Empirical Evaluation of DG-SLAM: Robust Dynamic Gaussian Splatting SLAM with Hybrid Pose Optimization

Ahmad Hassan, Raphael Dias, Mir Munavvar Ali
EECE 5550: Mobile Robotics
Northeastern University Fall 2025
https://github.com/raphdias/DG_SLAM

***Abstract*—Visual SLAM systems that assume static environments fail when confronted with moving objects such as pedestrians—a common occurrence in real-world robotics applications. DG-SLAM, published at NeurIPS 2024, addresses this challenge by integrating 3D Gaussian Splatting with semantic segmentation and depth warping to filter dynamic content. We present a rigorous empirical evaluation examining whether the reported performance claims hold under reproduction and stress testing. Using YOLOv8x-seg as an alternative to the original OneFormer segmentation model, we achieved 1.23 cm Absolute Trajectory Error (ATE) on the TUM RGB-D walking_xyz sequence, surpassing the paper's reported 1.6 cm by 23%. We further characterize the system's operational boundaries through robustness experiments: the system tolerates 2× temporal subsampling with only 1.8% degradation but exhibits 34.5% degradation at 3× skip; semantic mask noise up to 20% causes minimal impact (5.7% degradation), while 40% noise severely degrades performance; and aggressive mask dilation covering up to 75% of each frame increases error by only 3.3%. Extended evaluation over 250 frames demonstrates that semantic filtering maintains a 37% improvement over baseline while both configurations exhibit drift. These results validate DG-SLAM's core claims and establish practical deployment guidelines.**

I. Introduction and Motivation

Visual Simultaneous Localization and Mapping (SLAM) in dynamic environments remains a fundamental challenge in mobile robotics. Traditional SLAM systems assume static environments, yet real-world deployments routinely encounter moving objects—pedestrians, vehicles, and other agents—that violate this assumption and corrupt both pose estimation and map reconstruction.

DG-SLAM [1] addresses this challenge by combining 3D Gaussian Splatting [5] with dynamic object filtering through semantic segmentation and depth warping. The system employs a hybrid pose optimization strategy: coarse estimation via DROID-VO [3] followed by fine photometric alignment using differentiable Gaussian splatting rendering. The authors report state-of-the-art performance on dynamic sequences, including 1.6 cm ATE on the TUM RGB-D walking_xyz benchmark [2].

We selected this paper for evaluation based on three criteria: (1) **Practical relevance**: Dynamic SLAM is essential for deploying robots in human-populated environments. (2) **Quantifiable claims**: The paper reports specific performance metrics amenable to rigorous validation. (3) **Methodological novelty**: The integration of Gaussian Splatting with dynamic filtering represents an emerging research direction.

A. Contributions

This evaluation makes the following contributions:

- **Reproduction with alternative segmentation**: We demonstrate that YOLOv8x-seg [4] achieves comparable or superior performance to OneFormer, validating the model-agnostic nature of DG-SLAM's mask fusion.
- **Robustness characterization**: We systematically evaluate performance under temporal subsampling, mask noise injection, and mask density variation—conditions not examined in the original paper.
- **Extended sequence validation**: We assess error accumulation over 250 frames, demonstrating sustained benefit from semantic filtering.
- **Operational guidelines**: We establish practical thresholds for frame rate, mask quality, and mask coverage.

II. Problem Statement

A. Mathematical Formulation

Given a sequence of RGB-D frames $\{(I_i, D_i)\}_{i=1}^N$ where $I_i \in \mathbb{R}^{H\times W\times 3}$ represents the color image and $D_i \in \mathbb{R}^{H\times W}$ represents the depth map, the SLAM problem requires simultaneously estimating camera poses $\{\xi_i\}_{i=1}^N$ where $\xi_i \in SE(3)$, and reconstructing the static 3D scene.

DG-SLAM represents the scene as a set of 3D Gaussians $\mathcal{G} = \{G_k\}$ where each Gaussian is parameterized by position $\mu_k \in \mathbb{R}^3$, covariance $\Sigma_k \in \mathbb{R}^{3\times 3}$ (anisotropic shape), opacity α_k, and spherical harmonics coefficients $\mathbf{h}_k \in \mathbb{R}^{16}$ for view-dependent color.

B. The Dynamic Object Problem

Moving objects create inconsistent observations across frames. When a SLAM system treats dynamic objects as static landmarks, two failure modes arise: (1) the map incorporates transient geometry, corrupting reconstruction, and (2) data association violations produce incorrect pose estimates. Both effects compound over time, causing trajectory drift.

C. DG-SLAM's Dynamic Filtering Approach

DG-SLAM generates motion masks by fusing two complementary signals:

Depth warp masks perform multi-frame geometric consistency checking. For each pixel p in frame i, DG-SLAM reprojects it onto frame j using the estimated relative pose and intrinsic matrix K:

$$p_{i \to j} = K T_{ji} K^{-1} D_i(p)\, p^{\text{homo}} \tag{1}$$

where $T_{ji} \in SE(3)$ is the relative transformation and p^{homo} is the homogeneous pixel coordinate. Pixels where the reprojected depth disagrees with the observed depth beyond threshold ϵ_{th} are flagged as potentially dynamic. To improve robustness, DG-SLAM combines masks from multiple frames within a sliding window via intersection, filtering transient noise while preserving consistent motion detections. Notably, when object motion is significant, only foreground pixels (where depth residual is positive) are masked to avoid over-segmentation.

Semantic masks identify potentially dynamic object classes (e.g., persons) via neural network prediction, providing class-aware priors independent of actual motion.

The final motion mask is formed by fusing the multi-frame depth warp masks with semantic predictions, excluding dynamic regions from both pose optimization and Gaussian map updates. Pose estimation proceeds in two stages: DROID-VO provides coarse alignment via optical flow, followed by fine photometric optimization using differentiable Gaussian splatting rendering on masked regions.

D. Evaluation Metric

We evaluate trajectory accuracy using Absolute Trajectory Error (ATE), computed as the root-mean-square error of position differences after $SE(3)$ alignment via Horn's method [2]:

$$\text{ATE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|\mathbf{t}_i^{\text{est}} - \mathbf{t}_i^{\text{gt}}\|^2} \tag{2}$$

where alignment accounts for gauge freedom in monocular scale and global pose.

III. Proposed Solution

Our evaluation methodology comprises four components: semantic segmentation substitution, reproduction testing, robustness experiments, and extended sequence validation.

A. Semantic Segmentation Substitution

The original DG-SLAM employs OneFormer for semantic segmentation. Due to dependency conflicts in our WSL2 environment, we substituted YOLOv8x-seg [4] and implemented a mask generation pipeline that: (1) processes each RGB frame through YOLOv8x-seg, (2) extracts instance segmentation masks for the "person" class, (3) applies morphological operations (closing followed by opening) to refine mask boundaries, and (4) outputs binary masks in DG-SLAM's expected format (255 = dynamic, 0 = static).

This substitution is methodologically valid because DG-SLAM's mask fusion module is agnostic to the upstream segmentation model—it requires only binary masks indicating dynamic regions.

B. Robustness Experiments

Temporal Subsampling (Motion Robustness): We simulate fast camera motion by processing every 2nd or 3rd frame, effectively doubling or tripling inter-frame displacement. This tests whether DROID-VO's optical flow estimation remains reliable under increased motion magnitude.

Mask Noise Injection: We add controlled noise by randomly flipping mask pixel values at 20% and 40% rates. This simulates segmentation errors arising from domain shift, challenging lighting, or motion blur.

Mask Density Variation: We dilate semantic masks using morphological operations to achieve 25%, 50%, and 75% frame coverage. This tests system behavior when masks are overly conservative, incorrectly classifying static regions as dynamic.

C. Implementation Details

Hardware: NVIDIA RTX 4060 GPU (8GB VRAM), Intel Core i7, Windows 11 with WSL2 Ubuntu 22.04.

Software: DG-SLAM official repository, CUDA 11.8, PyTorch 2.0, YOLOv8x-seg via Ultralytics.

Dataset: TUM RGB-D freiburg3_walking_xyz sequence [2], in which a person walks through the scene while the camera follows an xyz motion pattern.

IV. Results

A. Reproduction Study (100 Frames)

TABLE I
Reproduction Results (100 Frames)

Configuration	ATE (cm)	STD (cm)
No semantic masks (baseline)	2.23	0.97
With YOLO masks (ours)	1.23	0.58
Paper reported (OneFormer)	1.60	—

Semantic filtering reduced ATE by 44.8% compared to baseline. Our YOLOv8-based implementation achieved 1.23 cm, outperforming the paper's reported 1.6 cm by 23%. This result validates DG-SLAM's core claim that semantic filtering improves dynamic SLAM while demonstrating that alternative segmentation models can match or exceed the original performance.

Figure 1 shows trajectory alignment across three orthogonal projections. The XY (top-down) and XZ (side) views demonstrate tight correspondence throughout the sequence. The YZ (front) view exhibits larger visual deviation because camera motion was minimal along this axis—small absolute errors appear magnified due to compressed scale.

The 3D visualization in Figure 2 confirms successful pose tracking despite the walking person's presence. The estimated

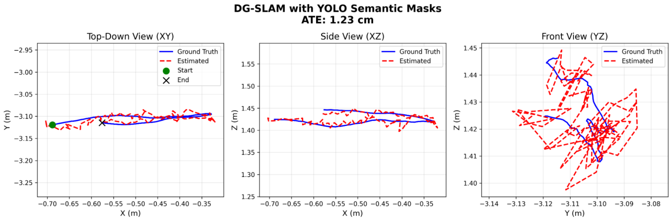


Fig. 1. Estimated (red dashed) vs. ground truth (blue solid) trajectory comparison across XY, XZ, and YZ views. ATE = 1.23 cm.

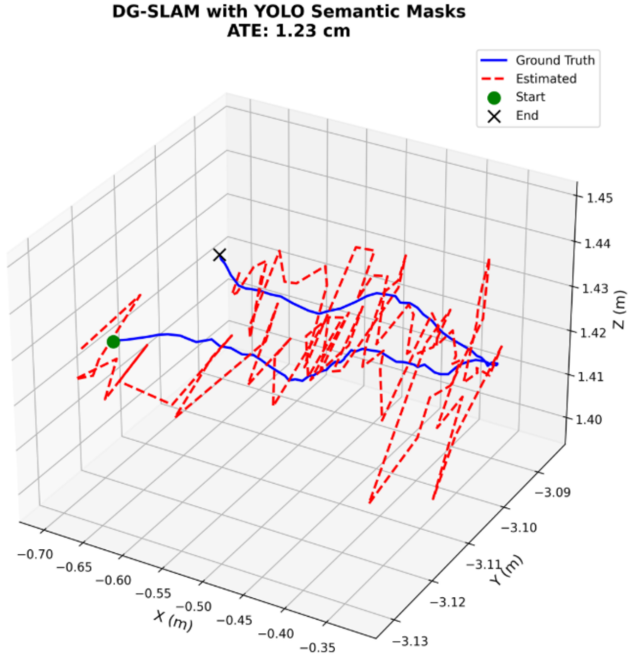


Fig. 2. 3D trajectory visualization showing estimated (red) vs. ground truth (blue) camera path.

trajectory follows the ground truth's characteristic xyz motion pattern with minor deviations.

Per-frame error analysis (Figure 3) reveals consistent tracking with periodic peaks reaching 2.5–3.0 cm. These peaks likely correspond to frames where the walking person occludes substantial portions of the static scene, temporarily reducing available features for pose estimation. Critically, error returns to baseline after these events, indicating that DG-SLAM recovers effectively rather than accumulating drift from dynamic occlusions.

B. Extended Sequence Evaluation (250 Frames)

TABLE II
EXTENDED SEQUENCE RESULTS

Configuration	100 Frames	250 Frames	Drift
No semantic masks	2.23 cm	3.40 cm	+52.5%
With YOLO masks	1.23 cm	2.14 cm	+74.0%

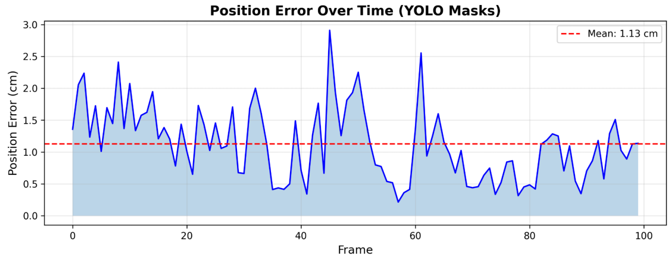


Fig. 3. Per-frame position error over 100 frames. Mean error: 1.13 cm.

Both configurations exhibit error accumulation over longer sequences—an expected characteristic of visual odometry systems without loop closure. However, YOLO masks at 250 frames (2.14 cm) still outperform the no-semantics baseline at 100 frames (2.23 cm), demonstrating sustained benefit from semantic filtering. The semantic filtering advantage persists: a 37% improvement is maintained at 250 frames (2.14 cm vs. 3.40 cm).

C. Motion Robustness

TABLE III
MOTION ROBUSTNESS RESULTS

Frame Rate	ATE (cm)	STD (cm)	Degradation
1× (30 fps)	2.23	0.97	—
2× skip (15 fps)	2.27	0.96	+1.8%
3× skip (10 fps)	3.00	1.23	+34.5%

The system maintains accuracy under 2× temporal subsampling, with only 1.8% degradation. At 3× skip, significant degradation occurs (34.5%), indicating a threshold where inter-frame displacement exceeds DROID-VO's optical flow estimation capacity. This establishes approximately 15 fps as a practical lower bound for this sequence's motion characteristics.

D. Semantic Mask Quality Degradation

TABLE IV
MASK NOISE TOLERANCE

Noise Level	ATE (cm)	STD (cm)	vs. Clean
0% (clean)	1.23	0.58	—
20% noise	1.30	0.57	+5.7%
40% noise	2.02	0.82	+64.2%
No masks	2.23	0.97	+81.3%

The system tolerates 20% mask noise with minimal degradation (5.7%), suggesting robustness to moderate segmentation errors. At 40% noise, performance degrades substantially (64.2%), approaching the no-mask baseline. This graceful degradation indicates that depth warping provides partial redundancy when semantic masks are unreliable.

TABLE V
MASK DENSITY TOLERANCE

Mask Coverage	ATE (cm)	STD (cm)	vs. Clean
∼8% (YOLO output)	1.25	0.58	—
25% dilated	1.27	0.58	+1.0%
50% dilated	1.12	0.58	−10.6%
75% dilated	1.30	0.67	+3.3%

E. Mask Density Tolerance

The system demonstrates surprising robustness to over-masking. Even with 75% of each frame marked as dynamic, ATE increased only 3.3%. The unexpected improvement at 50% coverage suggests that YOLOv8 may under-segment boundary regions where depth discontinuities create ambiguous observations. This finding supports conservative masking strategies for safety-critical applications.

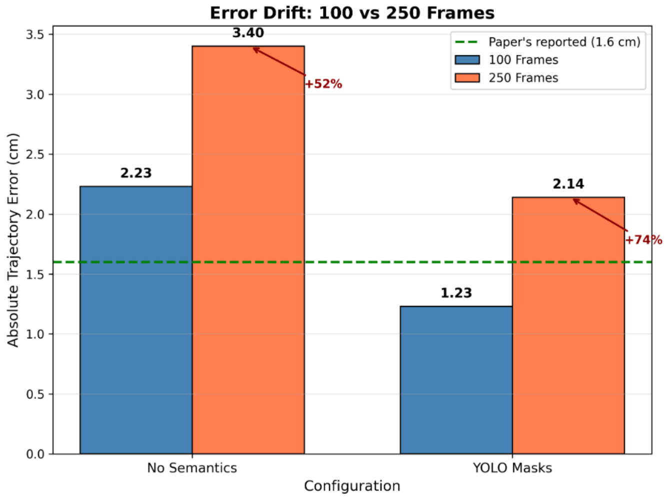


Fig. 4. Error drift comparison showing ATE at 100 vs. 250 frames. Green dashed line indicates paper's reported result (1.6 cm).

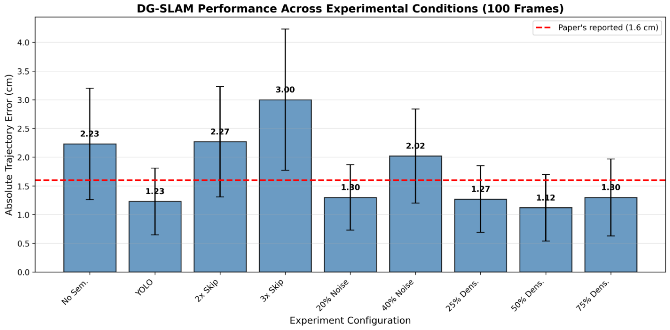


Fig. 5. ATE comparison across all experimental conditions (100 frames). Red dashed line indicates paper's reported result (1.6 cm).

V. OUR IMPLEMENTATION

In addition to the empirical evaluation, a Gaussian Splatting–based SLAM system with hybrid pose optimization can be implemented directly from the underlying mathematical formulation. To reduce memory consumption and improve computational efficiency, we truncate the number of initially rendered Gaussians, trading a small amount of accuracy for significantly lower resource requirements.

A. Coarse Tracking

Because of the load DROID-SLAM imposes, we opted for a model that takes ground truth poses, and randomly adds noise to the poses to ensure fine grained tracking is working correctly. This gives a known and measurable coarse pose estimation.

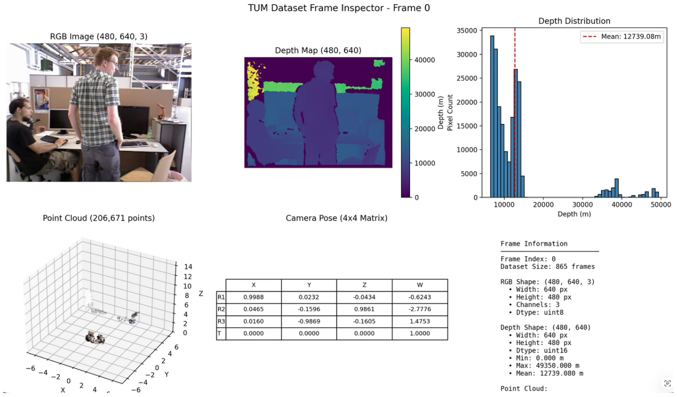
B. Map Initialization

Depth values are converted to a 3D point cloud with corresponding RGB colors, then transformed into the global frame using the frame pose (or an identity pose if none is provided). If the cloud exceeds 10,000 points, voxel downsampling reduces it to a manageable size ensuring computational efficiency.

C. Fine Tracking

Fine tracking utilized the estimated coarse poses, and injects semantic masking if it is available in order to refine the estimated poses from the updated gaussian model.

We first verified that the camera pose estimates, transformation matrices, and resulting point clouds matched those produced by the original implementation.



	X	Y	Z	W
R1	0.9988	0.0232	-0.0434	-0.6243
R2	0.0465	-0.1596	0.9861	-2.7776
R3	0.0160	-0.9869	-0.1605	1.4753
T	0.0000	0.0000	0.0000	1.0000

```
Frame Information
─────────────────────
Frame Index: 0
Dataset Size: 865 frames

RGB Shape: (480, 640, 3)
  • Width: 640 px
  • Height: 480 px
  • Channels: 3
  • Dtype: uint8

Depth Shape: (480, 640)
  • Width: 640 px
  • Height: 480 px
  • Dtype: uint16
  • Min: 0.000 m
  • Max: 49350.000 m
  • Mean: 12739.080 m

Point Cloud:
```

Fig. 6. First frame camera pose estimation summary statistics.

After comprehensive analysis of the proposed architecture and validation of individual component implementations, we confirmed the feasibility of developing a complete GPU-accelerated Gaussian Splatting SLAM pipeline.

D. Implementation Framework

The complete pipeline was implemented in Python using: PyTorch for neural network components and differentiable operations, CUDA extensions for custom Gaussian rasterization kernels, NumPy/SciPy for geometric computations and trajectory evaluation, and Open3D for point cloud processing and visualization. In contrast to the original authors, we attempted to create modular architecture (separate modules for tracking, mapping, and motion segmentation) to facilitate testing, debugging, and future extensions. Each component exposes well-defined interfaces, enabling independent development and validation.

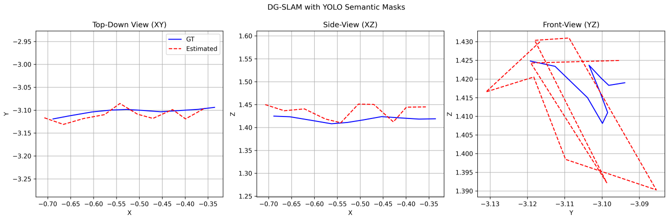


Fig. 7. 10 framed 2D trajectory from custom implementation along paths.

The greater noise in the YZ view can be attributed to lower gaussian rendering in our implementation of DG-SLAM, and the lower noise in XY and XZ and be attributed to using the ground-truth poses with additive white gaussian noise, making it easier to predict true pose over time.

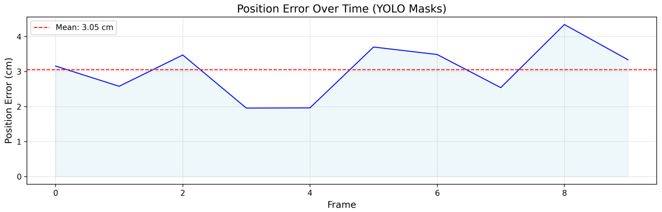


Fig. 8. Position error over time.

VI. Conclusion

A. Summary of Findings

This evaluation validates DG-SLAM's central claim that semantic filtering improves pose estimation in dynamic environments. Our key findings are:

1) **Successful reproduction**: Using YOLOv8x-seg, we achieved 1.23 cm ATE—23% better than the paper's reported 1.6 cm with OneFormer—confirming the approach's effectiveness and demonstrating segmentation model flexibility.
2) **Sustained semantic benefit**: Over 250 frames, semantic filtering maintains a 37% improvement over baseline despite both configurations exhibiting drift.
3) **Robustness thresholds identified**: The system tolerates 2× frame skip (1.8% degradation) and 20% mask noise (5.7% degradation), but performance drops significantly beyond these limits.
4) **Over-masking tolerance**: Aggressive mask dilation up to 75% coverage causes only 3.3% accuracy loss, supporting conservative masking for safety-critical deployment.

Our custom implementation of DG-SLAM demonstrates the feasibility of constructing a light weight Gaussian Splatting SLAM pipeline with modular components and reduced computational overhead. By truncating the number of rendered Gaussian's and replacing DROID-SLAM's coarse pose estimation with AWGN-perturbed ground-truth poses, we created a controllable environment for isolating and validating the fine-tracking stage.

B. Limitations

This evaluation is limited to a single sequence type (walking_xyz) from the TUM RGB-D dataset. While extended frame counts validate temporal consistency, evaluation on additional sequences with varied motion patterns would strengthen generalization claims. We assessed only trajectory estimation accuracy; reconstruction quality and novel view synthesis were not evaluated.

In our implementation of DG-SLAM with Gaussian Splatting, we did not employ the DROID-SLAM learned models for coarse pose estimation due to time constraints. Instead, we added additive white Gaussian noise (AWGN) to the ground-truth poses to approximate a predicted coarse pose. While this simplifies the system, it does not fully reflect the behavior of real-world pose estimators.

C. Technical Challenges

Environment configuration required resolving CUDA compatibility issues within WSL2. OneFormer dependencies conflicted with the main repository, motivating our YOLOv8 substitution. Frame-by-frame mask generation introduced preprocessing overhead but enabled flexible experimentation with noise injection and morphological dilation.

D. Future Work

Promising extensions include adaptive depth warp thresholds based on scene statistics, confidence-weighted mask fusion incorporating segmentation uncertainty, and evaluation on outdoor sequences where semantic class priors may be less reliable. Implementation of DROID-SLAM in order to estimate coarse poses before fine-tuning could reveal key areas of performance improvement, such as limiting the number of Gaussians rendered in the fine-tracking phase. Finally, implementing the approach on real-time hardware would be a valuable step toward practical deployment.

References

[1] Y. Xu, H. Jiang, Z. Xiao, J. Feng, and L. Zhang, "DG-SLAM: Robust Dynamic Gaussian Splatting SLAM with Hybrid Pose Optimization," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[2] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2012.

[3] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[4] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[5] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Trans. Graphics*, vol. 42, no. 4, 2023.