AeroGS: Scale-Aware Gaussian Splatting for Pose-Free Dynamic UAV Scene Reconstruction

Supplementary Material

A. Implementation Details

In this section, we provide the complete training algorithm, detailed network specifications, the mathematical formulation of our B-Spline basis, and specific hyperparameters to support reproducibility.

Training Pipeline. We summarize the complete training pipeline of AeroGS in Algorithm 1. The process consists of an initialization phase using MASt3R [9] for the first few frames, followed by an incremental sliding window optimization that incorporates our three decoupling modules: Ego-Object Decoupling, Appearance-Deformation Decoupling, and Scale-Complexity Decoupling. The algorithm concludes with a final global refinement step to optimize all camera poses and S^2A-Anchors jointly.

Algorithm 1 AeroGS Training Pipeline

Require: Monocular UAV video $\{I_t\}_{t=1}^{T}$
Ensure: Camera poses $\{P_t\}$, S^2A-Anchors $\mathcal{A}$
Initialization: Estimate $\{P_t\}_{t=1}^{N_{init}}$ via MASt3R, initialize $\mathcal{A}$
for $t = N_{init} + 1$ to T **do**
  Estimate P_t via PnP-RANSAC, expand $\mathcal{A}$, define window W
  for $iter = 1$ to K_ℓ **do**
    M1: Ego-Object Decoupling
    $\alpha_i \leftarrow \text{MLP}_{filter}(f_i, \gamma(\mu_i), \gamma(t))$
    $\mathcal{L}_{static} \leftarrow \sum \alpha_i \cdot \mathcal{L}_{recon}, \mathcal{L}_{motion} \leftarrow \sum (1-\alpha_i) \cdot \mathcal{L}_{cycle}$
    M2: Appearance-Deformation Decoupling
    $\{c_t, o_t, s_t, q_t\} \leftarrow D(f_i, t)$
    $\tilde{\mu}_i(t) \leftarrow \mu_i + (1-\alpha_i) \cdot \text{HSTD}(f_i, \gamma(t), \gamma(s_i))$
    Optimize: $\mathcal{L}_{total} = \mathcal{L}_{static} + \lambda_{motion}\mathcal{L}_{motion} + \lambda_{entropy}\mathcal{L}_{entropy} + \lambda_{sparse}\mathcal{L}_{sparse}$
  end for
  M3: Scale-Complexity Decoupling
  if complexity$(A_i) > \tau_{complex}$ **then**
    Split A_i into 8 children at finer scale
  end if
  if $t \mod N_{sync} == 0$ **then**
    Global sync for K_g iterations
  end if
end for
Final Refinement: Optimize all $\{P_t\}$, $\mathcal{A}$ for K_r iterations
return $\{P_t\}$, $\mathcal{A}$

Network Architectures. Our framework, AeroGS, employs lightweight Multi-Layer Perceptrons (MLPs) for various decoupling tasks. The detailed architectures are as follows:

- **Anchor Filter (MLP_{filter}):** Used to predict the static confidence α_i. It takes the canonical feature f_i (32D), positional encoding of the anchor center $\gamma(\mu_i)$ (24D), and temporal encoding $\gamma(t)$ (4D) as input. It consists of two hidden layers with 64 channels and a Sigmoid activation at the output.
- **Appearance Decoder ($\mathcal{D}$):** Maps the canonical feature f_i to Gaussian attributes. It takes f_i and frame index embedding as input. It is a 2-layer MLP with 64 hidden units, outputting color c_t (SH coefficients), opacity o_t, scale s_t, and rotation q_t.
- **HSTD Control Point Network ($F_{ctrl}^{(l)}$):** Predicts the control points for B-spline deformation. Inputs include f_i, $\gamma(t_{norm})$, and the scale encoding $\gamma(s_i)$. It uses 3 hidden layers of width 128 to ensure sufficient capacity for capturing complex deformations.
- **Motion Modulation Network (F_{mod}):** A lightweight 2-layer MLP (32 hidden units) that takes f_i and the velocity v_i as input to output the modulation factor $m_i \in [-1, 1]^3$.

Explicit B-Spline Matrix Formulation. In the HSTD module, we employ B-spline curves as the temporal basis to ensure temporal smoothness and local control. Following the efficient matrix formulation utilized in recent trajectory modeling works [8], we precompute the basis functions to accelerate training. Given a B-spline of order k, the position $\mathbf{p}(u)$ at normalized time u within a knot interval $[t_j, t_{j+1})$ is computed as:

$$\mathbf{p}(u) = [1, u, u^2, \ldots, u^{k-1}]\mathbf{M}_k[\mathbf{p}_{j-k+1}, \ldots, \mathbf{p}_j]^T, \quad (1)$$

where $u = \frac{t-t_j}{t_{j+1}-t_j} \in [0, 1)$. The matrix $\mathbf{M}_k$ is constant. For the cubic B-splines ($k = 4$) used in our base deformation level, the basis matrix is:

$$\mathbf{M}_4 = \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}. \quad (2)$$

This matrix formulation allows us to implement the temporal deformation as a simplified tensor operation on the GPU, significantly reducing the computational overhead compared to recursive De Boor's algorithm.

Attribute Inheritance in Densification. As described in the main paper, we employ a complexity-aware densification strategy. When an anchor A_i is identified for splitting (based on Eq. 13 in the main paper), it spawns eight child anchors. The attribute inheritance follows these rules:

1. **Static Attributes:** The child anchors inherit the parent's canonical feature f_i and static confidence α_i. This preserves the learned semantic information (e.g., whether the region is likely background) and appearance priors.
2. **Dynamic Parameters:** The motion parameters, including the base velocity $\theta_{dyn,i}$ and the HSTD control points $P_i^{(l)}$, are re-initialized to zero or small random values. This is crucial because the child anchors are intended to capture fine-grained residual deformations that the coarser parent anchor could not resolve. Inheriting coarse deformation parameters would hinder the optimization of high-frequency local details.
3. **Spatial Distribution:** The positions of the eight children are initialized at the centers of the eight sub-quadrants of the parent's voxel.

B. Experimental Setup Details

Datasets. We evaluate our model on four diverse datasets to verify robustness across varying scales and scenarios:

- **VisDrone** [17]: We utilize 8 challenging sequences capturing urban landscapes from varying altitudes. This dataset features numerous small moving agents (vehicles and pedestrians) and large camera motions, serving as a primary benchmark for reconstruction quality.
- **UAVDT** [3]: We select 6 sequences characterized by complex flight altitudes and diverse camera angles. This dataset is particularly crucial for evaluating the efficacy of our Scale-Complexity Decoupling module under significant scale changes.
- **KITTI Odometry** [5]: Following the protocol of PVG [2], we use sequences 0001, 0002, 0006 and 0007 to quantitatively evaluate trajectory estimation accuracy (ATE/RPE) in dynamic driving environments.
- **Au-air** [1]: We employ this dataset for ablation studies due to its distinct mix of low-altitude static terrain and dynamic traffic, which facilitates the analysis of component contributions.

Baselines. We compare AeroGS against two categories of state-of-the-art methods.

- **Pose-Required Methods:** We compare with dynamic reconstruction approaches including 4DGS [15], Grid4D [16], DeGauss [13], S3Gaussian [6], PVG [2], AD-GS [8], and DeSiRe-GS [11]. Since ground-truth poses are unavailable for the UAV datasets, we provide these methods with poses estimated by COLMAP [12] (for VisDrone) or PI3 [14] (for UAVDT).
- **Pose-Free Methods:** We benchmark against recent unposed methods: HT-3DGS [7], LongSplat [10], and CF-3DGS [4]. To ensure a fair comparison, all pose-free baselines are initialized using the same protocol as ours (using MASt3R for the first $N_{init} = 3$ frames).
- **CF-3DGS+4DGS:** We additionally implement a two-stage baseline that first runs CF-3DGS [4] to extract poses and then trains 4DGS [15] with these fixed poses. This baseline is designed to demonstrate the necessity of our joint optimization framework compared to a naive sequential pipeline.

Figure 1. Qualitative ablation of temporal basis on UAVDT dataset.

C. Additional Results

C.1. Ablation of B-Spline Temporal Basis

To validate the HSTD module, we compare the proposed B-spline temporal basis with an MLP-based deformation field and Fourier feature encoding. As reported in Table 1 and visualized in Figure 1, the three approaches exhibit distinct behaviors.

Table 1. Ablation of temporal basis on UAVDT dataset.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
Ours (MLP-based)	26.85	0.831	0.210
Ours (Fourier)	27.12	0.840	0.195
Ours (B-Spline)	**27.77**	**0.852**	**0.185**

The MLP-based deformation produces overly smoothed motion fields, which blur the shapes of moving vehicles, especially under rapid acceleration or sudden turns. Fourier features, while able to represent high-frequency signals, introduce noticeable ringing artifacts and background noise, degrading the reconstruction of static structures such as road surfaces and building edges. In contrast, the B-spline representation strikes an effective balance between flexibility and stability: it captures local high-frequency motion while preserving temporal smoothness. This enables AeroGS to reconstruct sharper object boundaries and more stable background geometry, confirming the suitability of

B-splines for modeling rigid and near-rigid motion trajectories.

C.2. Qualitative Comparisons on UAV Datasets

We further evaluate AeroGS on three UAV datasets with diverse altitudes, object scales, and motion patterns.

VisDrone. As seen in Figure 5, baseline methods such as LongSplat and PVG frequently fail to reconstruct small-scale moving objects, leading to ghosting artifacts or incomplete vehicle shapes. This degradation is more pronounced in low-resolution regions with small pedestrians and cars. AeroGS produces cleaner temporal consistency and sharper outlines, owing to its Scale–Complexity Decoupling which assigns higher deformation capacity to fine-scale anchors.

UAVDT. Figure 6 highlights the effect of large altitude variation and perspective changes. Methods including Grid4D and 4DGaussians exhibit distortions in road markings and inconsistent geometry across adjacent views, particularly in low-texture road regions. AeroGS maintains coherent lane boundaries and consistent road surfaces, demonstrating the benefits of jointly adjusting deformation capacity and anchor density based on spatial scale.

Au-Air. In Figure 7, AeroGS achieves clearer separation between static terrain and dynamic traffic compared to DeSiRe-GS and AD-GS. Competing methods exhibit motion leakage into static regions, often misclassifying parked vehicles or vegetation. The learned static confidence effectively routes gradients to the appropriate anchors, resulting in more stable static reconstructions and cleaner dynamic layers.

C.3. Driving Scenarios (KITTI)

We additionally evaluate AeroGS on the KITTI benchmark, which presents long trajectories and complex vehicle motions. Figures 3 and 4 compare AeroGS with state-of-the-art pose-free baselines including LongSplat [10], CF-3DGS [4], and HT-3DGS [7]. These methods often produce curved or discontinuous lane markings and blurry vehicle structures due to entangled camera and object motion. By explicitly decoupling these motions, AeroGS reconstructs straight road geometry and produces sharper vehicle contours, achieving visual quality comparable to methods relying on accurate poses. Failure cases illustrated in Figure 2 indicate that extremely fast motion or abrupt appearance changes remain challenging, consistent with our discussion in Section D.

C.4. Rendering Speed and Memory

We benchmark AeroGS on an NVIDIA RTX 3090 GPU. For a 300-frame UAV sequence at 1920×1080 resolution, training completes within 3–4 hours, including pose refinement. During inference, AeroGS achieves real-time rendering at 45 FPS with peak memory usage around 14GB. This indicates that despite its additional motion decoupling and scale-aware modeling, AeroGS remains computationally feasible for interactive visualization and offline reconstruction tasks.

D. Limitations

While AeroGS achieves state-of-the-art performance in joint pose-dynamics optimization, we acknowledge specific limitations in handling extreme dynamic scenarios, as visualized in Figure 2. Specifically, for objects moving at extreme speeds (illustrated in the second row), the large inter-frame pixel displacement can exceed the effective capture range of our self-supervised cycle consistency loss, causing the optimization to fall into local minima and resulting in severe motion blur or ghosting artifacts. Furthermore, as shown in the first row, the method struggles with objects that appear suddenly or are visible for an extremely brief duration (e.g., less than 10 frames). In such cases, the insufficient temporal observations prevent the model from accumulating adequate gradients to robustly learn the canonical features and temporal deformations, leading to incomplete geometry or flickering artifacts. Future work could address these challenges by incorporating long-range 2D feature matching or few-shot generative priors.

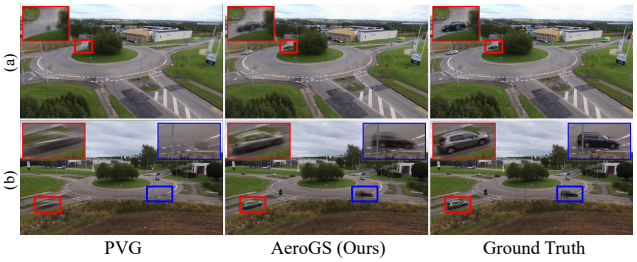

Figure 2. Qualitative comparisons on the Au-Air [1] dataset showing sudden appearance (a) and fast motion (b).

References

[1] Ilker Bozcan and Erdal Kayacan. Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8504–8510. IEEE, 2020. 2, 3, 5

[2] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv:2311.18561*, 2023. 2

[3] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–386, 2018. 2, 5

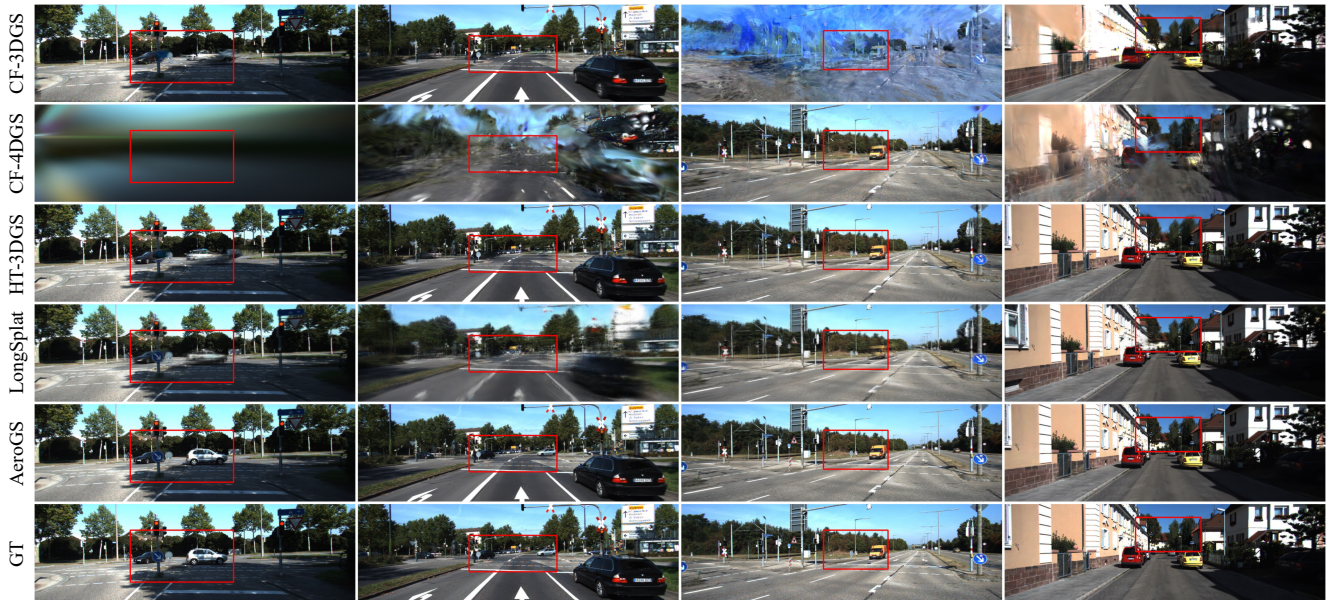

Figure 3. Qualitative comparisons on the KITTI [5] dataset with pose-free methods.

Figure 4. Qualitative comparisons on the KITTI [5] dataset.

[4] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20796–20805, 2024. 2, 3

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 4

[6] Nan Huang, Xin Wei, Wenzhao Zheng, Peng An, Min Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shaoqing Zhang. S^3gaussian: Self-supervised street gaussians for autonomous driving, 2024. 2

[7] Bo Ji and Angela Yao. Sfm-free 3d gaussian splatting via hierarchical training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21654–21663, 2025. 2, 3

[8] Xu Jiawei, Deng Kai, Fan Zexin, Wang Shenlong, Xie Jin, and Yang Jian. AD-GS: Object-aware B-Spline Gaussian splatting for self-supervised autonomous driving. *International Conference on Computer Vision*, 2025. 1, 2

[9] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024. 1

[10] Chin-Yang Lin, Cheng Sun, Fu-En Yang, Min-Hung Chen, Yen-Yu Lin, and Yu-Lun Liu. Longsplat: Robust unposed 3d gaussian splatting for casual long videos. In *ICCV*, 2025. 2, 3

[11] Chensheng Peng, Chengwei Zhang, Yixiao Wang, Chenfeng Xu, Yichen Xie, Wenzhao Zheng, Kurt Keutzer, Masayoshi Tomizuka, and Wei Zhan. DeSiRe-GS: 4d street gaussians for static-dynamic decomposition and surface reconstruction

Figure 5. Qualitative comparisons on the VisDrone [17] dataset.

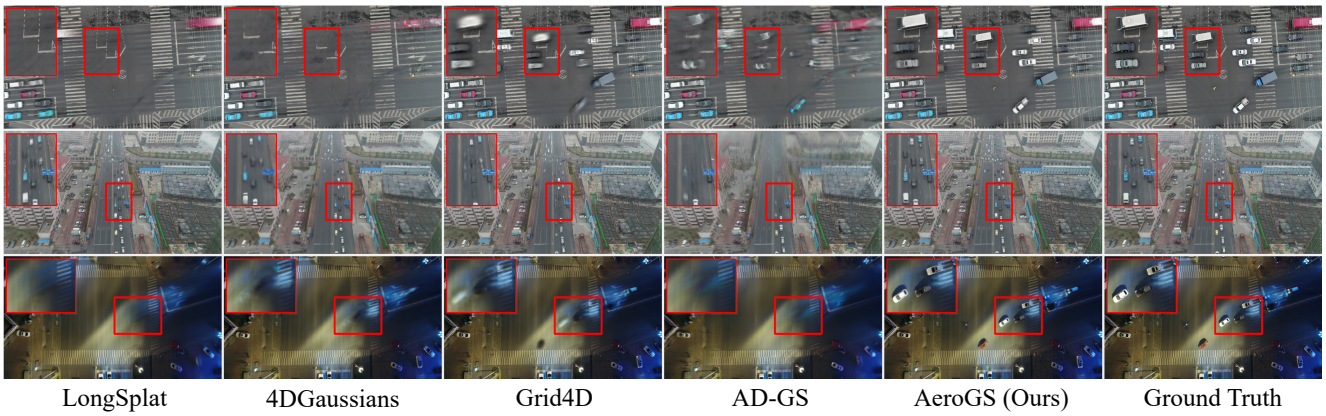

Figure 6. Qualitative comparisons on the UAVDT [3] dataset.

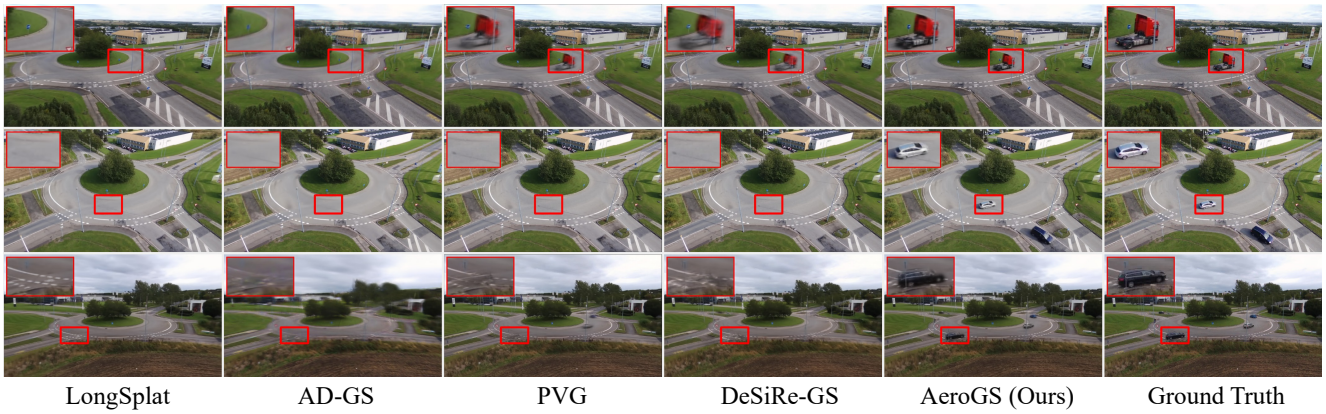

Figure 7. Qualitative comparisons on the Au-Air [1] dataset.

for urban driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[12] Johannes L Schonberger and Jan-Michael Frahm. Structure-

from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2

[13] Rui Wang, Quentin Lohmeyer, Mirko Meboldt, and Siyu Tang. Degauss: Dynamic-static decomposition with gaussian splatting for distractor-free 3d reconstruction, 2025. 2

[14] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning, 2025. 2

[15] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 2

[16] Jiawei Xu, Zexin Fan, Jian Yang, and Jin Xie. Grid4D: 4d decomposed hash encoding for high-fidelity dynamic gaussian splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[17] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2021. 2, 5