

A Implementation Details

Network Architecture. The detailed architecture of Scene Feature Extractor Network and Blur Prediction Network in our framework are illustrated in Figure 1. We enhance the ability to capture high-frequency details by using positional encoding p for pixel coordinates x and a discrete variable embedding module e (implemented with PyTorch) for camera indices i .

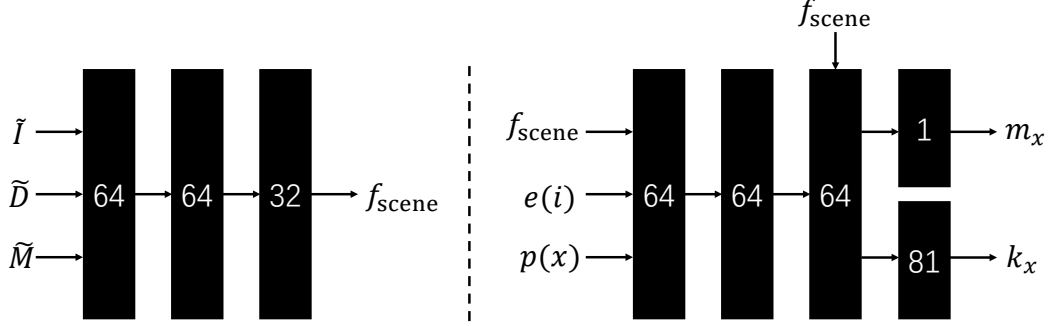


Figure 1: **Scene Feature Extractor Network (Left).** Scene Feature Extractor Network takes rendered image \tilde{I} , rendered depth \tilde{D} and rendered mask \tilde{M} as input, and outputs scene feature f_{scene} . Besides the last layer, each layer outputs 64-dimensional features with ReLU activations. **Blur Prediction Network (Right).** Blur Prediction Network takes scene feature f_{scene} , camera embedding vector $e(i)$ and pixel positional encoding $p(x)$ as input, and outputs blur kernel k_x and blur intensity m_x . Each layer outputs 64-dimensional features with ReLU activations, except for the last layer. Note, in the last layer m_x is obtained via Sigmoid activations, while k_x is obtained via Softmax activations.

B Additional Quantitative and Qualitative Results

We compare our method with dynamic scene reconstruction methods [9, 7] that use video-deblurred images as input. Figure 2 presents a visual comparison of novel view synthesis on the motion blur dataset, where we can see that our method outperforms existing methods that are fed with deblurred images produced by a state-of-the-art video deblurring method. The reason is that video deblurring methods cannot effectively ensure 3D scene consistency in the deblurred images. Please see the video supplementary material for additional novel view synthesis comparison results.

B.1 Novel View Synthesis Comparison

D2RF and DyBluRF Dataset. We compare our approach against BAGS [4] and De3DGS [1], two methods designed to reconstruct sharp static scenes from blurred static images, and evaluate them on the D2RF [2] and DyBluRF [6] datasets. Table 1 and Figure 3 present the comparison results. Clearly, our method demonstrates significant advantages over other methods, producing sharper novel view images while better preserving realistic motion details.

D2RF-v2 and DyBluRF-v2 Dataset. We evaluate our method on two datasets (DyBluRF-v2 and D2RF-v2) with both motion and defocus blur occurring simultaneously. Note that we obtain the DyBluRF-v2 dataset by applying depth-of-field (DoF) rendering technique in Bokehme [5] to the original DyBluRF dataset [6] to simulate defocus blur, and create the D2RF-v2 dataset with motion blur by processing the original D2RF dataset [2] using the motion blur generation method in Davanet [11]. Table 2 present the comparison results. As shown, our method clearly outperforms all the compared methods on the two datasets, verifying the advantage of our method in handling cases with motion and defocus blur occurring jointly.

Deblur-NeRF Dataset. We compare our approach against De3DGS [1], which is designed to reconstruct sharp static scenes from blurred static images, and evaluate them on the Deblur-NeRF [3] dataset. Table 3 and Figure 4 present the comparison results. Clearly, our method demonstrates significant advantages over other methods, producing sharper novel view images.

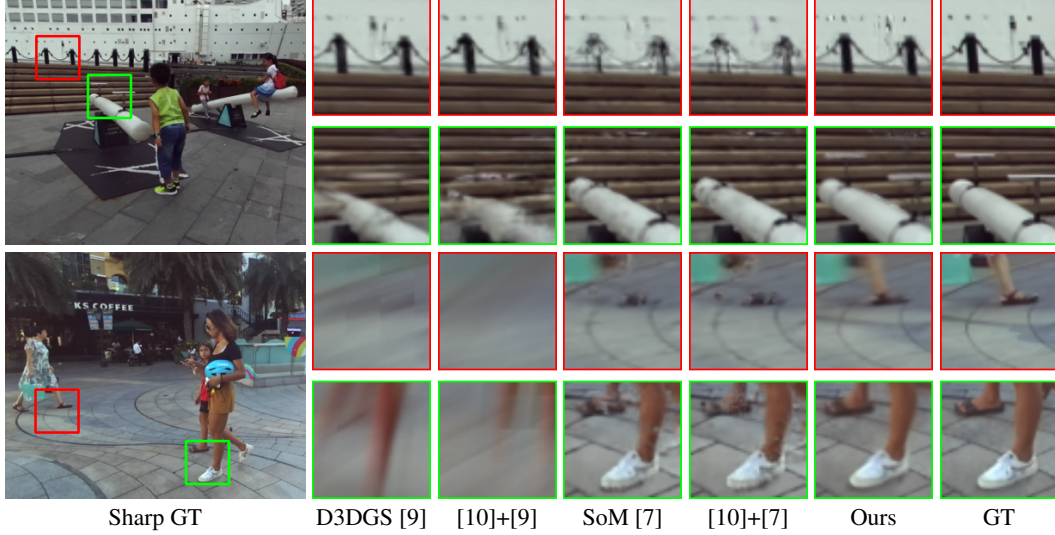


Figure 2: **Visual comparison of novel view synthesis on the DyBluRF motion blur dataset [6].** Here, we also compare with methods fed with deblurred images produced by a state-of-the-art video deblurring method [10] to manifest the effectiveness of our method.

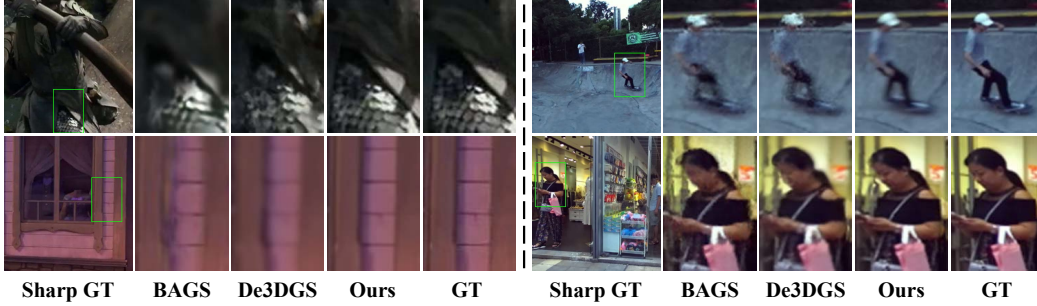


Figure 3: **Visual comparison of novel view synthesis.** Here, we compare our method with methods that are designed to reconstruct sharp static scenes from blurred static scene images. Note, the left column demonstrates results for defocus blur, while the right column presents motion blur outcomes.

Table 1: **Quantitative comparison of novel view synthesis on the D2RF defocus blur dataset [2] and the DyBluRF motion blur dataset [6].**

Method	Defocus Blur			Motion Blur		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
BAGS [4]	24.41	0.730	0.167	24.27	0.723	0.208
De3DGS [1]	23.74	0.716	0.190	22.45	0.689	0.253
Ours	29.39	0.859	0.078	27.01	0.876	0.056

B.2 Deblurring Comparison

We compare the deblurring capability of our method with a broad range of existing methods, including 3DGS- and NeRF-based methods for both dynamic and static scenes [2, 8, 4, 1], as well as transformer-based video deblurring method [10]. Specifically, we compare the sharp images produced at training views. For 3DGS- and NeRF-based methods, these images are rendered from the trained sharp scene representations using the same training views. Table 4 and Figure 5 present the comparison results. Our method outperforms 3DGS- and NeRF-based deblurring approaches and achieves performance comparable to state-of-the-art video deblurring methods.



Figure 4: **Visual comparison of novel view synthesis on the Deblur-NeRF dataset [3].**

Table 2: **Quantitative comparison of novel view synthesis on the D2RF-v2 defocus blur dataset and the DyBluRF-v2 motion blur dataset.** The numerical results of defocus blur are obtained on the Shop and Car scenes of the D2RF-v2 dataset, and the numerical results of motion blur are obtained on the Man and Seesaw scenes of the DyBluRF-v2 dataset.

Method	Defocus Blur			Motion Blur		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
D3DGS [9]	23.66	0.739	0.257	21.75	0.655	0.289
SoM [7]	29.04	0.820	0.094	27.28	0.791	0.103
D2RF [2]	27.82	0.795	0.132	25.89	0.722	0.133
DyBluRF [6]	27.30	0.771	0.150	26.54	0.753	0.112
De4DGS [8]	29.74	0.856	0.078	27.97	0.824	0.087
Ours	30.26	0.885	0.062	28.55	0.859	0.064

Table 3: **Quantitative comparison of novel view synthesis on the Deblur-NeRF dataset [3].**

Method	Defocus Blur			Motion Blur		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
De3DGS [1]	23.71	0.747	0.110	26.61	0.822	0.108
Ours	24.22	0.768	0.095	27.14	0.835	0.096

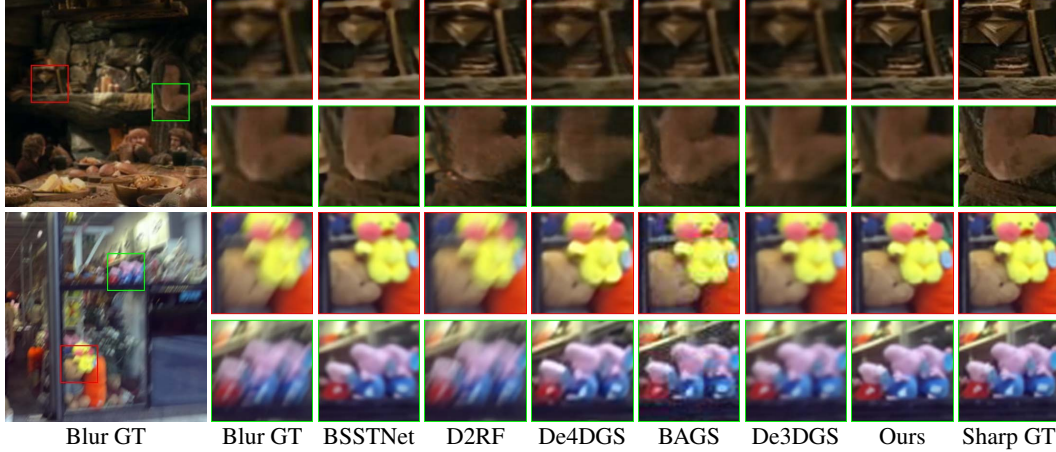


Figure 5: **Visual comparison of deblurring.** Our method enables the synthesis of high-quality deblurring results for videos with defocus blur (top) and motion blur (bottom).

Table 4: **Quantitative comparison of deblurring on the D2RF defocus blur dataset [2] and the DyBluRF motion blur dataset [6].**

Method	Defocus Blur			Motion Blur		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
BSSTNet [10]	33.54	0.961	0.039	33.71	0.965	0.030
D2RF [2]	34.33	0.976	0.028	32.14	0.949	0.049
De4DGS [8]	32.92	0.953	0.044	33.27	0.958	0.035
BAGS [4]	30.69	0.940	0.084	30.55	0.935	0.092
De3DGS [1]	30.36	0.941	0.090	29.84	0.924	0.105
Ours	34.85	0.977	0.027	33.45	0.960	0.036

C Additional Ablation Results

C.1 Ablation on BP-Net

We conduct an ablation study to evaluate the contribution of BP-Net. Specifically, we compare three different blur modeling methods: (i) the motion blur and defocus blur modeling method used in De3DGS [1] (w/ blur modeling in De3DGS [1]), (ii) the motion blur modeling method in De4DGS [8] (w/ blur modeling in Deblur4DGS [8]), and (iii) the defocus blur modeling method in D2RF [2] (w/ blur modeling in D2RF [2]). We report the quantitative results in Table 5, where we can see that our method with the proposed BP-Net produces better results than these alternatives, demonstrating the effectiveness of the BP-Net.

Table 5: **Effect of BP-Net.**

Method	Defocus Blur			Motion Blur		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
w/ blur modeling in De3DGS [1]	28.31	0.812	0.098	26.05	0.823	0.118
w/ blur modeling in De4DGS [8]	28.63	0.829	0.094	26.74	0.859	0.060
w/ blur modeling in D2RF [2]	28.96	0.832	0.094	26.30	0.825	0.109
Ours with BP-Net	29.39	0.859	0.078	27.01	0.876	0.056

C.2 Ablation on Blur Kernel Size

Table 6 further perform quantitative evaluation on how different blur kernel sizes (denoted as K) affect the performance of our method. As shown, a larger blur kernel helps to obtain better results. However, this trend becomes less obvious when K is larger than 9. To balance the performance and the computational cost, we thus choose $K = 9$ as our default choice.

Table 6: **Effect of varying blur kernel size K .** The numerical results of defocus blur are obtained on the Gate and Dock scenes of the D2RF dataset, and the numerical results of motion blur are obtained on the Skating and Man scenes of the DyBluRF dataset.

Blur kernel size	Defocus Blur			Motion Blur		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
$K=5$	27.84	0.817	0.098	29.53	0.906	0.092
$K=7$	28.12	0.836	0.074	29.79	0.913	0.067
$K=9$	28.29	0.842	0.067	30.01	0.921	0.052
$K=11$	28.30	0.843	0.066	30.02	0.920	0.050
$K=13$	28.31	0.842	0.067	30.04	0.922	0.051

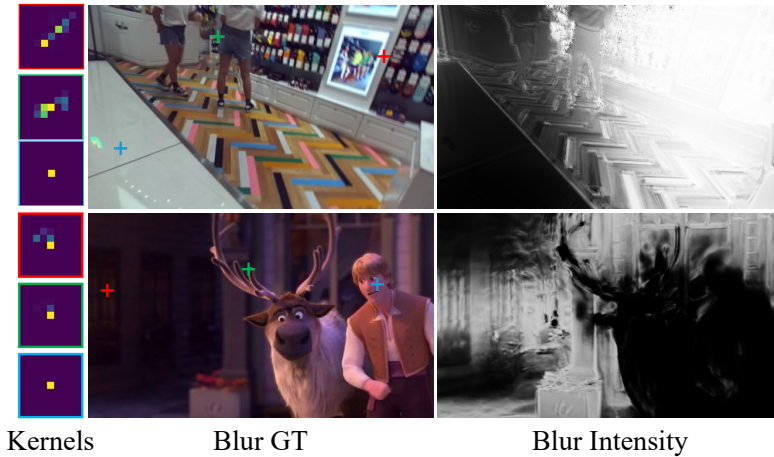


Figure 6: **Visualization of the blur kernel and blur intensity predicted by BP-Net.** Note that the top row shows an image with defocus blur, while the bottom row shows an image with motion blur. In the blur ground truth (GT) image, blue markers indicate pixels with almost no blur, green markers denote pixels with mild blur, and red markers represent pixels with severe blur. A higher blur intensity value corresponds to a more heavily blurred area. Clearly, BP-Net can accurately predict blur regions in images with different types of blur and estimate the corresponding blur kernels at pixel locations with varying blur levels.

D Robustness to Preprocessing and Segmentation Errors

Our method represents image blurring effects using two components: a blur kernel k and a blur intensity m , as illustrated in Figure 6. The blur intensity m effectively emphasizes the spatial regions affected by blur within each training image. Moreover, the type of blur can be intuitively inferred from the estimated kernels. Specifically, kernels corresponding to motion blur capture structured trajectories that reflect the camera’s movement, while those associated with defocus blur present Gaussian-shaped patterns that vary with the distance of the pixel from the focal plane.

To evaluate the accuracy of the blur kernel k predicted by BP-Net, we compare the ground truth blur kernel with the blur kernel predicted by BP-Net on a blurry monocular video dataset with ground truth blur kernel. To this end, we construct two datasets with ground truth blur kernels, referred to as D2RF-v3 and DyBluRF-v3, by randomly sampling two global Gaussian and linear distribution blur kernels of size 9×9 and then respectively applying them to the ground truth sharp images in D2RF [2] and DyBluRF [6] to obtain the corresponding blurry images. With the two datasets, we quantitatively compare our estimated blur kernels and the ground truth blur kernels using PSNR and KL divergence as metrics. Table 7 and Figure 7 present the comparison results. Clearly, our estimated blur kernels are highly similar to the ground truth blur kernels in numerical metrics, manifesting the effectiveness of the BP-Net in predicting different types of blur.

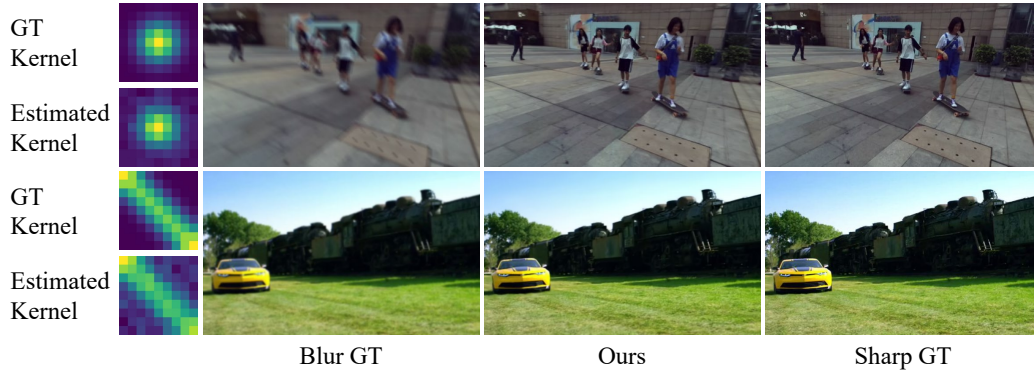


Figure 7: **Visual comparison of estimated kernel on the D2RF-v3 defocus blur dataset and the DyBluRF-v3 motion blur dataset.** Note that the top row shows an image with defocus blur, while the bottom row shows an image with motion blur. Clearly, BP-Net can accurately estimate blur kernels from various types of blurry images and thus recover sharp images.

Table 7: **Comparison of ground truth kernels and estimated kernels.**

	D2RF-v3	DyBluRF-vs
PSNR \uparrow	32.516	29.941
KL Div. \downarrow	0.214	0.247

E Robustness to Preprocessing and Segmentation Errors

In the Table 8, we quantitatively evaluate how errors from external preprocessing steps affect the robustness of our method. To simulate errors from depth estimation, we randomly scale and shift the estimated depth maps within the range of $[0.8, 1.2]$ and $[-20, 20]$, respectively. To simulate errors from 2D point tracking and SAM, we randomly shift the 2D tracking points within the range of $[-30, 30]$, and randomly add or delete five 25×25 mask regions towards the mask predicted by SAM. As shown, our results produced with the artificially perturbed depth, tracking points, and motion mask are comparable to those produced with the originally estimated depth, tracking points, and motion mask, indicating that our method has some tolerance to errors from external preprocessing steps.

Table 8: **Analysis on the impact of errors from external preprocessing steps.**

Method	Defocus Blur			Motion Blur		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Ours w/ depth perturbation	29.19	0.844	0.088	26.79	0.862	0.074
Ours w/ tracking perturbation	29.21	0.854	0.084	26.86	0.874	0.060
Ours w/ mask perturbation	29.25	0.854	0.085	26.74	0.865	0.067
Ours	29.39	0.859	0.078	27.01	0.876	0.056

References

- [1] Lee , B., Lee , H., Sun , X., Ali , U., & Park , E. (2024) Deblurring 3d gaussian splatting. In *ECCV*
- [2] Luo , X., Sun , H., Peng , J., & Cao , Z. (2024) Dynamic neural radiance field from defocused monocular video. In *ECCV*
- [3] Ma , L., Li , X., Liao , J., Zhang , Q., Wang , X., Wang , J., & Sander , P. V. (2022) Deblur-nerf: Neural radiance fields from blurry images. In *CVPR*
- [4] Peng , C., Tang , Y., Zhou , Y., Wang , N., Liu , X., Li , D., & Chellappa , R. (2024) Bags: Blur agnostic gaussian splatting through multi-scale kernel modeling. In *ECCV*
- [5] Peng , J., Cao , Z., Luo , X., Lu , H., Xian , K., & Zhang , J. (2022) Bokehme: When neural rendering meets classical rendering. In *CVPR*
- [6] Sun , H., Li , X., Shen , L., Ye , X., Xian , K., & Cao , Z. (2024) Dyblurf: Dynamic neural radiance fields from blurry monocular video. In *CVPR*
- [7] Wang , Q., Ye , V., Gao , H., Austin , J., Li , Z., & Kanazawa , A. (2024) Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*
- [8] Wu , R., Zhang , Z., Chen , M., Fan , X., Yan , Z., & Zuo , W. (2024) Deblur4dgs: 4d gaussian splatting from blurry monocular video. *arXiv preprint arXiv:2412.06424*
- [9] Yang , Z., Gao , X., Zhou , W., Jiao , S., Zhang , Y., & Jin , X. (2024) Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*
- [10] Zhang , H., Xie , H., & Yao , H. (2024) Blur-aware spatio-temporal sparse transformer for video deblurring. In *CVPR*
- [11] Zhou , S., Zhang , J., Zuo , W., Xie , H., Pan , J., & Ren , J. S. (2019) Davanet: Stereo deblurring with view aggregation. In *CVPR*