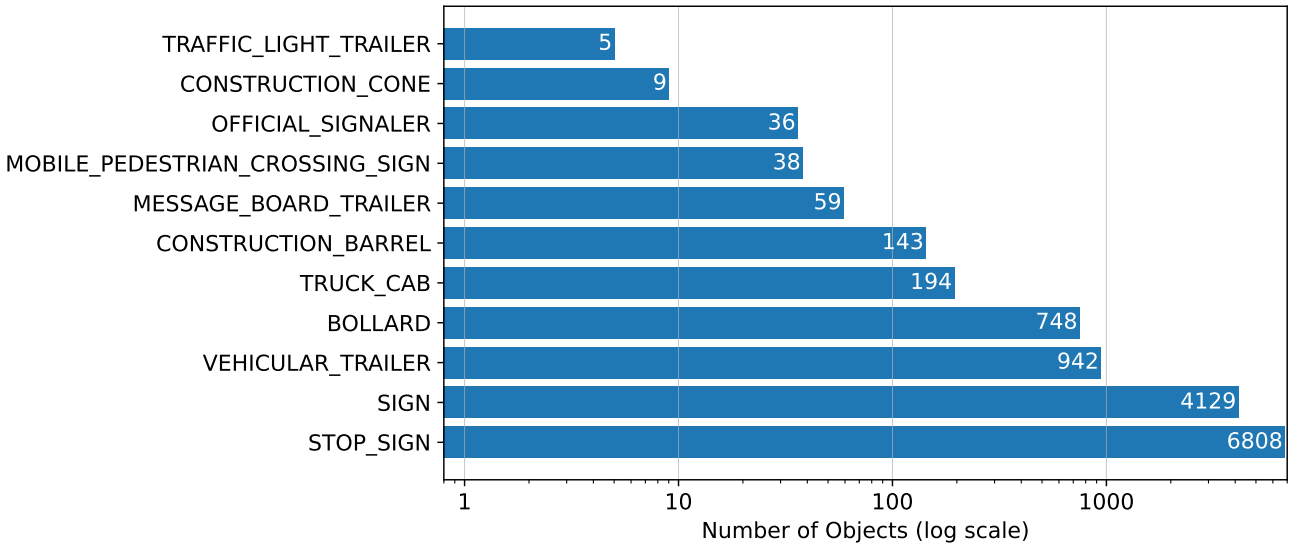

Figure 2. Statistic of outdated objects from P_{out} in *SceneEdited* dataset. Each object will add a number of voxels (points) into P^*_{upd}.

spatially coherent regions that more accurately capture the extent of each changed object, enabling robust training and evaluation of image-based models.

To obtain the dense map Fig. 3j, we further refine the

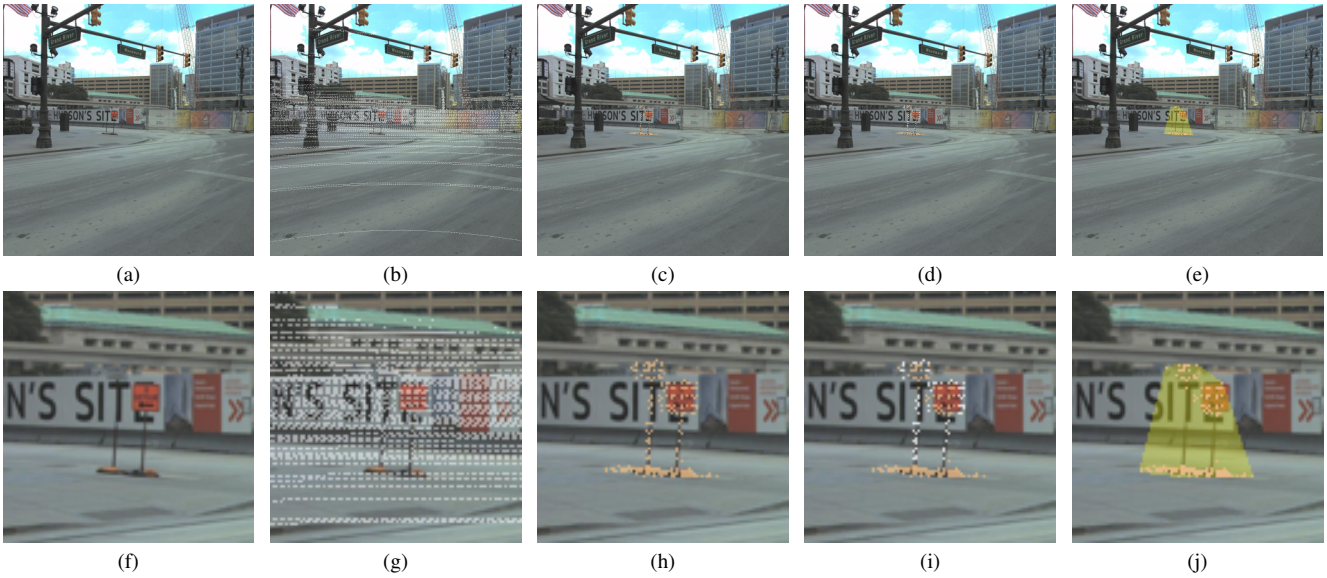

Figure 3. Step-by-step illustration of inferring image-space change maps from outdated 3D scenes. (a) Raw RGB image. (b) RGB image with synchronized LiDAR scan (white). (c) RGB image with sparse P_{out} change map (tangerline). (d) Sparse change map (tangeline) after removing occluded or spurious detections (white) using LiDAR scans. (e) Final convex hull of the filtered change points (yellow). (f–j) Zoomed-in views of the corresponding steps in the first row.

sparse projections Fig. 3h by checking the consistency of each pixel with the raw LiDAR scan Fig. 3g. Specifically, we compare the depth of a projected change pixel against nearby LiDAR measurements in its local neighborhood. If the change point is likely occluded by closer geometry, it is removed Fig. 3i. This process reduces spurious detections and ensures that only visible change regions are retained before generating the final dense masks.

D. Supplemental: Point Addition Comparison

To examine the effect of input image count on reconstruction, we conducted additional experiments across a range of image numbers for each method. The results in Fig. 4 show that increasing the number of images does not improve robustness; instead, it often exacerbates failure rates and degrades accuracy. This suggests that simply feeding more images into these predictors does not necessarily yield more reliable reconstructions.

Based on these observations, we set the number of input images to 5 in the main paper. This choice minimizes the risk of instability while still allowing each method to function within its reliable regime. Using a small but sufficient number of images provides a fairer ground for comparison, avoids the severe failure cases observed at larger scales, and ensures that the reported benchmarks are representative of stable model performance.

References

[1] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8748–8757, 2019. 1

[2] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 4

[3] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 4

[4] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 4

[5] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. 1

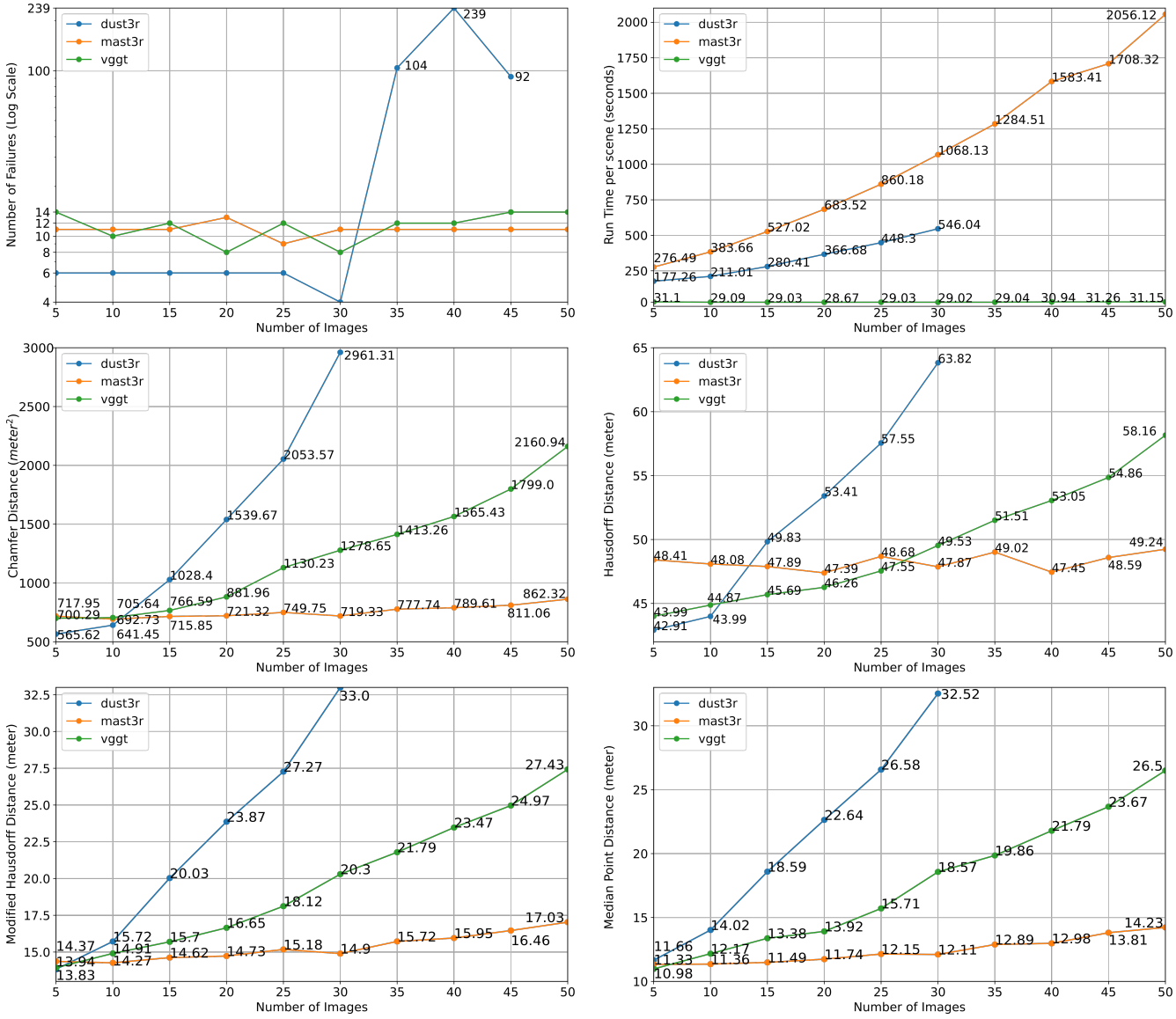

Figure 4. Quantitative comparison of reconstruction performance across different numbers of input images for DUSt3R[4], MASt3R[2], and VGGT[3] Metrics include: (top-left) number of reconstruction failures (log scale), (top-right) runtime per scene, (middle-left) Chamfer distance, (middle-right) Hausdorff distance, (bottom-left) modified Hausdorff distance, and (bottom-right) median point distance. Note that we do not report DUSt3R results beyond 35 input images due to excessive failure rates, as highlighted in the top-left plot.