Supplemental Materials for BAGS: Blur Agnostic Gaussian Splatting through Multi-Scale Kernel Modeling

Cheng Peng*, Yutao Tang*, Yifan Zhou, Nengyu Wang, Xijun Liu, Deming Li, and Rama Chellappa

Johns Hopkins University, Baltimore MD 21218, USA
{cpeng26,ytang67,yzhou223,nwang43,xliu253,dli90,rchella4}@jhu.edu

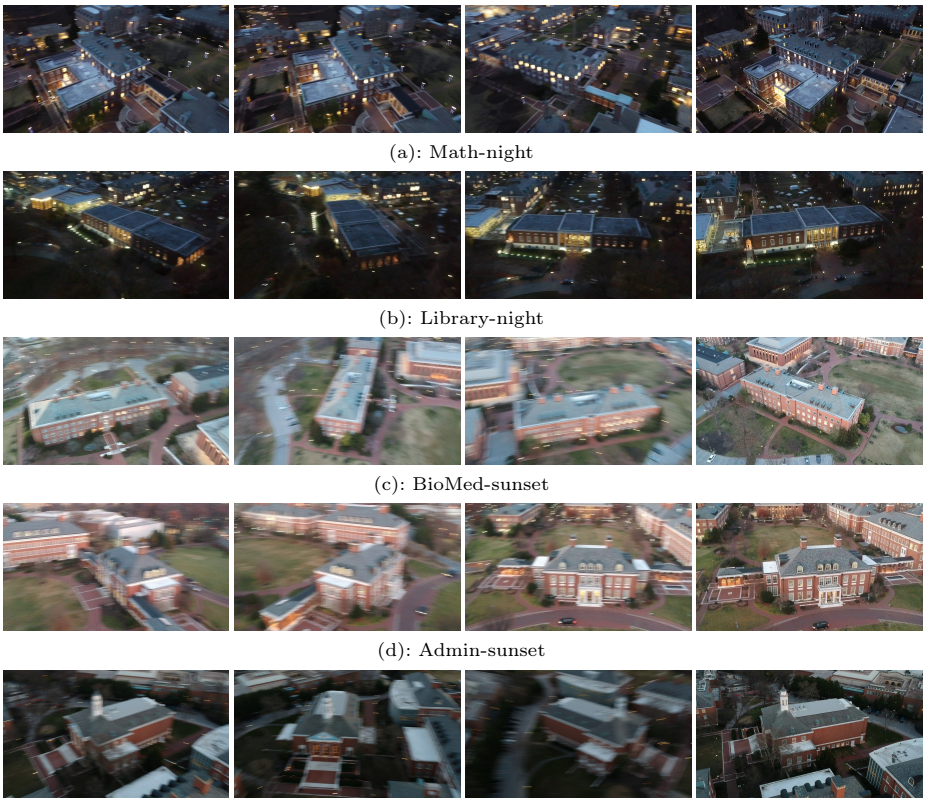


Fig. 8: Visualization of our five collected unbounded drone dataset. Each row showcases a different scene with three training views and one test view.

*Equal contribution.

1 Unbounded Drone Dataset Details

We collect a blurry dataset captured by a drone in 360 unbounded format, which provides more realistic scenes for evaluation in unconstrained pose geometry. This dataset contains 5 scenes of different buildings which are all collected at sunset or night. Specifically, we collect 96, 96, 112, 112, and 84 training images for each building. For each scene, we also collect clear images from a stable drone position for testing. Visualization of the collected dataset is shown in Fig. 8. The first three columns show the training images and the last column shows the testing images. We note that the two scenes collected at night have less blur, as the drone flies at slower speed compared to at sunset due to various constraints and safety considerations. This explains the lesser PSNR improvement of BAGS compared to the baselines. All data and implementation code will be publicly released upon acceptance.

2 Additional Implementation Details

In BPN's $\mathcal{F}_{\text{feat}}$, ReLU activation function and Instance Normalization [8] are used for the intermediate layers. We project the pixel coordinates to a sinusoids space with the dimension of 16 and use a learnable view embedding of size 32. For multi-scale training, we train the first two scales with 3000 steps, and the highest resolution scale with a maximum of 30000 steps for 360 unbounded scenes; however, forward-bounded scenes converge much earlier than that. We set the loss weights to be $\lambda_{\text{photo}} = 0.8$, $\lambda_{\text{DS}} = 0.2$, and $\lambda_{\text{mask}} = 0.001$.

We train NeRFacto in Nerfstudio [7] as a baseline method on mix resolution and low light motion blur scenarios for 30k iterations with its default setting, *i.e.* a batch size of 4096 rays with 96 and 48 samples in the first and second iteration of the proposal sampler, respetively. Adam [1] optimizer is used with an exponential decay learning rate schedule from 0.01 to 0.0001.

3 Deblurring at High Resolution

Following the discussion in the main manuscript, we have explored ways to reduce the computational cost of BAGS at very high resolution. To this end, we employ a simple yet effective extension of BAGS based on sub-pixel convolution [6]. The modified BAGS has three operations:

1. Pixel unshuffling, which converts spatial resolution to the channel dimension, *i.e.* rearranging an image of shape $C \times H \times W$ into $Cr^2 \times \frac{H}{r} \times \frac{W}{r}$, where r is the shuffling factor.
2. Convolution operation, which estimates and performs convolution at lower spatial resolution $\frac{H}{r} \times \frac{W}{r}$.
3. Pixel shuffling, which converts channel dimension back into the spatial dimension, obtaining an image of original resolution.

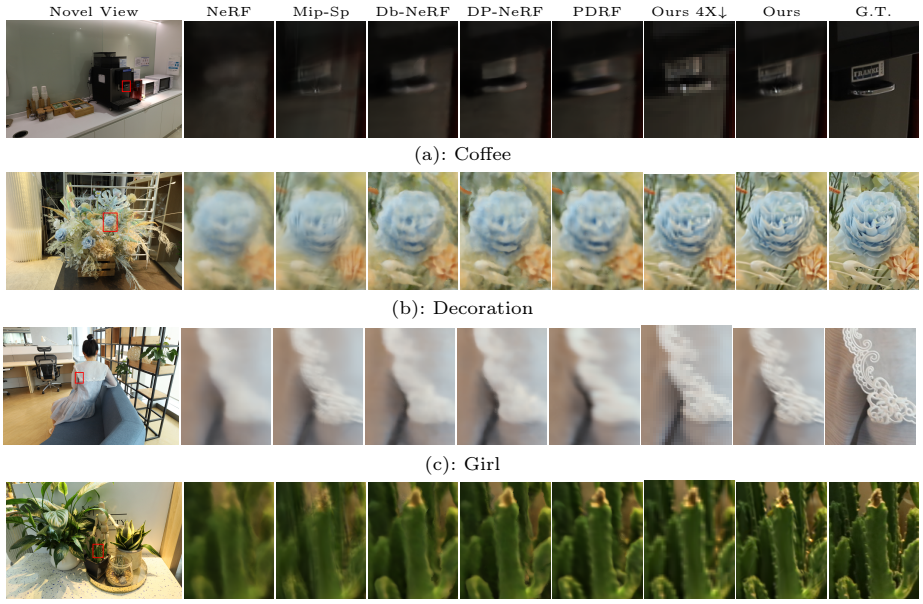


Fig. 9: Visualizations of test views on camera motion blur dataset at 2K resolution. Mip-Sp and Db-NeRF are short for Mip-Splatting [9] and Deblur-NeRF [3]. All methods other than "Ours 4X↓" are trained on 2K resolution.

Concretely, we modify Eq. (5) as

$$\begin{aligned} C', D' &= \mathcal{PUS}(C), \mathcal{PUS}(D), \\ f_{\text{RGBD}}(x,i) &= \mathcal{F}_{\text{feat}}(C'(x,i) \oplus D'(x,i)), \\ h'(x,i), m'(x,i) &= \mathcal{F}_{\text{kernel}}(l(i) \oplus p(x) \oplus f_{\text{RGBD}}(x,i)). \end{aligned} \tag{9}$$

where $\mathcal{PUS}$ denotes the pixel unshuffling operation and $0 \leq m'(x,i) \leq 1$. In our experiments, we use $r = 4$. The convolution operation in Eq. (1) can be modified to be

$$\tilde{C}'(x) = \sum_{x_k \in \mathcal{N}(x)} C'(x_k) h'(x_k), \quad \tilde{C}(x) = \mathcal{PS}(\tilde{C}'(x)). \tag{10}$$

where $\mathcal{PS}$ denotes the pixel shuffling operation. The estimated $\tilde{C}(x)$ would be of the same resolution as the high resolution observation $\tilde{C}_{obs}$. In effect, we estimate one kernel across multiple sub-pixel channels, which greatly reduces cost, as both the spatial resolution of the images and the kernel size are reduced. Intuitively, this also makes sense; *i.e.* in a local $r \times r$ patch, the blur kernel should be relatively consistent.

We present the visualizations of deblurring results for the real scene acquisitions [3] in Fig. 9 and 10. All methods are trained on the original 2K resolution.

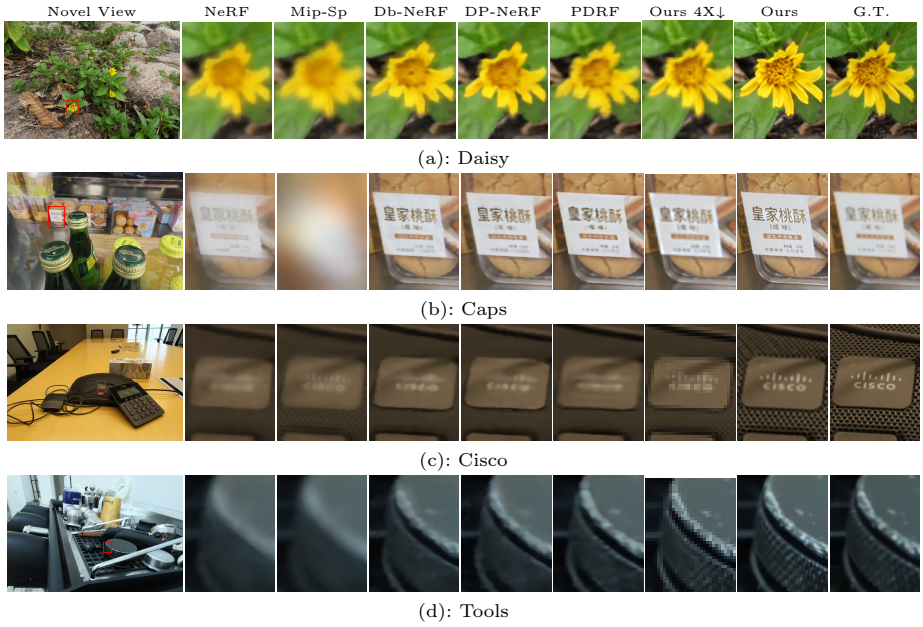
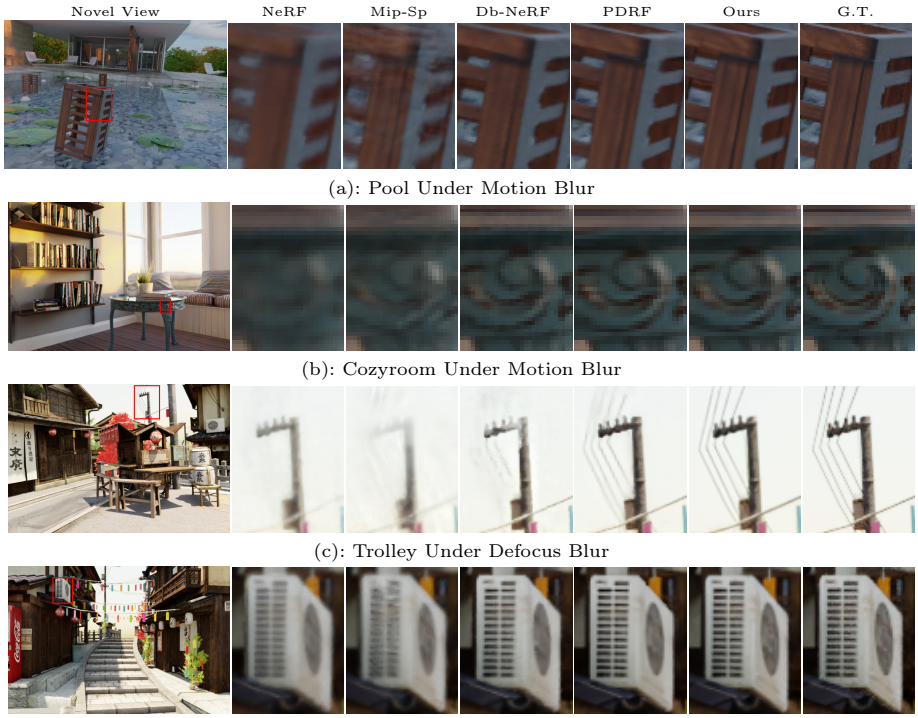


Fig. 10: Visualizations of test views on defocus blur dataset at 2K resolution. All methods other than "Ours 4X↓" are trained on 2K resolution.

We observe that BAGS achieves much sharper results compared to other methods [2–5, 9]. For example, in Fig. 9b, we clearly observe the sharp outlines of the blue flower and its pedals. Additionally, BAGS is capable of reconstructing the fine details whereas the other methods can only capture a contour. For instance, in Fig. 10b and 10c, we can clearly identify the trademarks in our results while the other methods merely show highly blurry silhouettes, especially in Fig. 10c, or jagged representations. Additionally, compared with the results of BAGS from 4X downscaled training images, we can clearly see that the full resolution model is much sharper at details. Quantitatively, we also observe significantly better LPIPS scores from BAGS compared to competing methods, while other metrics like PSNR and SSIM fluctuate based on the quality of the ground truth.

4 Synthetic Scenes

Deblur-NeRF [3] provides a synthetic dataset that contains 5 scenes for both camera motion and defocus blur. Examining the effectiveness of Guassian-Splatting-based methods on synthetic scenes is difficult, as Guassian-Splatting-based methods require point cloud initialization. For synthetic scenes, ground truth camera poses are provided without Structure-from-Motion and thus do not come with a point cloud. On one hand, if we re-calibrate the scene with blurry images, this will lead to suboptimal poses; On the other hand, if we re-calibrate the scene with ground truth images, this will be unfair as the point cloud contains



(a): Pool Under Motion Blur

(b): Cozyroom Under Motion Blur

(c): Trolley Under Defocus Blur

(d): Tanabata Under Defocus Blur

Fig. 11: Visualizations of test views on synthetic datasets. We note that our method and Mip-Splatting use poses estimated from blurry images instead of the ground truth pose.

	Deblur-NeRF			DP-NeRF			PDRF			BAGS		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Factory	28.03	0.863	0.113	29.26	0.879	0.104	30.90	0.914	0.111	30.46	0.926	0.067
Cozyroom	31.85	0.918	0.048	32.11	0.922	0.039	32.29	0.931	0.044	32.06	0.933	0.028
Pool	30.52	0.825	0.190	31.44	0.853	0.156	30.97	0.841	0.191	29.10	0.832	0.114
Tanabata	26.26	0.852	0.100	27.05	0.864	0.078	28.18	0.901	0.078	29.08	0.928	0.047
Trolley	25.18	0.807	0.144	26.79	0.840	0.117	28.07	0.880	0.120	28.12	0.894	0.091
Average	28.37	0.853	0.119	29.33	0.871	0.099	30.08	0.893	0.109	29.76	0.903	0.069

Table 3: Quantitative comparisons on synthetic defocus blur.

pixel information from clear images. We choose the former option as the worst-case scenario for BAGS and present the quantitative comparisons in Table 3 on synthetic defocus blur. Despite using sub-optimal poses, BAGS shows better structural and visual similarity compared to previous methods. However, the performance on synthetic camera motion blur is less consistent. This is likely because images in a few synthetic scenes with camera motion blur are heavily blurred and difficult to calibrate. As we show in all other experiments, this is not an issue for real scenes.

We also provide visualizations of BAGS's results in Fig. 11 and we generally find that BAGS achieves good results in defocus blur. Notably, the synthetic dataset was not constructed with Splatting-based methods in mind, we mainly focus on real world acquisitions in this work.

5 Training Time

On average, BAGS's training time on the camera motion and defocus blur scenes is 0.55 and 0.4 hour respectively. Compared to DeblurNeRF (20 hours), DP-NeRF (37 hours), and PDRF (1.15 hours), BAGS is significantly faster. We note that PDRF leverages explicit representations, similar to InstantNGP, to accelerate training, and is still slower than BAGS. All methods are measured with one NVIDIA A5000 GPU. We note that BAGS reconstructs these scenes without blur and with lower number of Gaussians compared to popular Splatting-based methods; *e.g.*, BAGS achieves 50% lower number of Gaussians compared to Mip-Splatting, as no excessive Gaussians are produced to fit to observed blur.

6 Supplementary Video

We provide rendered videos using our proposed BAGS and, for comparison, using Mip-Splatting [9], for further visualizations. Please refer to the attached videos to observe our compelling results.

References

1. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 2
2. Lee, D., Lee, M., Shin, C., Lee, S.: Dp-nerf: Deblurred neural radiance field with physical scene priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12386–12396 (2023) 4
3. Ma, L., Li, X., Liao, J., Zhang, Q., Wang, X., Wang, J., Sander, P.V.: Deblur-nerf: Neural radiance fields from blurry images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12861–12870 (2022) 3, 4
4. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12346, pp. 405–421. Springer (2020) 4
5. Peng, C., Chellappa, R.: Pdrf: progressively deblurring radiance field for fast scene reconstruction from blurry images. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2029–2037 (2023) 4
6. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016) 2

7. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23 (2023) 2
8. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016) 2
9. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. arXiv preprint arXiv:2311.16493 (2023) 3, 4, 6