

FlareGS: 4D Flare Removal using Gaussian Splatting for Urban Scenes

Mayank Chandak¹ Sai Sri Teja Kuppa¹ Rahul Gopi Raju Matta Vinayak Gupta Kaushik Mitra
Indian Institute of Technology Madras, India
{me22b224, ee23s042, me20b145, ee17d021}@smail.iitm.ac.in
vinayakguptapokal@gmail.com, kmitra@ee.iitm.ac.in

Supplementary Material

A. Ablation Study

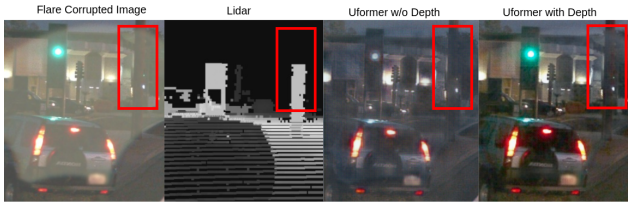


Figure A. Qualitative ablation showing the impact of depth (LiDAR) as an auxiliary input.

As shown in Fig. A, incorporating LiDAR (depth) information alongside RGB input significantly improves reconstruction in regions heavily obstructed by flare artifacts. Depth provides a flare-invariant prior, enabling the model to better infer occluded object boundaries.

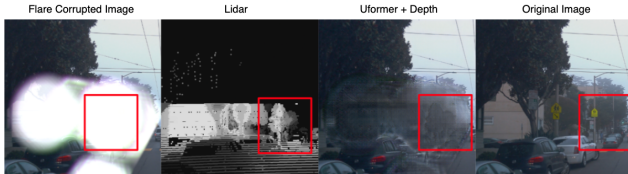


Figure B. Extreme flare case where RGB input lacks information, but depth preserves structural cues.

In Fig. B, the input RGB frame is synthetically corrupted with a large, opaque ghost flare for stress testing. While most scene content is visually obscured, the LiDAR modality preserves geometric contours of occluded structures such as the tree. Notably, a streetlight present in the RGB input—positioned in front of the tree—is absent in the LiDAR signal. Consequently, the Uformer output reconstructs the tree’s structure but not the streetlight, suggesting that the model relies on flare-free depth cues for structural inference under extreme visual degradation.

¹Equal contribution

Light Source	R	G	B
Cyan Atmos.	0–95	158–255	160–255
Warm Street	255	69–255	0
Red Emergency	178–255	0–99	0–71
Cool Blue	0–135	0–206	180–255
Green Neon	0–124	205–255	0–154
Purple Atmos.	128–221	0–160	128–226

Table A. Flare color presets used for synthetic augmentation. Each RGB range corresponds to a common light source encountered in nighttime driving scenes.

B. Flare Color Presets

To simulate realistic nighttime driving conditions, we define a set of flare color presets corresponding to common urban light sources. Each preset specifies a range of RGB values designed to emulate the characteristic hues of real-world illuminants. The Cyan Atmospheric preset captures ambient cool-toned haze, while Warm Street represents sodium-vapor or halogen streetlights with their distinctive orange glow. Red Emergency models flares from traffic signals or emergency vehicles, and Cool Blue emulates cooler flares often seen in sensor or headlight reflections. Green Neon is inspired by traffic signage and urban lighting, whereas Purple Atmospheric introduces magenta-violet hues to diversify the simulation space. These curated presets allow us to generate diverse and visually plausible flare artifacts during training (see Table A).

C. Loss Functions in *FlareGS*

We employ several auxiliary loss [1] functions within the *FlareGS* pipeline to regularize optimization and improve structural fidelity, especially in dynamic urban scenes.

Opacity Entropy Regularization. To encourage sharper opacity boundaries and suppress over-smoothed regions, we

apply an entropy-based regularization:

$$\mathcal{L}_{\text{opacity}} = -\mathbb{E} [\hat{\alpha} \log(\hat{\alpha})], \quad (1)$$

$$\text{where } \hat{\alpha} = \text{clamp}(\text{opacity}, \varepsilon, 1 - \varepsilon). \quad (2)$$

Inverse Depth Smoothness Loss. We impose spatial smoothness on inverse depth while preserving edges guided by image gradients:

$$\mathcal{L}_{\text{inv-depth}} = \left\| \nabla \left(\frac{1}{\hat{D} + \varepsilon} \right) \cdot \exp(-\|\nabla I_{\text{flare}}\|) \right\|_1. \quad (3)$$

Affine Regularization Loss. We regularize the affine transformations to remain close to the identity:

$$\mathcal{L}_{\text{affine}} = \|A - I\|_1 + \|t\|_1, \quad (4)$$

where $A \in \mathbb{R}^{3 \times 3}$ and $t \in \mathbb{R}^3$ denote the affine rotation and translation, respectively.

Dynamic Region Reconstruction Loss. To improve fidelity in dynamic regions (e.g., moving vehicles or pedestrians), we apply an L1 loss restricted to high-opacity dynamic regions:

$$\mathcal{L}_{\text{dyn}} = \left\| \hat{I}_{\text{flare}} - I_{\text{flare}} \right\|_1, \quad \text{for pixels where } \hat{\alpha}_{\text{dyn}} > \tau. \quad (5)$$

Gaussian Prior Regularization. We regularize per-class Gaussian components (e.g., vehicles, roads, sky) using category-specific priors:

$$\mathcal{L}_{\text{gauss}} = \sum_{c \in \mathcal{C}} \mathcal{L}_{\text{reg}}^{(c)}, \quad (6)$$

where $\mathcal{L}_{\text{reg}}^{(c)}$ is computed independently for each semantic class c .

Training Setup. Both Uformer and *FlareGS* were trained using an NVIDIA RTX 4090 GPU. Uformer was trained for 100 epochs, and *FlareGS* for 30,000 iterations.

Uformer Loss Weights. The loss for Uformer combines reconstruction, perceptual, adversarial, and feature losses: $\lambda_{L1} = 100$, $\lambda_{\text{recons}} = 1$, $\lambda_{\text{vgg}} = 10$, $\lambda_{\text{adv}} = 0.01$, and $\lambda_{\text{feat}} = 1$.

FlareGS Loss Weights. The total loss is a weighted sum of photometric, geometric, and perceptual objectives: $\lambda_{\text{rgb}} = 0.8$, $\lambda_{\text{ssim}} = 0.2$, $\lambda_{\text{depth}} = 0.01$, $\lambda_{\text{opacity}} = 0.05$, $\lambda_{\text{inv-depth}} = 0.0$, $\lambda_{\text{affine}} = 0.00001$, $\lambda_{\text{flare}} = 0.8$, $\lambda_{L1} = 100$, $\lambda_{\text{recons}} = 1$, $\lambda_{\text{vgg}} = 10$, $\lambda_{\text{adv}} = 0.01$, and $\lambda_{\text{feat}} = 1$.

References

- [1] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnire: Omni urban scene reconstruction. In *The Thirteenth International Conference on Learning Representations*, 2025.