Supplementary Material: CasDeblurGS: Cascaded 2D-to-3D Multi-View Consistency for 3D Gaussian Splatting from Two Blurry Images

Haeyun Choi[1*†], Minhyuk Jang[2*], and I-Gil Kim[2]

[1] University of Virginia, Charlottesville, VA, USA
phh3ps@virginia.edu
[2] KT R&D Center, Seoul, Republic of Korea
{minhyuk.jang,i-gil.kim}@kt.com

Supplementary Contents

[*] Equal contribution.
[†] This work was done at KT R&D Center.

1 Additional Implementation Details

In this section, we provide additional implementation details for reproducibility, including dataset construction, frozen modules, stage-wise training, and baseline adaptation.

1.1 Training and Evaluation Data

For training and validation, we use the synthetic camera-motion-blur dataset introduced in DeepDeblurRF [3]. The training split contains 65 scenes, each with 29 viewpoints. For each viewpoint, one blurry image and its corresponding sharp target are provided. The validation split contains 10 additional scenes with the same structure. Blurred images are generated by simulating 6-DoF camera motion during exposure: multiple intermediate views are rendered along a smooth camera trajectory, averaged in linear color space, and paired with the temporally central frame as the ground-truth sharp image.

For evaluation, we use the camera-motion-blur subset of Deblur-NeRF [5], which contains five synthetic scenes and ten real-world scenes. Each scene provides motion-blurred input images and blur-free ground-truth target views for evaluation. To ensure fair and reproducible comparison in the extreme two-view setting, we use a fixed set of per-scene evaluation instances, where each instance is defined by two blurry input views and one held-out target view for novel view synthesis. The complete index mappings are provided in Tab. 1, and all compared methods are evaluated using the same instance definitions. For both training and evaluation, all images are first resized while preserving aspect ratio and then center-cropped to 256×256, following PixelSplat-style preprocessing [1].

1.2 Frozen Modules and OCGM Implementation

We use the width-64 NAFNet [2] pretrained on GoPro [6] as the frozen stabilizer $\mathcal{S}_\phi$, and RAFT-Large [8] from Torchvision [9] for optical flow estimation in OCGM. For each pair of stabilized views, we estimate bidirectional optical flow and compute validity masks using forward–backward consistency together with out-of-bounds checks. A pixel is marked invalid if its mapped coordinate falls outside the image boundary or if the forward–backward residual exceeds the motion-adaptive threshold defined in the main paper. These masks suppress unreliable regions caused by occlusions, motion boundaries, and blur-induced mismatches before masked warps are constructed for Stage 1 guidance. Invalid regions are zeroed out in the warped reference image, and the resulting masked warp together with the validity mask are fed to the Stage 1 deblurring network. This results in more reliable Stage 1 guidance by removing unstable correspondences near occlusions and motion boundaries. The same bidirectional flow fields are used for both validity checking and masked warping.

1.3 Frozen 3DGS Backbone and Input-View Re-Rendering

We adopt the official NoPoSplat [13] checkpoint pretrained on RealEstate10K with two-view inputs at 256×256, and keep it frozen without further fine-tuning. During Stage 2, the intermediate restorations and their camera intrinsics are fed into the backbone to construct a canonical-space 3D Gaussian representation. We then re-render this representation to the two input viewpoints to obtain the global 3D guidance images R_1 and R_2 for final restoration. These rendered RGB images are used directly as Stage 2 guidance. Concretely, the first input view defines the canonical frame, and the second guidance image is rendered at the relative pose inferred by the backbone under the same canonical-space formulation. No ground-truth input-view poses are used. The same backbone is then applied to the restored views and their intrinsics to construct the output 3D representation used for novel view synthesis evaluation.

1.4 Training Details

Only the two deblurring networks, $\mathcal{D}_\theta^{2D}$ and $\mathcal{D}_\psi^{3D}$, are optimized during training. The stabilizer $\mathcal{S}_\phi$, RAFT, and the pose-free 3DGS backbone h_η remain frozen throughout, with no gradients propagated through them. Both trainable networks follow a NAFNet-style encoder–decoder architecture with base width 64. We use an encoder block configuration $[1, 1, 1, 28]$, a single middle block, and a decoder configuration $[1, 1, 1, 1]$. Stage 1 takes a 7-channel input formed by concatenating the 3-channel blurry base view, the 3-channel masked warp, and the 1-channel validity mask, while Stage 2 takes a 6-channel input formed by concatenating the 3-channel blurry base view and the 3-channel re-render guidance. Both networks output a restored 3-channel RGB image.

Training is performed in a stage-wise manner. In Stage 1, the frozen stabilizer and RAFT module are used to construct the masked warps and validity masks, and $\mathcal{D}_\theta^{2D}$ is trained to predict the intermediate restorations $(\tilde{S}_1, \tilde{S}_2)$ from the blurry inputs and OCGM guidance. After Stage 1 training, $\mathcal{D}_\theta^{2D}$ is frozen. In Stage 2, the frozen Stage 1 network first produces $(\tilde{S}_1, \tilde{S}_2)$, which are passed to the frozen pose-free 3DGS backbone to obtain the re-render guidance (R_1, R_2). Using these re-rendered images, $\mathcal{D}_\psi^{3D}$ is trained to predict the final restorations $(\hat{S}_1, \hat{S}_2)$.

For each stage, we use AdamW with learning rate 1×10^{-3}, weight decay 1×10^{-3}, and $(\beta_1, \beta_2) = (0.9, 0.9)$. Each stage is trained for 400k iterations with cosine annealing to $\eta_{\min} = 1 \times 10^{-7}$. We use a batch size of 32 per GPU on two NVIDIA A100 80GB GPUs, with four dataloader workers per GPU. The training objective for both stages combines an L_1 reconstruction loss and a VGG19-based perceptual loss [7, 10]:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\ell_1} + \lambda_p \mathcal{L}_{\text{perc}}, \tag{1}$$

where $\lambda_1 = 0.9$ and $\lambda_p = 0.1$. No style loss is used.

Table 1: Per-scene evaluation index mappings for the real-world and synthetic Deblur-NeRF scenes. Each instance is defined as $\{I_{blur}\} \rightarrow I_{tgt}$, where $\{I_{blur}\}$ denotes the two blurry input views and I_{tgt} denotes the held-out target view for novel-view synthesis.

Scene	Inst. 1	Inst. 2	Inst. 3	Inst. 4	Inst. 5	Inst. 6	Inst. 7
Real-world Scenes							
BlurBall	{1, 2} → 0	{6, 8} → 7	{13, 15} → 14	{19, 20} → 21	–	–	–
BlurBasket	{1, 2} → 0	{10, 11} → 7	{18, 19} → 14	{22, 23} → 21	{27, 29} → 28	{38, 39} → 35	{41, 43} → 42
BlurBuick	{1, 2} → 0	{10, 11} → 7	{15, 18} → 14	{4, 6} → 21	{27, 33} → 28	{36, 38} → 35	{39, 43} → 42
BlurCoffee	{3, 8} → 0	{10, 11} → 6	{1, 17} → 12	{5, 19} → 18	{25, 26} → 24	–	–
BlurDecoration	{1, 2} → 0	{4, 9} → 6	{3, 17} → 12	{19, 33} → 18	{16, 25} → 24	{14, 29} → 30	{31, 40} → 36
BlurGirl	{1, 2} → 0	{6, 8} → 7	{9, 11} → 14	{12, 34} → 21	{18, 29} → 28	{30, 36} → 35	–
BlurHeron	{1, 11} → 0	{7, 15} → 8	{15, 27} → 16	{23, 25} → 24	{19, 33} → 32	–	–
BlurParterre	{2, 9} → 0	{4, 7} → 6	{11, 13} → 12	{19, 20} → 18	{23, 27} → 24	{1, 21} → 30	–
BlurPuppet	{1, 2} → 0	{10, 11} → 6	{7, 8} → 12	{15, 16} → 18	{19, 21} → 24	{27, 29} → 30	{23, 37} → 36
BlurStair	{1, 2} → 0	{7, 9} → 6	{8, 17} → 12	{21, 23} → 18	{16, 26} → 24	{22, 33} → 30	–
Synthetic Scenes							
All Scenes	{1, 2} → 0	{7, 9} → 8	{15, 17} → 16	{23, 25} → 24	{31, 33} → 32	–	–

1.5 Baseline Adaptation Details

All baselines are evaluated under the same fixed two-view protocol using the per-scene input–target index mappings listed in Tab. 1. Methods originally designed for different input regimes are adapted to this protocol while following their official implementations and recommended hyperparameter settings as closely as possible. For methods that require camera poses, we use the benchmark poses provided with Deblur-NeRF [5].

For SE-GS [14], we optimize each scene from the same two blurry source views defined by the protocol. For GAURA [4], we follow the official feed-forward inference pipeline in the two-view setting. For CoherentGS [12], we retain its official alternating optimization framework while restricting the observations to the same two-view protocol.

For the control baselines DAVANet [15] and Difix3D+ [11], the restored outputs are passed to the same frozen pose-free 3DGS backbone [13] used in our method. Thus, the "Pose-Free" and "Generalizable" labels reported in the main paper refer to the complete pipelines formed by each restoration model together with the shared frozen 3DGS backbone, rather than to the restoration models alone.

2 Qualitative Analysis of OCGM Thresholds

Following the OCGM formulation in Eqs. (8)–(10) of the main paper, we qualitatively examine how the minimum tolerance τ and the motion-dependent scale factor α affect the validity mask $M_{b\leftarrow r}$ and the masked cross-view warp $W_{b\leftarrow r}$ used as Stage 1 guidance.

For each example, C_b denotes the stabilized base view, while C_r denotes the stabilized reference view. The unmasked warp $\mathcal{W}(C_r, F_{b\leftarrow r})$ transfers content from C_r into the coordinates of C_b. OCGM retains only the regions considered geometrically reliable and provides the resulting masked warp, together with

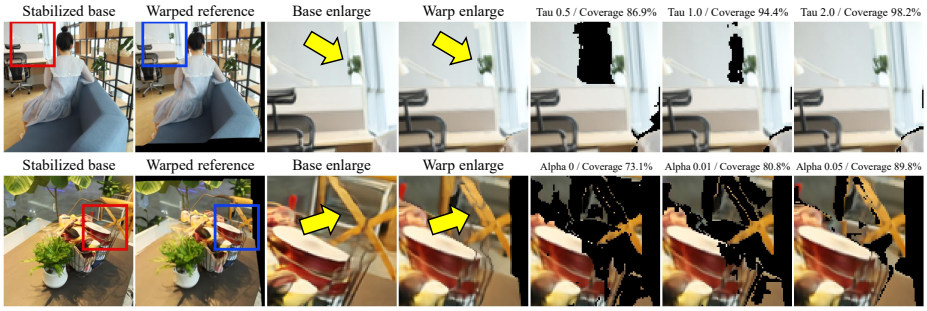

Fig. 1: Qualitative effects of the OCGM thresholds. In each row, the stabilized base view C_b provides the current-view context, while the unmasked warp $\mathcal{W}(C_r, F_{b\leftarrow r})$ transfers complementary content from the stabilized reference view C_r into the base-view coordinates. The first row varies $\tau \in \{0.5, 1.0, 2.0\}$ with $\alpha = 0.01$, whereas the second varies $\alpha \in \{0, 0.01, 0.05\}$ with $\tau = 1.0$. Black regions in the masked guidance indicate rejected correspondences. The reported coverage denotes the retained fraction and is shown only to illustrate the trade-off between rejecting miswarped content and preserving useful cross-view information.

its validity mask, to the Stage 1 restoration network. The stabilizer output is therefore used only as an alignment-friendly observation: Stage 1 combines the original blurry base view with reliable complementary information from the reference view to produce a better restoration than independent stabilization alone.

For each threshold sweep, we keep (C_b, C_r) and their bidirectional RAFT flows fixed while varying one parameter at a time. Thus, the unmasked warp remains identical across the compared settings; only the regions retained as Stage 1 guidance change.

As shown in Fig. 1, the thresholds control the trade-off between rejecting unreliable warps and preserving valid cross-view information. In the BlurGirl example, the unmasked warp incorrectly transfers the highlighted plant tip. The permissive setting $\tau = 2.0$ retains this miswarped structure, whereas $\tau = 1.0$ rejects it while preserving most of the surrounding valid guidance. The more conservative $\tau = 0.5$ also removes the error but discards additional correctly warped content, as reflected by its lower coverage.

The α sweep on BlurBasket shows a similar pattern in a large-displacement region. The setting $\alpha = 0.01$ selectively suppresses the distorted chair geometry while retaining much of the surrounding useful reference content. In contrast, $\alpha = 0$ rejects additional valid chair regions, whereas $\alpha = 0.05$ admits distorted boundaries that should be excluded. Together, these examples illustrate how overly conservative thresholds reduce the amount of usable guidance, while overly permissive thresholds propagate inaccurate reference content.

We use the fixed default $\tau = 1.0$ and $\alpha = 0.01$ for all training and evaluation scenes. These values are not selected as per-scene optima; rather, they provide a practical scene-independent balance between filtering unreliable warps and retaining sufficient guidance across diverse scenes and motion patterns. The

alternative settings are shown only to illustrate OCGM's masking behavior and do not represent separately trained models.

3 Additional Component Diagnostics

Table 2 and Fig. 2 provide complementary quantitative and qualitative diagnostics beyond the ablation in the main paper.

Effect of alignment preconditioning. We estimate RAFT flows and construct OCGM guidance directly from the original blurry views while reusing the same Stage 1 and Stage 2 checkpoints as the full model. This inference-time diagnostic removes the stabilizer only from the flow and guidance construction path, without retraining either restoration network. Compared with the full cascade, PSNR and SSIM decrease by 0.33 dB and 0.020, respectively. These results support the role of the stabilizer in producing alignment-friendly observations for reliable correspondence guidance, rather than serving as the final restoration module.

Effect of global 3D guidance without Stage 1. We additionally evaluate an inference-time 3D-only diagnostic that bypasses the stabilizer, RAFT, OCGM, and the entire Stage 1 restoration process. The original blurry input pair is first passed to the frozen NoPoSplat backbone to produce input-view re-render guidance. The blurry inputs and corresponding re-renders are then processed by the existing Stage 2 checkpoint, and the resulting restorations are used for final reconstruction with the frozen NoPoSplat backbone.

This configuration obtains 20.26 dB PSNR and 0.627 SSIM, substantially underperforming both Stage 1 and the full cascade. The result indicates that global 3D re-render guidance derived directly from blurry inputs cannot replace the locally reliable observations established by Stage 1. Because the Stage 2 checkpoint is reused without retraining, this experiment should be interpreted as an inference-time diagnostic rather than an optimized 3D-only architecture.

Qualitative interpretation. Figure 2 further illustrates how each diagnostic configuration affects the final novel-view reconstruction. Independent stabilization improves each view separately but does not explicitly enforce cross-view consistency, leaving residual discrepancies that are difficult for the frozen pose-free 3DGS backbone to aggregate coherently. Direct RAFT estimates correspondences directly from the original blurry inputs, producing less reliable OCGM guidance. This degrades the Stage 1 restorations and consequently the 3D guidance available to Stage 2.

Stage 1 combines stabilization and OCGM to establish more reliable local correspondences and therefore yields a more coherent reconstruction. However, it lacks the global 3D feedback provided by Stage 2, leaving residual inconsistencies that cannot be resolved through local 2D warping alone. Conversely, 3D-only constructs global guidance directly from the blurry inputs, before locally reliable

Table 2: Extended component diagnostics on real-world Deblur-NeRF scenes. The first four rows reproduce the progressive ablation from the main paper; the last two are inference-time diagnostics using the trained checkpoints without retraining.

Setting	Configuration	PSNR ↑	SSIM ↑
Progressive cascade			
Blurry inputs	No restoration or cross-view guidance	20.07	0.612
Stabilizer	Independent per-view stabilization	20.30	0.655
Stage 1	Stabilizer and OCGM-based local 2D guidance	20.88	0.684
Full cascade	Complete CasDeblurGS pipeline	**21.13**	**0.700**
Additional diagnostics			
Direct RAFT	RAFT and OCGM applied directly to the original blurry views	20.80	0.680
3D-only	Stage 1 bypassed; Stage 2 uses blurry inputs and 3D re-renders	20.26	0.627

Fig. 2: Qualitative novel-view synthesis results for component diagnostics on a synthetic Deblur-NeRF scene. The Inputs column shows the two blurry views; for each reconstruction setting, the lower row enlarges the red-boxed region.

observations have been established. The full cascade addresses both limitations by first improving local cross-view reliability and then refining the restored views using globally aggregated 3D guidance.

4 Sparse Camera Reprojection Analysis

We further describe the sparse reprojection analysis reported in the main paper. We compare the intermediate restorations produced by Stage 1 with the final restorations produced by Stage 2 using the same fixed real-world two-view pairs.

For each restored image pair, we extract OpenCV SIFT features with at most 4,000 keypoints and perform brute-force matching using the ℓ_2 distance. We apply a Lowe ratio threshold of 0.75 and retain only bidirectionally mutual matches. The resulting correspondences are triangulated using `cv2.triangulatePoints` and the benchmark camera projection matrices.

A triangulated point is considered valid if its homogeneous coordinates are finite, its homogeneous denominator is nonzero, and it has positive depth in both camera frames. For each valid point, we compute the symmetric reprojection error as the average pixel ℓ_2 error after reprojecting the point into the two input views.

For each evaluation pair, we measure the number of valid triangulated points, the median symmetric reprojection error, the number of points with reprojection error below one pixel, and the corresponding inlier ratio. The values reported in the main paper are obtained by averaging these per-pair measurements over the 60 real-world evaluation pairs. The reported inlier ratio is therefore the mean of the per-pair ratios, rather than the mean inlier count divided by the mean number of valid points. Benchmark camera poses are used only for this post-hoc evaluation and are never provided as inputs to CasDeblurGS.

5 Runtime Analysis

Although CasDeblurGS employs multiple frozen modules, all inference stages are feed-forward and require no per-scene optimization. Table 3 reports average runtime, PSNR, and SSIM across both the real-world and synthetic Deblur-NeRF subsets. CasDeblurGS improves PSNR by 4.48 dB over GAURA, the fastest baseline, while running 7.5% faster than Difix3D+ and 31.8% faster than CoherentGS, the closest competing methods in reconstruction quality. These results demonstrate a favorable trade-off between reconstruction quality and computational cost.

Table 3: Average runtime and reconstruction quality across both the real-world and synthetic Deblur-NeRF subsets. DAVANet and Difix3D+ are combined with the same frozen NoPoSplat backbone used in CasDeblurGS. Best and second-best results are highlighted in **bold** and underlined, respectively.

Method	Runtime (s) ↓	PSNR ↑	SSIM ↑
SE-GS [14]	<u>28.48</u>	17.33	0.479
GAURA [4]	**25.24**	17.79	0.516
CoherentGS [12]	96.14	20.51	<u>0.674</u>
DAVANet [15] + NoPoSplat	59.37	19.48	0.599
Difix3D+ [11] + NoPoSplat	70.85	<u>20.62</u>	0.642
CasDeblurGS (Ours)	65.52	**22.27**	**0.748**

6 Additional Experimental Results

In this section, we provide per-scene quantitative evaluations and additional qualitative comparisons on Deblur-NeRF to further support the findings in the main paper.

Per-scene quantitative results. Tables 4 and 5 present per-scene quantitative comparisons on the synthetic and real-world Deblur-NeRF datasets, respectively. Consistent with the average gains reported in the main paper, our method achieves the best average performance and remains competitive across individual scenes. These results show that the proposed cascaded 2D-to-3D guidance performs consistently across diverse scene structures and blur patterns.

Additional qualitative results. Figures 3 and 4 provide additional qualitative comparisons that further support the quantitative results. Under the challenging two-view setting with severe blur, existing methods often suffer from structural collapse or ghosting artifacts. For example, in the *BlurFactory* scene, our framework reconstructs the staircase with more coherent geometry and finer texture details, whereas prior 3DGS-based methods struggle to preserve its topology. In the *BlurCoffee* scene, CoherentGS exhibits noticeable color artifacts, while Difix3D+ produces overly blurred results; by contrast, our method recovers clearer text on the notice. Overall, by combining local 2D correspondence guidance with global 3D re-render guidance, our approach improves multi-view consistency and yields higher-quality novel-view synthesis, especially in geometrically challenging regions.

7 Supplementary Video

The supplementary video provides an animated overview of CasDeblurGS, illustrating how local 2D correspondence guidance is progressively complemented by global 3D re-render guidance. It also presents novel-view synthesis results from our method on nine scenes spanning the real-world and synthetic Deblur-NeRF subsets, enabling temporal inspection of rendering sharpness and structural consistency beyond the still-image results in the paper.

Table 4: Quantitative results of novel view synthesis on synthetic scenes of the Deblur-NeRF dataset, reported for each individual scene. The best and second-best results are highlighted in **bold** and underlined, respectively.

Method	Scene	PSNR ↑	SSIM ↑	LPIPS ↓	Method	Scene	PSNR ↑	SSIM ↑	LPIPS ↓
SE-GS	BlurCozy2room	20.65	0.686	0.209	DAVANet	BlurCozy2room	21.83	0.770	0.185
	BlurFactory	17.36	0.420	0.487		BlurFactory	16.27	0.385	0.488
	BlurPool	22.87	0.660	0.265		BlurPool	21.53	0.607	0.270
	BlurTanabata	15.72	0.497	0.377		BlurTanabata	18.93	0.673	0.268
	BlurWine	17.19	0.536	0.353		BlurWine	19.50	0.691	0.247
	Average	18.76	0.560	0.338		Average	19.61	0.625	0.292
GAURA	BlurCozy2room	17.85	0.604	0.345	Difix3D+	BlurCozy2room	23.60	0.801	0.138
	BlurFactory	16.96	0.409	0.504		BlurFactory	18.12	0.479	0.457
	BlurPool	23.04	0.625	0.331		BlurPool	26.33	0.742	0.168
	BlurTanabata	16.08	0.526	0.431		BlurTanabata	19.10	0.682	0.277
	BlurWine	15.63	0.458	0.459		BlurWine	19.35	0.663	0.273
	Average	17.91	0.524	0.414		Average	<u>21.30</u>	0.673	0.262
CoherentGS	BlurCozy2room	24.91	0.828	0.160	**Ours**	BlurCozy2room	25.46	0.866	0.097
	BlurFactory	18.55	0.530	0.385		BlurFactory	23.73	0.818	0.203
	BlurPool	24.33	0.705	0.227		BlurPool	27.14	0.775	0.142
	BlurTanabata	18.56	0.674	0.276		BlurTanabata	19.99	0.748	0.202
	BlurWine	19.21	0.696	0.247		BlurWine	20.75	0.771	0.182
	Average	21.11	<u>0.687</u>	<u>0.259</u>		Average	**23.41**	**0.796**	**0.165**

Table 5: Quantitative results of novel view synthesis on real-world scenes of the Deblur-NeRF dataset, reported for each individual scene. The best and second-best results are highlighted in **bold** and underlined, respectively.

Method	Scene	PSNR ↑	SSIM ↑	LPIPS ↓	Method	Scene	PSNR ↑	SSIM ↑	LPIPS ↓
SE-GS	BlurBall	18.80	0.518	0.404	DAVANet	BlurBall	20.74	0.598	0.342
	BlurBasket	14.91	0.358	0.479		BlurBasket	19.54	0.578	0.378
	BlurBuick	13.93	0.376	0.467		BlurBuick	18.46	0.602	0.340
	BlurCoffee	18.74	0.677	0.351		BlurCoffee	23.11	0.779	0.280
	BlurDecoration	14.10	0.281	0.495		BlurDecoration	16.88	0.457	0.411
	BlurGirl	15.32	0.530	0.359		BlurGirl	19.35	0.713	0.291
	BlurHeron	15.97	0.334	0.413		BlurHeron	18.85	0.487	0.390
	BlurParterre	16.58	0.313	0.454		BlurParterre	19.30	0.468	0.406
	BlurPuppet	14.18	0.284	0.505		BlurPuppet	18.76	0.524	0.343
	BlurStair	16.45	0.304	0.499		BlurStair	18.42	0.527	0.415
	Average	15.90	0.398	0.443		Average	19.34	0.573	0.360
GAURA	BlurBall	20.11	0.569	0.433	Difix3D+	BlurBall	21.35	0.622	0.309
	BlurBasket	13.47	0.406	0.499		BlurBasket	19.98	0.622	0.326
	BlurBuick	14.30	0.383	0.506		BlurBuick	18.45	0.621	0.293
	BlurCoffee	22.53	0.800	0.310		BlurCoffee	23.74	0.815	0.203
	BlurDecoration	16.21	0.433	0.490		BlurDecoration	17.69	0.497	0.369
	BlurGirl	17.23	0.632	0.381		BlurGirl	19.52	0.768	0.227
	BlurHeron	17.57	0.446	0.451		BlurHeron	18.47	0.493	0.373
	BlurParterre	18.86	0.451	0.461		BlurParterre	19.73	0.508	0.366
	BlurPuppet	17.35	0.467	0.454		BlurPuppet	19.25	0.567	0.299
	BlurStair	19.10	0.492	0.456		BlurStair	21.25	0.595	0.346
	Average	17.67	0.508	0.444		Average	<u>19.94</u>	0.611	0.311
CoherentGS	BlurBall	21.75	0.672	0.303	**Ours**	BlurBall	22.87	0.730	0.228
	BlurBasket	20.04	0.668	0.281		BlurBasket	21.00	0.703	0.234
	BlurBuick	16.61	0.546	0.357		BlurBuick	20.17	0.709	0.217
	BlurCoffee	24.78	0.826	0.247		BlurCoffee	25.21	0.859	0.156
	BlurDecoration	16.69	0.517	0.379		BlurDecoration	17.97	0.545	0.309
	BlurGirl	19.04	0.734	0.256		BlurGirl	21.56	0.843	0.158
	BlurHeron	18.88	0.599	0.321		BlurHeron	20.42	0.651	0.260
	BlurParterre	19.72	0.618	0.306		BlurParterre	20.41	0.586	0.281
	BlurPuppet	19.64	0.666	0.267		BlurPuppet	19.87	0.633	0.232
	BlurStair	21.88	0.755	0.204		BlurStair	21.82	0.736	0.209
	Average	19.90	<u>0.660</u>	<u>0.292</u>		Average	**21.13**	**0.700**	**0.228**

Fig. 3: Additional qualitative results of novel view synthesis on synthetic scenes of the Deblur-NeRF dataset. The Inputs column shows the two blurry views; for each method, the lower row enlarges the red-boxed region.

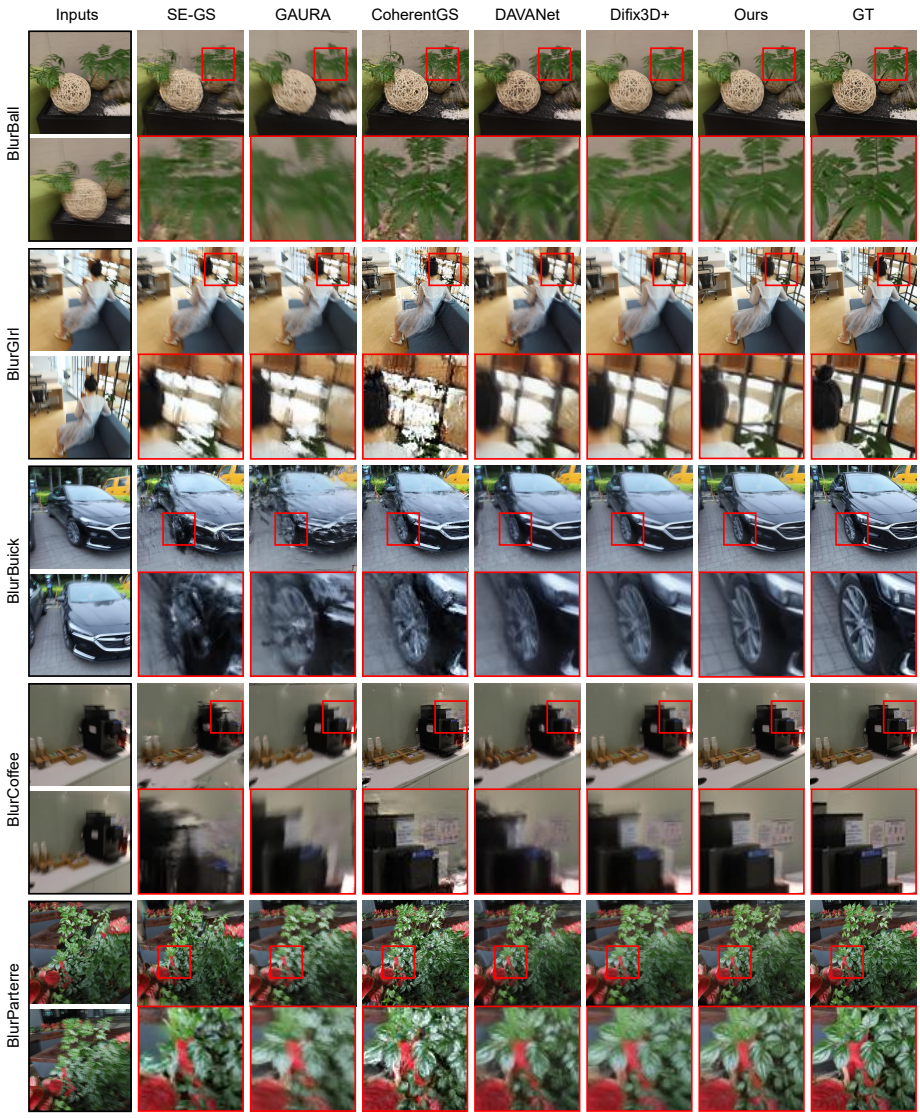

Fig. 4: Additional qualitative results of novel view synthesis on real-world scenes of the Deblur-NeRF dataset. The Inputs column shows the two blurry views; for each method, the lower row enlarges the red-boxed region.

References

1. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19457–19467 (2024)
2. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: European conference on computer vision. pp. 17–33. Springer (2022)
3. Choi, H., Yang, H., Han, J., Cho, S.: Exploiting deblurring networks for radiance fields. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6012–6021 (2025)
4. Gupta, V., Girish, R.S.V., Mukund Varma, T., Tewari, A., Mitra, K.: Gaura: Generalizable approach for unified restoration and rendering of arbitrary views. In: European Conference on Computer Vision. pp. 249–266. Springer (2024)
5. 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)
6. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (July 2017)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
8. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
9. TorchVision maintainers and contributors: TorchVision: PyTorch's computer vision library. `https://github.com/pytorch/vision` (2016)
10. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: International Conference on Computer Vision Workshops (ICCVW) (2021)
11. Wu, J.Z., Zhang, Y., Turki, H., Ren, X., Gao, J., Shou, M.Z., Fidler, S., Gojcic, Z., Ling, H.: Difix3d+: Improving 3d reconstructions with single-step diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26024–26035 (2025)
12. Xu, Z., Feng, C., Li, Y., Zhao, J., Yang, J., Yu, W., Yuan, L., Tian, Y.: Breaking the vicious cycle: Coherent 3d gaussian splatting from sparse and motion-blurred views. arXiv preprint arXiv:2512.10369 (2025)
13. Ye, B., Liu, S., Xu, H., Li, X., Pollefeys, M., Yang, M.H., Peng, S.: No pose, no problem: Surprisingly simple 3D gaussian splats from sparse unposed images. In: International Conference on Learning Representations (2025)
14. Zhao, C., Wang, X., Zhang, T., Javed, S., Salzmann, M.: Self-ensembling gaussian splatting for few-shot novel view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4940–4950 (2025)
15. Zhou, S., Zhang, J., Zuo, W., Xie, H., Pan, J., Ren, J.S.: Davanet: Stereo deblurring with view aggregation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10996–11005 (2019)