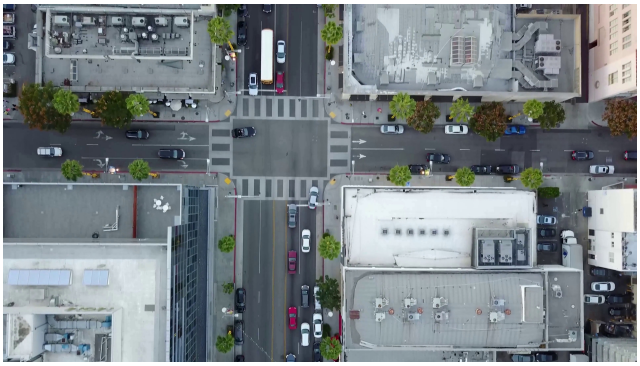
AeroDGS: Physically Consistent Dynamic Gaussian Splatting for Single-Sequence Aerial 4D Reconstruction

Supplementary Material

(a) Input aerial sequence (Downtown-High).

(b) Rendered video from our reconstructed model (Downtown-High).



(c) Input aerial sequence (Intersection-Day).

(d) Rendered video from our reconstructed model (Intersection-Day).

Figure 5. Demonstration videos. We compare the input aerial videos (a, c) with the rendered videos produced by our reconstructed model (b, d), demonstrating that our approach achieves high-fidelity and temporally consistent reconstructions. Please use **Adobe Reader / PDF-XChange Editor** to see animations.

In the appendix, we provide additional details and extended experimental analysis. Sec. A presents the supplementary video results. Sec. B provides further details of the proposed method. Sec. C reports additional information about the dataset. Sec. D offers further analysis and discussion of the experimental findings.

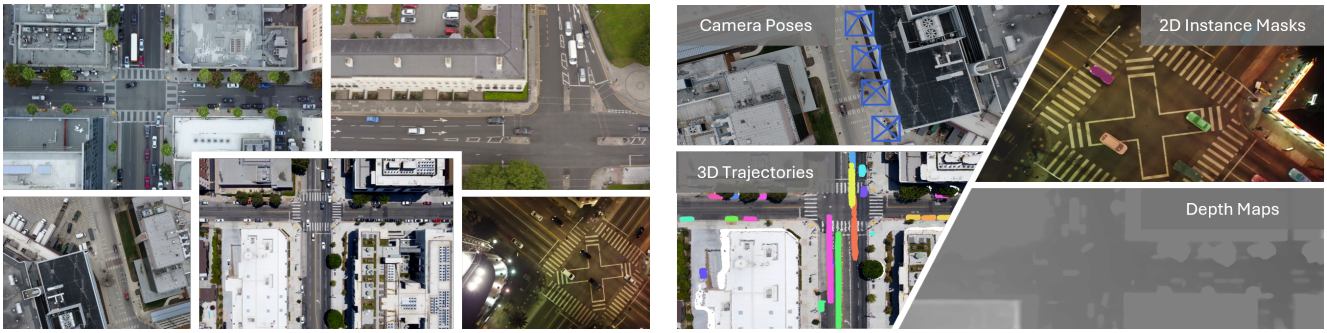
A. Demonstration videos

In Fig. 5, we compare the input aerial sequences with the rendered videos produced by our reconstructed models for the Downtown-High and Intersection-Day scenes. The animations show that our method accurately recovers dynamic scenes with diverse object scales. Together with Fig. 1, these results demonstrate that our approach robustly reconstructs dynamic urban scenes from aerial videos across varying illumination conditions, flight altitudes, and scene dynamics.

B. Method

Local ratio field. In the Monocular Geometry Lifting module, we resolve the scale variation of monocular depth through a local depth-ratio field. For each refined 3D keypoint, we compute the ratio between its triangulated depth and the corresponding monocular estimate, forming a sparse set of scale anchors. For any pixel, its refined depth is computed as:

$$\tilde{D}(x) = \rho(x)\, D(x), \tag{1}$$



(a) Aerial scenes **(b)** Annotations

Figure 6. Overview of Aero4D dataset. The dataset spans urban scenes with broad variations in altitude, UAV flight speed, illumination, and traffic density. Each sequence provides geometric and semantic supervision for both static structures and moving objects.

where $\rho(x)$ is obtained by bilinearly interpolating the ratios of the four nearest 3D keypoints. This produces a smooth, spatially varying correction field that enforces scale consistency across all depth maps.

Dynamic height estimation. We regress object height using a small MLP $h = f_\theta([w, \ell])$, where w and ℓ denote the footprint width and length. The network uses two fully connected hidden layers with 64 units and ReLU [1] activations and outputs a single scalar height value. We train this MLP offline using 3D bounding-box annotations from CARLA [2].

Ground plane estimation. For physics-guided optimization, we estimate a local ground plane Γ_t around each dynamic instance using the refined 3D points obtained from monocular geometry lifting. The plane is represented as $n_t^\top x + d_t = 0$, where n_t is the plane normal and d_t is the offset. RANSAC [3] is applied to fit the plane parameters robustly. To enhance stability and efficiency, the plane is fitted within an enlarged $10\text{m} \times 10\text{m}$ spatial window centered at each dynamic instance. The resulting normal n_t is further employed as the reference orientation underlying the upright stability regularization.

C. Dataset

We provide additional details of our constructed Aero4D dataset. We curate five representative aerial urban scenarios spanning different flight altitudes, UAV flight speeds, illumination conditions, and traffic dynamics. The sequences cover both daytime and nighttime flights, and they capture vehicles approaching, slowing down, and accelerating through intersections, providing a diverse set of appearance and motion patterns. Note that long single-pass flight sequences are avoided, as insufficient side-overlap typically introduces accumulated pose drift and leads to a curved ground plane in the reconstruction. Correcting such distortion requires additional ground control points, which falls outside the scope of this work. Illustrations of the dataset are provided in Fig. 6. For each frame, we introduce the camera pose and depth map, and for dynamic objects, we additionally supply 2D instance masks, 3D bounding boxes, and temporally consistent trajectories.

D. Experiments

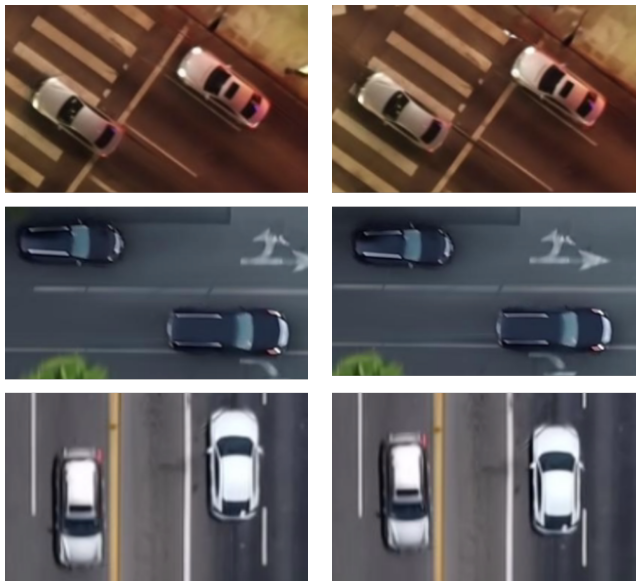
Table 4. **Dynamic reconstruction accuracy.** Comparison between reconstructed dynamic object trajectories and ground-truth motion on the UAV3D [4] dataset.

	Mean Error ↓	Standard Deviation ↓
3D Position Error (m)	1.24	1.06
Depth Error (m)	0.91	0.78
Depth Error Ratio	1.1%	0.9%
Lateral Error (m)	0.57	0.32
Yaw Error (°)	3.51	2.80

Dynamic object reconstruction accuracy. We evaluate the reconstruction accuracy of dynamic objects on the UAV3D [4] dataset. As shown in Tab. 4, our physically consistent formulation yields a low depth error ratio of 1.1%, indicating that the depth ambiguity inherent to monocular aerial views is effectively suppressed. This correction is essential. Small moving vehicles occupy only a few pixels, and monocular depth often underestimates their distance, pushing their reconstructed centers below the ground. Pure photometric supervision is largely insensitive to such misplacements, allowing the optimization to converge to a degenerate solution that either ignores these objects or keeps them in invalid 3D locations. Even when the rendered images appear correct, the underlying 3D positions may remain floating or misaligned. With the proposed physical constraints, dynamic objects are consistently aligned with the ground and exhibit compact 3D positions and stable orientations.

Table 5. **Dynamic PSNR comparison.** Dynamic PSNR measured on training and evaluation renderings.

	Dyn-PSNR (Train) ↑	Dyn-PSNR (Evaluation) ↑
Intersection-Night	27.15	17.65
Downtown-High	26.43	22.47
Intersection-Day	26.61	21.75



(a) Training rendering **(b)** Evaluation rendering

Figure 7. Visual comparison between training renderings and evaluation renderings. Dynamic objects exhibit consistent visual appearance across both settings despite the drop in Dyn-PSNR on evaluation views, confirming that the degradation originates from interpolation deviations rather than reduced reconstruction fidelity.

Dynamic PSNR analysis. To further assess the PSNR discrepancy in dynamic regions, we evaluate dynamic PSNR separately on the training renderings and the evaluation renderings. The results are summarized in Tab. 5. Dynamic PSNR decreases by more than 4 dB on the evaluation views compared with the training renderings, while the perceived visual quality remains largely unchanged, as illustrated in Fig. 7. This behavior suggests that the PSNR drop is primarily attributed to interpolation deviations when rendering moving objects at unseen timestamps, rather than to an actual degradation in reconstruction fidelity.

Runtime analysis. We evaluate the computational cost of AeroDGS on the Downtown-High scene using an RTX 6000 Ada GPU, and present the results in Tab. 6. Monocular geometry lifting accounts for a notable portion of the runtime, as it establishes reliable initialization for both static backgrounds and dynamic objects under monocular aerial observations. Peak GPU memory usage occurs during dynamic instance tracking, and the relatively high demand may limit deployment on memory-constrained hardware, indicating a potential direction for future improvement. The rendering speed reaches 22.1 FPS at 4K resolution, which is moderate given the large Gaussian count required by wide-area aerial scenes and the overhead introduced by high-resolution rendering.

Table 6. **Runtime analysis.** Results are reported on the Downtown-High scene using an RTX 6000 Ada GPU.

Scene	Downtown-High	
Resolution	3840×2160	
Stage	**Time (min)**	**GPU Memory (GB)**
Monocular geometry lifting		
Depth estimation	1.5 (2.3%)	10.9
Keypoint tracking	1.3 (2.0%)	25.3
Dynamic instance segmentation	2.9 (4.4%)	13.4
Dynamic instance tracking	3.7 (5.7%)	44.9
Geometric Consolidation	13.0 (19.9%)	2.7
Gaussian optimization	43.0 (65.7%)	8.3
Total	65.4 (100%)	–
Rendering	22.1 FPS	–

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 2

[2] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16. PMLR, 2017. 2

[3] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[4] Hui Ye, Raj Sunderraman, and Shihao Ji. Uav3d: A large-scale 3d perception benchmark for unmanned aerial vehicles. In *The 38th Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 2