GS-RealBlur: A Flexible Data Acquisition Framework for Real-World Image Deblurring

Mingyang Chen, Zhilu Zhang, Honglei Xu, Renlong Wu, Xiaohe Wu, Wangmeng Zuo

Harbin Institute of Technology, Harbin, China

youngmchan269@gmail.com, cszlzhang@outlook.com, {cshongleixu, hirenlongwu, csxhwu, cswmzuo}@gmail.com

Project: https://youngmchan.github.io/GS-RealBlur/

Abstract

High-quality, large-scale paired data is essential for training learning-based image deblurring models. However, synthetic blurry images generally lack realism, while real-world captured images require complex and inflexible camera systems. In this work, we propose GS-RealBlur, a data acquisition framework for real-world image deblurring, achieving both blur realism and acquisition flexibility. Specifically, we use a handheld camera to capture blurry images, and deploy a gimbal to densely capture sharp images of the same scene. We reconstruct the 3D representation of sharp images and calibrate the camera pose of each blurry frame within this 3D. The image rendered from this 3D according to the pose serves as the sharp counterpart. To better align the rendered image with the blurry image, we introduce a Blur-aware Pose Refinement (BPR) module that refines the pose using appearance consistency and centroid alignment constraints. Leveraging GS-RealBlur, we construct a high-quality and diverse dataset. Extensive experiments demonstrate that a deblurring model trained on our dataset achieves superior generalization performance across various real-world deblurring benchmarks, consistently outperforming models trained on existing synthetic and real-world datasets. The code and dataset will be made publicly available.

Introduction

As a fundamental challenge in image restoration (Su, Xu, and Yin 2022; Zhai et al. 2023; Chen et al. 2021a; Zhang et al. 2021; Lin et al. 2026), image motion deblurring (Chen et al. 2022; Zamir et al. 2022; Wang et al. 2022; Liang et al. 2021; Tsai et al. 2022; Tao et al. 2018; Park et al. 2020; Gao et al. 2019; Zhou et al. 2019a) aims to reconstruct a latent sharp image from a blurry one degraded by motion from the camera and dynamic objects. To address this, learning-based methods (Nah, Hyun Kim, and Mu Lee 2017; Nah et al. 2019; Su et al. 2017; Zhou et al. 2019b; Shen et al. 2019; Deng et al. 2021; Li et al. 2021; Brooks and Barron 2019; Zhang et al. 2020) leverage paired datasets comprising blurry and sharp images, enabling supervised training of deblurring models. As a result, the performance and generalization capability of these models are directly contingent upon the quality and scale of the training data. Datasets of higher quality and larger scale are conducive to the development of more robust and effective deblurring models.

However, obtaining a large number of high-quality blurry-sharp image pairs is not trivial. A straightforward strategy is to aggregate multiple consecutive frames to synthesize blurry images. Early works (Nah, Hyun Kim, and Mu Lee 2017; Nah et al. 2019; Su et al. 2017; Zhou et al. 2019b; Shen et al. 2019; Deng et al. 2021; Li et al. 2021) employ high-speed cameras to capture sharp videos for blur synthesis. Recently, GS-Blur (Lee, Park, and Lee 2024) (see Figure 1(c)) reconstructs multi-view images into a 3D representation using 3D Gaussian Splatting (Kerbl et al. 2023), and then renders much denser images along randomly generated motion trajectories, averaging them to produce more realistic blurry images. Nevertheless, the blur synthesized by such methods still exhibits a distribution gap compared to real-world blur. This discrepancy arises because the discrete aggregation process fundamentally differs from the continuous photoelectric integration in a physical camera. While it is possible to approximate, completely closing this gap remains a significant challenge.

An alternative approach involves capturing real-world blurry-sharp image pairs. The primary challenge in this paradigm is ensuring the alignment of data pairs in terms of color and spatial position. RBVD (Zhu et al. 2022) addresses this by controlling the position and movement of a camera mounted on a robotic arm to capture aligned short- and long-exposure pairs, as illustrated in Figure 1(a). Other methods, such as those used to create the RealBlur (Rim et al. 2020), BSD (Zhong et al. 2020) and RSBlur (Rim et al. 2022) datasets, employ beam splitters. It can split the beam into two cameras, thereby enabling synchronous capture of short- and long-exposure pairs, as shown in Figure 1(b). However, these hardware setups are characterized by complex designs and lack portability. The inherent inflexibility of such systems restricts the scale and diversity of the datasets they produced.

It is evident that existing data acquisition methods for deblurring are constrained by a trade-off between blur realism and acquisition flexibility. In this work, we aim to design a framework that simultaneously achieves both objectives. This framework is guided by several key principles: (1) the blurry images must be captured in real-world scenarios, (2) the capture device should be portable and as simple as possible, (3) the image pairs should be aligned while the sharp images should be high-quality.

To this end, we propose GS-RealBlur, a flexible data acqui-

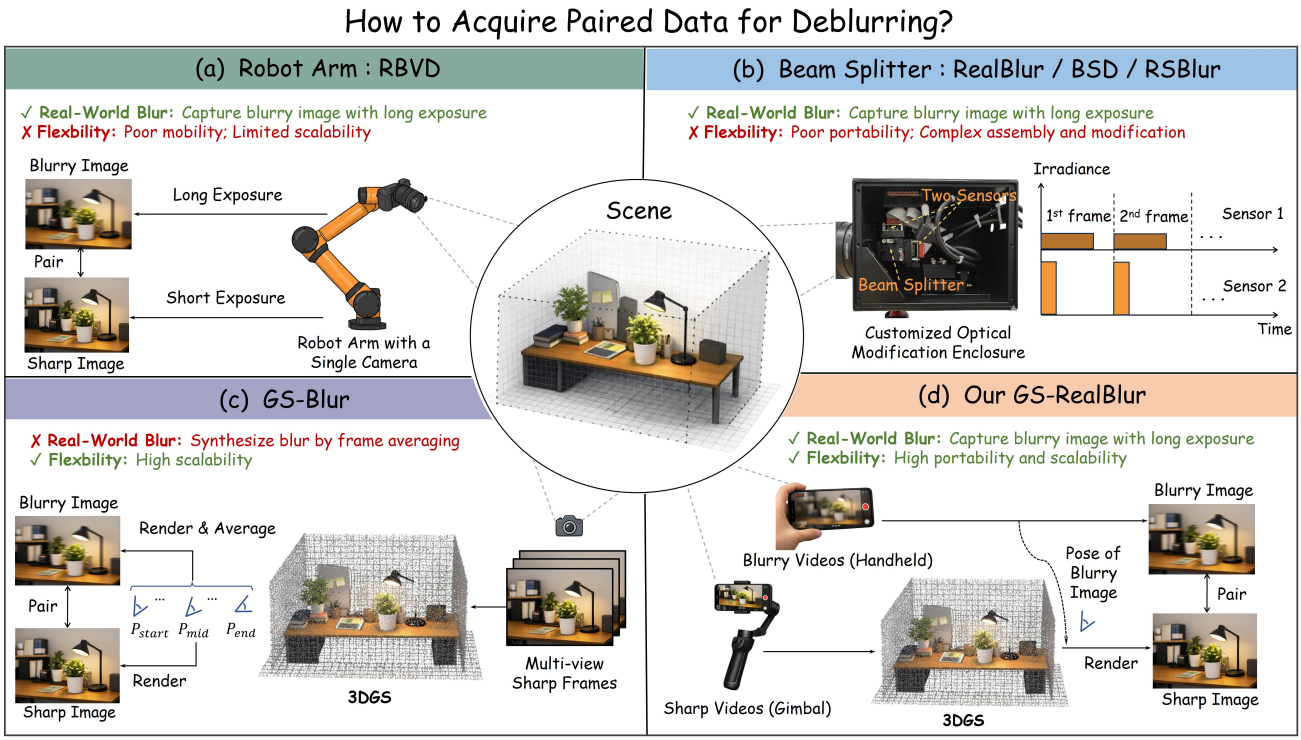


Figure 1: **Comparison of paired data acquisition methods for deblurring.** Existing methods either capture real-world blur with restricted flexibility (*e.g.*, (a) Robot arm (Zhu et al. 2022) and (b) Beam-splitter (Rim et al. 2020, 2022; Zhong et al. 2020)) or synthesize unrealistic blur (*e.g.*, (c) GS-Blur (Lee, Park, and Lee 2024)). In contrast, our GS-RealBlur (d) achieves both blur realism and acquisition flexibility by pairing real-world blurry images from consumer-grade devices with sharp correspondences rendered from 3DGS (Kerbl et al. 2023).

sition framework for real-world image deblurring, as shown in Figure 1(d). Specifically, we capture videos of a scene using consumer-grade devices (*e.g.*, smartphones) in both handheld and gimbal-mounted configurations. Frames captured with the handheld setup naturally exhibit motion blur, while those captured with the gimbal remain sharp. We then use the sharp frames to reconstruct a 3D representation of the scene and calibrate the camera pose of blurry frames within this 3D. This enables the rendering of a sharp image that is spatially aligned with each blurry frame. A critical challenge in this pipeline is that directly estimating the poses of blurry images using structure-from-motion tools like GLOMAP (Pan et al. 2024) often yields inaccurate results due to the detrimental effects of blur degradation. To address this issue, we introduce a Blur-aware Pose Refinement (BPR) module. BPR optimizes the pose by an appearance consistency loss and a centroid alignment regularizer, enabling the rendering to be better aligned with the blurry counterpart. Specifically, the centroid alignment regularizer serves as a regularizer to mitigate off-center translation misalignment between a blurry and a rendered sharp image. This is achieved by enforcing that the centroid of the estimated blur kernel coincides with its geometric center, ensuring that the rendered sharp frame is modeled as corresponding to the temporal midpoint of the blur formation process.

Finally, we use the data acquisition framework to construct a single-image deblurring dataset consisting of diverse indoor and outdoor scenes, under both daytime and nighttime conditions, with 13,209 blurry-sharp pairs in total. Extensive experiments demonstrate that a deblurring model trained on our dataset exhibits superior generalization performance on real-world scenarios compared to models trained on existing datasets, validating the effectiveness of GS-RealBlur.

The contributions are summarized as follows:

- We propose GS-RealBlur, a data acquisition framework for real-world image deblurring, which achieves both blur realism and acquisition flexibility.
- We propose Blur-aware Pose Refinement (BPR) to optimize the pose of blurry images by appearance consistency and centroid alignment regularizer.
- We construct a single-image deblurring dataset using GS-RealBlur. Extensive experiments demonstrate that the deblurring model trained on this dataset performs better generalization than ones trained on existing datasets in real-world scenarios.

Related Work

Image Deblurring Methods

Image deblurring is a long-standing challenge in computer vision. Traditional methods (Richardson 1972; Lucy 1974; Krishnan, Tay, and Fergus 2011; Krishnan and Fergus 2009)

typically formulate it as a MAP-based optimization problem (Fergus et al. 2006) under the assumption of a uniform blur kernel. However, they often struggle in real-world scenarios where blur is non-uniform and kernels are difficult to estimate. With the development of deep learning, methods are shifted toward end-to-end restoration models. Early CNN-based approaches (Nah, Hyun Kim, and Mu Lee 2017; Gao et al. 2019; Tao et al. 2018) adopted multi-scale architectures for deblurring. Subsequent studies (Chen et al. 2021b; Zamir et al. 2021) explored multi-stage restoration frameworks to enhance feature interaction. More recently, Transformer-based (Zamir et al. 2022; Wang et al. 2022; Vaswani et al. 2017; Liang et al. 2021; Tsai et al. 2022) architectures, further introduced attention mechanisms to model long-range dependencies. Despite these increasingly complex designs, recent studies (Chen et al. 2022; Cho et al. 2021) suggest that carefully designed simple baselines, can still achieve competitive performance by optimizing basic block designs. However, these models still depend on large-scale, high-quality blurry–sharp pairs, which have motivated extensive efforts in data acquisition.

Deblurring Datasets

Synthetic Deblurring Datasets. Synthetic blur datasets are widely used for training deblurring models. Mainstream methods synthesize blur images by averaging high-frame-rate video frames. GoPro (Nah, Hyun Kim, and Mu Lee 2017) pioneered this strategy using high-speed sequences captured by a GoPro camera. Subsequent works extended this paradigm in different settings. REDS (Nah et al. 2019) extends it to 120 FPS dynamic scenes, while DVD (Su et al. 2017) adopts a similar pipeline with handheld devices. Stereo Blur (Zhou et al. 2019b) applies it to binocular videos to produce stereo pairs. Except for GoPro, most datasets (Nah et al. 2019; Su et al. 2017; Zhou et al. 2019b; Shen et al. 2019; Deng et al. 2021) interpolate frames before averaging to accumulation to approximate better exposure blur. Although scalable, these datasets rely on high-speed cameras that require sufficient illumination(capture with short exposure) (Zhang et al. 2025), limiting coverage of low-light or night-time scenes. To avoid this dependency, GS-Blur (Lee, Park, and Lee 2024) reconstructs scenes using 3DGS (Kerbl et al. 2023) and synthesizes blur by integrating multi-view renderings along simulated camera trajectories. However, these methods are still constrained by the simplified mechanism, resulting in a distribution gap from real camera imaging processes.

In addition, several works move beyond the multi-frame averaging paradigm. Brooks et al. (Brooks and Barron 2019) synthesize blur by predicting spatially varying linear blur kernels from two consecutive sharp frames, but the linear assumption is insufficient. Zhang et al. (Zhang et al. 2020) employ GAN (Goodfellow et al. 2020) learned from real blurry images to generate blur images from a single sharp input, but the absence of explicit physical imaging modeling limits blur realism.

Real-world Deblurring Datasets. To improve performance in real scenarios, several works (Rim et al. 2020, 2022; Zhong et al. 2020)collect real blurry–sharp pairs using dual-sensor systems with beam splitters, which split incoming light into two paths and capture images with different exposure times. Representative datasets include BSD (Zhong et al. 2020), RealBlur (Rim et al. 2020), and RSBlur (Rim et al. 2022). Table 1 shows the information about existing deblurring datasets. Although these methods better reflect real blur formation, they require complex hardware and precise calibration, leading to poor flexibility and limited scalability. Differences between sensors may also cause photometric inconsistencies. RBVD (Zhu et al. 2022) uses a robotic arm to precisely control camera motion, enabling repeatable trajectories and accurate alignment of training pairs. the system has a large physical footprint, requires complex setup, and offers limited mobility, resulting in restricted scene coverage and poor scalability.

Table 1: **Comparison of Deblurring Datasets**

Dataset	Blur Type	#Pairs	Exp. Time (ms)	Resolution
RealBlur	Real	4,738	500	680×772
RBVD	Real	2,164	16–33	1443×960
BSD	Real	33,000	1–24	1280×720
RSBlur	Real + Synthetic	13,358	100	1920×1200
GS-Blur	Synthetic	156,209	–	–
GS-RealBlur	Real	13,209	1–100	3840×2160

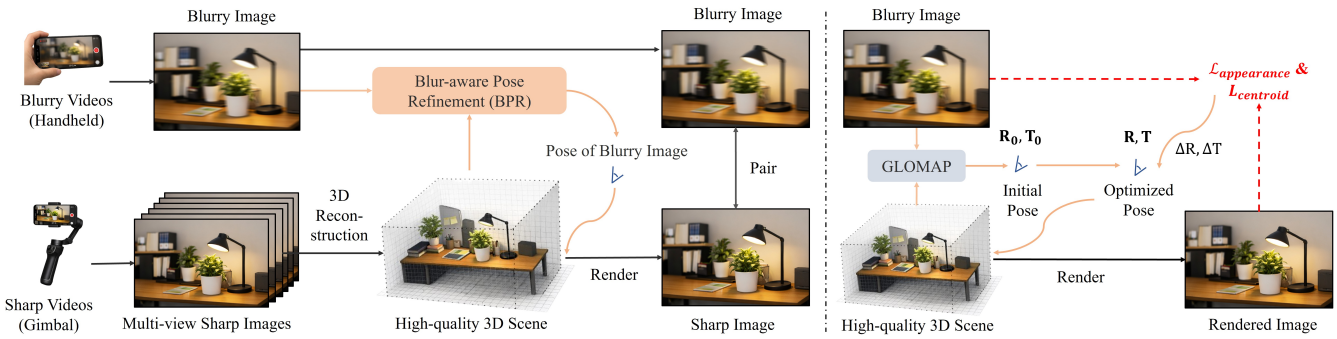
Some other works (Su et al. 2017; Zhang et al. 2020; Köhler et al. 2012; Xu et al. 2025) collect blurry images in real-world scenarios, but the lack of reliable aligned sharp images limits their use for supervised training. In general, existing real-world datasets remain limitation due to hardware constraints and capture difficulty.

Method

Motivation

Existing data acquisition methods either synthesize blurry images or use sophisticated camera systems to capture real-world blurry images. They are constrained by a trade-off between blur realism and acquisition flexibility.

Blur Realism. During exposure, the camera sensor integrates the incoming light over time to form an image. If the camera moves during exposure, motion blur occurs. To simulate this process for synthesizing blurred images, synthetic methods (Nah, Hyun Kim, and Mu Lee 2017; Nah et al. 2019; Su et al. 2017; Zhou et al. 2019b; Shen et al. 2019; Deng et al. 2021; Li et al. 2021) use high-speed cameras to capture sharp videos and aggregate consecutive sharp frames, but the discrete temporal sampling still introduces a gap from real blur. GS-Blur (Lee, Park, and Lee 2024) instead reconstructs scenes with 3DGS (Kerbl et al. 2023) and averages much denser renderings images along randomly generated motion trajectories, producing more realistic blurry images. Nevertheless, such synthetic blur still exhibits a distribution gap to real-world blur in three main aspects (Rim et al. 2022). First, discrete frame averaging cannot fully reproduce the continuous photoelectric integration of physical cameras. Second, synthetic blur is usually generated in sRGB or linear RGB



(a) Overview of Data Acquisition Pipeline (b) Blur-aware Pose Refinement

Figure 2: **Overview of GS-RealBlur**. We first capture real-world blurry images with handheld consumer-grade devices. Then we densely capture multi-view sharp images with a gimbal for high-quality 3D scene reconstruction. Given the camera pose of a blurry image optimized by Blur-aware Pose Refinement (BPR) module, the rendered image from this 3D is taken as the sharp counterpart.

space, while real blur forms in the RAW sensor stage and is further affected by the ISP and nonlinear camera response function (CRF). Third, synthetic methods struggle to model realistic sensor noise and saturated pixel caused by the limited sensor dynamic range, both of which commonly appear in real scenes.

Acquisition Flexibility. The primary challenge in capturing real-world blurry–sharp pairs is ensuring spatial and photometric alignment. Existing methods handle it through carefully designed hardware systems. RBVD (Zhu et al. 2022) uses a robotic arm to precisely control camera position and motion, capturing aligned short- and long-exposure pairs. But its large robotic platform and limited mobility restrict scene diversity and hinder the scalability of data collection. Other methods (Rim et al. 2020; Zhong et al. 2020; Rim et al. 2022) employ beam splitters to synchronously capture short- and long-exposure pairs with two sensors. But these systems require optical modifications, precise mechanical assembly, and careful calibration to ensure accurate alignment, making them bulky and less portable. Moreover, differences in sensor response require additional post-processing to align the brightness and color of the captured pairs.

In this work, we aim to design a framework that achieves both blur realism and acquisition flexibility. Specifically, the framework should (1) capture real-world blurry images, (2) the capture device should be portable, (3) the data pairs should be aligned while the sharp images should be high-quality. To this end, we propose GS-RealBlur, which achieves the above expectations by building a bridge between captured 2D videos and real 3D scenes, as shown in Figure 2. The acquisition pipeline is described below.

Data Acquisition Pipeline

Capturing Data. We capture videos using an iPhone 16 Pro Max in manual mode. We record each scene in both handheld and gimbal-mounted configurations: the handheld setup naturally introduces motion blur, whereas the gimbal setup keeps the frames sharp. For sharp videos, we move the camera slowly and smoothly to maintain substantial overlap between adjacent frames. All videos are captured at 4K resolution. These dense, high-resolution sharp frames facilitate high-quality 3D reconstruction. Besides, the two setups share the same imaging parameters (*e.g.*, ISO, aperture, shutter speed, and white balance) to maintain the photometric consistency across video frames and blurry-sharp images.

Aligned Sharp Image Rendering. Although the captured sharp frames may be not aligned with the blurry frames, they are sufficiently dense to cover nearly the entire scene content. Therefore, these sharp frames contain the ground truth information required to recover the blurry frames, and we extract it by reconstructing a 3D representation. After reconstructing the 3D representation, we can render a sharp image aligned with the blurry image. Specifically, we first use GLOMAP (Pan et al. 2024) to calibrate the camera pose of each blurry image within the 3D. Then, we refine the pose by the Blur-aware Pose Refinement (BPR) module to obtain more accurate one. With the refined pose, we render a sharp image from the reconstructed 3D representation and use it as supervision for training deblurring models.

Artifact Filtering. Despite our careful 3D reconstruction, novel-view renderings may still contain artifacts. To ensure reliable supervision, we adopt a three-stage filtering strategy. First, at the scene level, we compute the average PSNR between the ground-truth images and the corresponding rendered sharp images at held-out views. Then, we discard scenes whose average PSNR is below 36,dB, following GS-Blur (Lee, Park, and Lee 2024). Second, at the image level, we assess the quality of each sharp image rendered from the retained scenes using MUSIQ (Ke et al. 2021). Images with a MUSIQ score below 63 are excluded. Third, we manually inspect the remaining images to remove samples containing noticeable artifacts that are not detected by automatic filtering, further guaranteeing the overall dataset quality.

Blur-aware Pose Refinement (BPR)

Existing structure-from-motion methods (Schonberger and Frahm 2016; Pan et al. 2024) estimate camera poses by

matching local keypoints, relying on high-frequency textures. However, for blurry images, blur degradation smooths image textures, reducing matching reliability and leading to inaccurate pose estimation. Therefore, we propose a Blur-aware Pose Refinement (BPR) module to optimize the poses estimated by GLOMAP (Pan et al. 2024).

Overview of BPR. First, for each blurry image $\mathbf{B}$, we estimate an initial pose $\mathbf{P}_0 = (\mathbf{R}_0, \mathbf{T}_0)$ with GLOMAP (Pan et al. 2024). The optimized pose $\mathbf{P} = (\mathbf{R}, \mathbf{T})$ can be formulated as:

$$\mathbf{R} = \Delta\mathbf{R} \cdot \mathbf{R}_0, \mathbf{T} = \mathbf{T}_0 + \Delta\mathbf{T}, \tag{1}$$

where $\Delta\mathbf{R}$ and $\Delta\mathbf{T}$ denote the learnable rotation increment and translation increment, and $\cdot$ denotes the matrix multiplication. Subsequently, given the optimized pose $\mathbf{P}$, we render a sharp image $\mathbf{S}$ from the reconstructed 3D representation with frozen parameters. Finally, the blurry image $\mathbf{B}$ and the sharp image $\mathbf{S}$ are used to compute the appearance consistency loss and the centroid-alignment regularizer to iteratively update $\Delta\mathbf{R}$ and $\Delta\mathbf{T}$.

Appearance Consistency Loss. Although the rendered sharp image exhibits richer high-frequency details than the blurry one, they should have highly consistent low-frequency structures. Therefore, the appearance consistency loss is formulated as:

$$\mathcal{L}_{appearance} = \|\mathbf{S}_{\downarrow} - \mathbf{B}_{\downarrow}\|_1, \tag{2}$$

where $\downarrow$ denotes a $\times 4$ downsampling operator, $\mathbf{S}$ denotes the rendered sharp image, and $\mathbf{B}$ denotes the blurry image.

Centroid Alignment Regularization. In our experiments, we find that although the appearance consistency loss provides effective content-aware supervision, it remains insufficient for translation alignment. We therefore introduce a centroid alignment regularizer to provide an additional geometric constraint. Specifically, we first estimate the blur kernel between the rendered sharp image and the blurry observation by solving (Cho and Lee 2017; Xu, Zheng, and Jia 2013; Cho and Lee 2009),:

$$\hat{\mathbf{K}} = \arg\min_{\mathbf{K}} \|\mathbf{K} * \nabla\mathbf{S} - \nabla\mathbf{B}\|_2^2 + \lambda\|\nabla\mathbf{K}\|_2^2, \tag{3}$$

where $\mathbf{K}$ denotes the blur kernel, $*$ denotes convolution operation, ∇ denotes a gradient operator, and $\lambda = 10^3$ denotes a regularization weight. Then, we calculate the centroid c and geometric center c_0 of the blur kernel $\mathbf{K}$:

$$c = \left(\frac{\sum_{i,j} i\,\hat{\mathbf{K}}(i,j)}{\sum_{i,j} \hat{\mathbf{K}}(i,j)}, \frac{\sum_{i,j} j\,\hat{\mathbf{K}}(i,j)}{\sum_{i,j} \hat{\mathbf{K}}(i,j)} \right), \quad c_0 = \left(\frac{H}{2}, \frac{W}{2} \right), \tag{4}$$

where (i, j) denotes the spatial coordinates of the kernel elements. H and W represent the height and width of the blur kernel, respectively. Finally, centroid alignment loss is defined as:

$$\mathcal{L}_{centroid} = \|c - c_0\|_2. \tag{5}$$

Final Learning Objective. We optimize the camera pose by jointly minimizing the appearance consistency loss and the centroid alignment regularization:

$$\mathcal{L} = (1 - \alpha)\mathcal{L}_{appearance} + \alpha\mathcal{L}_{\text{centroid}}. \tag{6}$$

where α is a weighting coefficient set to 0.1.

Data Augmentation for Object Motion Blur

The primary limitation of using 3DGS is its restriction to rendering static scenes, meaning our method mainly addresses motion blur caused by camera movement. However, in real-world blurry images, object motion blur and camera motion blur are independent, specifically manifested as inconsistent local blurring. To alleviate the scarcity of object motion blur in our dataset, we suggest CutMix-Dynamic, a training-time data augmentation strategy inspired by CutMix (Yun et al. 2019). Unlike the original CutMix, which randomly samples image patches, CutMix-Dynamic focuses on dynamic objects to better reflect real-world physical phenomena. Specifically, we first use YOLOv8 (Varghese and Sambath 2024) to extract spatially aligned blurry-sharp patch pairs containing dynamic objects. These patches preserve characteristic dynamic textures and real motion blur patterns, and are collected into a dynamic object bank for later sampling. Then, during training, we sample a pair from the bank and paste the blurry and sharp patches onto the blurry input and its sharp supervision, to simulate the inconsistent local blurring. Consequently, training with CutMix-Dynamic improves the generalization of the deblurring model to real-world blurry images. More implementation details and discussion are provided in the *Suppl*.

Experiments

Implementation Details

Datasets. To demonstrate the effectiveness of our proposed data acquisition framework, we train a representative deblurring model on different real-world datasets in a supervised manner respectively and evaluate its performance across multiple real-world benchmarks. Specifically, six datasets with ground truth references (*i.e.*, RealBlur (Rim et al. 2020), RSBlur (Rim et al. 2022), RBVD (Zhu et al. 2022), BSD (Zhong et al. 2020), GS-Blur (Lee, Park, and Lee 2024), and our dataset constructed in GS-RealBlur) are utilized for cross-dataset evaluation. Furthermore, we assess the out-of-distribution (OOD) generalization ability on two datasets without ground truth (*i.e.*, RWBI (Zhang et al. 2020) and DVD-Test (Su et al. 2017)).

Training Details. We adopt NAFNet (Chen et al. 2022) as the deblurring network, which is trained with the AdamW optimizer (Loshchilov and Hutter 2017a) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for 200k iterations. Cosine annealing strategy (Loshchilov and Hutter 2017b) is employed to steadily decrease the learning rate from 10^{-3} to 10^{-6}. We randomly crop patches and augment them with flips and rotations. The patch size is set to 256×256 and the batch size is set to 8. All experiments are conducted with PyTorch (Paszke et al. 2019) on an Nvidia GeForce RTX A6000 GPU.

Evaluation Configurations. For cross-dataset evaluation (*i.e.*, RealBlur (Rim et al. 2020), RBVD (Zhu et al. 2022), RSBlur (Rim et al. 2022) and BSD (Zhong et al. 2020)), we use PSNR, SSIM (Wang et al. 2004), and LPIPS (Zhang et al. 2018) as the evaluation metrics. For the evaluation of OOD generalization ability, we employ four recent no-reference metrics (*i.e.*, MUSIQ (Ke et al. 2021), MANIQA (Yang et al.

Table 2: **Quantitative Comparison of Cross-Validation on Real-world Datasets.** Models are separately trained on different training sets and evaluate across various real-world benchmarks. Best results are in red and second-best are in blue. Except for cases where training and testing sets match, the model trained on our GS-RealBlur achieves the best performance across all benchmarks.

Train set \ Test set		RealBlur			RBVD			RSBlur			BSD			Average		
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
RealBlur	NAFNet	28.89	0.907	0.151	26.00	0.892	0.249	30.61	0.824	0.342	30.00	0.914	0.125	28.88	0.884	0.221
	Restormer	29.27	0.878	0.257	26.08	0.821	0.313	29.85	0.778	0.429	30.31	0.903	0.211	28.88	0.845	0.303
	EVSSM	29.59	0.918	0.239	25.90	0.892	0.301	30.23	0.829	0.440	30.13	0.914	0.227	28.96	0.888	0.302
RBVD	NAFNet	27.16	0.863	0.228	26.51	0.907	0.231	29.59	0.793	0.388	29.36	0.907	0.136	28.16	0.867	0.246
	Restormer	26.93	0.814	0.333	26.66	0.835	0.289	29.08	0.747	0.417	29.17	0.892	0.220	27.96	0.822	0.315
	EVSSM	26.98	0.871	0.329	26.62	0.909	0.283	29.71	0.814	0.442	28.73	0.893	0.236	28.01	0.872	0.323
RSBlur	NAFNet	27.23	0.871	0.180	26.36	0.905	0.234	33.72	0.877	0.310	30.78	0.923	0.119	29.54	0.894	0.211
	Restormer	27.30	0.829	0.297	26.50	0.834	0.296	33.42	0.833	0.381	30.81	0.910	0.211	29.51	0.852	0.296
	EVSSM	27.44	0.876	0.287	26.42	0.904	0.300	34.02	0.875	0.354	31.47	0.934	0.204	29.84	0.897	0.286
BSD	NAFNet	26.88	0.864	0.217	26.36	0.902	0.248	30.93	0.832	0.377	33.87	0.952	0.078	29.53	0.888	0.230
	Restormer	27.01	0.823	0.316	26.44	0.825	0.318	31.19	0.792	0.421	33.68	0.943	0.167	29.58	0.846	0.306
	EVSSM	25.87	0.847	0.328	26.36	0.900	0.284	30.32	0.829	0.425	36.10	0.964	0.138	29.66	0.885	0.294
GS-Blur	NAFNet	27.33	0.879	0.147	26.26	0.904	0.201	32.87	0.860	0.317	31.37	0.934	0.109	29.46	0.895	0.192
	Restormer	27.32	0.841	0.276	26.52	0.833	0.293	32.42	0.817	0.403	31.26	0.920	0.205	29.38	0.853	0.294
	EVSSM	27.37	0.880	0.250	26.44	0.905	0.290	33.28	0.864	0.383	31.80	0.938	0.196	29.72	0.897	0.280
GS-RealBlur	NAFNet	27.67	0.886	0.140	26.71	0.910	0.186	33.15	0.863	0.311	31.92	0.939	0.093	29.86	0.900	0.183
	Restormer	27.65	0.852	0.271	26.85	0.838	0.280	32.59	0.820	0.400	31.63	0.925	0.203	29.68	0.859	0.288
	EVSSM	27.82	0.887	0.243	26.96	0.911	0.277	33.48	0.866	0.380	32.17	0.942	0.193	30.11	0.902	0.273

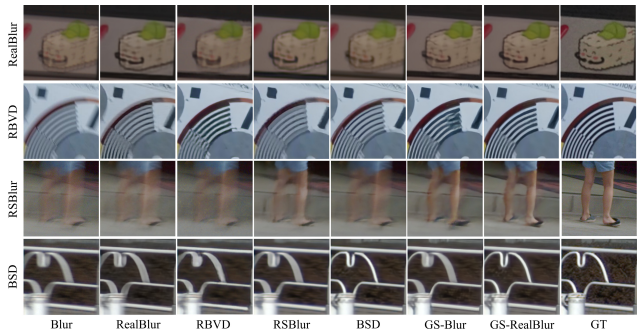


Figure 3: **Visual Comparison of Cross-Validation on Real-world Datasets**

2022) and CLIP-IQA (Wang, Chan, and Loy 2023), as there is no ground truth in the benchmarks (*i.e.*, RWBI (Zhang et al. 2020) and DVD-Test (Su et al. 2017)).

Generalization on Real-World Scenarios

Cross-Validation Results on Real-world Benchmarks. Table 2 reports the quantitative comparison of different training sets evaluated across various real-world benchmarks (*i.e.*, RealBlur (Rim et al. 2020), RBVD (Zhu et al. 2022), RSBlur (Rim et al. 2022) and BSD (Zhong et al. 2020)). The results indicate a severe domain gap among existing datasets, as models trained on themselves suffer significant performance drops when tested on others. In contrast, our GS-RealBlur remarkably alleviates this issue. Even without training on the target domain, the model trained on GS-RealBlur delivers highly competitive cross-domain performance, securing the highest average scores among all training sets. This strong performance highlights the superior quality and diversity of our dataset constructed via the proposed framework. Visual comparisons in Figure 3 further support our claims. In cross-domain scenarios (where training and testing sets differ), the model trained on GS-RealBlur consistently produces the most visually pleasing results. Remarkably, even in in-domain settings, it exhibits visual quality equivalent to or even higher than the models specifically optimized on those target datasets.

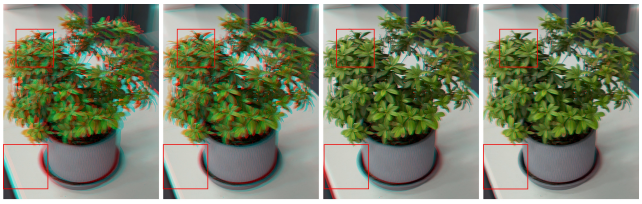
Table 3: **Quantitative Comparison of In-the-wild Data**

Train set \ Test set	RWBI			DVD-Test		
	MUSIQ↑	MANIQA↑	CLIP-IQA↑	MUSIQ↑	MANIQA↑	CLIP-IQA↑
RealBlur	58.552	0.266	0.341	45.040	0.231	0.290
RBVD	52.014	0.258	0.313	40.970	0.206	0.224
RSBlur	57.929	0.264	0.335	41.132	0.217	0.274
BSD	58.104	0.273	0.336	40.595	0.214	0.275
GS-Blur	61.330	0.295	0.367	45.371	0.235	0.284
GS-RealBlur	**61.610**	**0.300**	**0.372**	**46.604**	**0.242**	**0.294**

Generalization Results on In-the-wild Data. To further assess the out-of-distribution (OOD) generalization on in-the-wild data, we conduct evaluations on the RWBI (Zhang et al. 2020) and DVD-Test (Su et al. 2017) datasets in NAFNet (Chen et al. 2022). As reported in Table 3, the model trained on our GS-RealBlur demonstrates remarkable robustness, consistently achieving the highest scores across all metrics on both test sets and outperforming models trained on existing benchmarks by a clear margin.

Table 4: **Ablation on BPR and Artifact Filtering**

BPR	Artifact Filtering	RealBlur PSNR↑	RealBlur SSIM↑	RBVD PSNR↑	RBVD SSIM↑	RSBlur PSNR↑	RSBlur SSIM↑	BSD PSNR↑	BSD SSIM↑
✗	✗	26.36	0.848	26.06	0.892	30.89	0.821	29.92	0.917
✓	✗	27.13	0.863	26.33	0.897	32.41	0.853	30.86	0.928
✗	✓	27.38	0.880	26.46	0.899	32.61	0.854	31.42	0.930
✓	✓	**27.57**	**0.885**	**26.67**	**0.908**	**33.13**	**0.861**	**31.63**	**0.936**



(a) No Optimization (b) Just Centroid Loss (c) Just L1 Loss (d) The Full Loss

Figure 4: **Effect of BPR on Geometric Alignment.** Each visualization result is shown as a stereo-anaglyph image. 'Full loss' produces more aligned results around the plant leaves, table edges, and the tray.

Ablation Studies

We conduct comprehensive ablation studies to validate the effectiveness of each component in the proposed framework. Unless otherwise specified, all ablation experiments are trained on 5,000 samples using NAFNet (Chen et al. 2022), following the same training setting described in Training Details. Due to space limitations, the complete experimental results are provided in the supplementary material, while only selected metrics are reported here.

Effect of BPR and Artifact Filtering. The effect of Blur-aware Pose Refinement (BPR) and artifact filtering are summarized in Table 4. Without both BPR and artifact filtering, the constructed pairs suffer from geometric misalignment and rendering artifacts, leading to poor deblurring performance. BPR improves supervision alignment leading to performance improvement, indicating the importance of accurate pose estimation for supervision construction. Artifact filtering removes samples with severe artifacts, also enabling performance gain. Their combination performs best.

Effect of Loss Terms in BPR. We analyze the contribution of each loss term in BPR using pose errors and downstream deblurring performance. Since directly measuring the alignment of real blurry–sharp pairs is challenging, we construct a synthetic evaluation set using GS-Blur and regard the calibrated poses of the corresponding sharp images as ground truth. We report Translation Error (TE) and Rotation Error (RE) to evaluate alignment quality. As shown in Table 5, $\mathcal{L}_{\text{centroid}}$ primarily reduces translation error but offers limited improvement in deblurring when used alone, while $\mathcal{L}_{\text{appearance}}$ effectively reduces the rotation error, resulting in substantially improved deblurring performance. Combining them achieves the lowest pose errors and the best deblurring performance, demonstrating their complementarity. Figure 4 further confirms their complementarity, where the full loss produces the most accurate alignment by jointly correcting spatial displacement and texture misalignment.

Effect of CutMix-Dynamic. We conduct an ablation study to evaluate the effectiveness of CutMix-Dynamic, as shown in Table 6. The baseline performs consistently on both static and dynamic benchmarks. Introducing CutMix (Yun et al. 2019) improves performance on the dynamic datasets but slightly degrades results on the static RealBlur dataset, likely due to physically implausible local replacements. In contrast, by inserting textures extracted from dynamic objects, our CutMix-Dynamic achieves the best results on both datasets. It more accurately models real motion blur while preserving static scene fidelity.

Table 5: **Ablation on Loss Terms in BPR.**

$\mathcal{L}_{appearance}$	$\mathcal{L}_{centroid}$	Pose Error TE↓ / RE↓	RealBlur PSNR↑	RealBlur SSIM↑	RBVD PSNR↑	RBVD SSIM↑	RSBlur PSNR↑	RSBlur SSIM↑	BSD PSNR↑	BSD SSIM↑
✗	✗	1.036 / 1.102	27.38	0.880	26.46	0.899	32.61	0.854	31.42	0.930
✓	✗	0.702 / 0.503	27.52	0.884	26.63	0.907	32.95	0.858	31.59	0.935
✗	✓	0.452 / 0.841	27.43	0.883	26.52	0.903	32.75	0.856	31.51	0.934
✓	✓	**0.445 / 0.479**	**27.57**	**0.885**	**26.67**	**0.908**	**33.13**	**0.861**	**31.63**	**0.936**

Table 6: **Ablation on CutMix-Dynamic**

Methods	RealBlur PSNR↑	RealBlur SSIM↑	RBVD PSNR↑	RBVD SSIM↑	RSBlur PSNR↑	RSBlur SSIM↑	BSD PSNR↑	BSD SSIM↑
Baseline	27.52	0.884	26.60	0.908	33.09	**0.860**	31.52	0.935
+ CutMix	27.50	0.884	26.62	0.908	33.10	**0.860**	31.59	**0.936**
+ CutMix-Dynamic	**27.58**	**0.885**	**26.67**	**0.909**	**33.12**	**0.860**	**31.64**	**0.936**

Table 7: **Ablation on Dataset Scale**

Data Proportions	RealBlur PSNR↑	RealBlur SSIM↑	RBVD PSNR↑	RBVD SSIM↑	RSBlur PSNR↑	RSBlur SSIM↑	BSD PSNR↑	BSD SSIM↑
25%	27.52	0.883	26.47	0.907	33.08	0.862	31.58	0.935
50%	27.58	0.885	26.57	0.908	33.10	0.862	31.69	0.937
100%	**27.67**	**0.886**	**26.71**	**0.910**	**33.15**	**0.863**	**31.92**	**0.939**

Effect of Dataset Scale. To investigate the effect of dataset scale, we conduct an ablation study by training the model with different proportions of the training dataset. Specifically, we randomly sample 25%, 50%, and 100% of the training data while keeping all other training settings unchanged. As shown in Table 7, a consistent performance improvement as the amount of training data increases. When trained with only 25% of the data, the model already achieves competitive results. Increasing the training data to 50% leads to further improvements, and the model achieves the best performance when trained with the full dataset. This trend highlights the benefit of scaling up GS-RealBlur.

Conclusion

Existing data acquisition methods face a trade-off between blur realism and acquisition flexibility. With this motivation, we propose GS-RealBlur, a data acquisition framework that simultaneously achieves both objects for real-world image deblurring. GS-RealBlur first captures blurry and sharp images of a scene using consumer-grade devices in handheld and gimbal-mounted configurations. Then, the sharp frames are used to reconstructed a 3D representation, from which sharp references are rendered for blurry images. To address inaccurate pose under blur, we further introduce a Blur-aware Pose Refinement (BPR) module to optimize the pose by an appearance consistency loss and a centroid alignment regularizer. Finally, we use the data acquisition framework to construct a single-image deblurring dataset. Extensive experiments demonstrate that a deblurring model trained on our dataset exhibits superior generalization performance on real-world scenarios compared to models trained on existing datasets, validating the effectiveness of GS-RealBlur.

GS-RealBlur: A Flexible Data Acquisition Framework for Real-World Image Deblurring (Supplementary Material)

3D Gaussian Splatting

3DGS models a scene with a set of three-dimensional Gaussian ellipsoids $\mathcal{K}$, where each Gaussian primitive $k \in \mathcal{K}$ is represented by a quadruple

$$\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \sigma_k, \mathbf{S}_k\}_{k\in\mathcal{K}} \tag{A}$$

where $\boldsymbol{\mu}_k \in \mathbb{R}^3$ denotes the spatial position, $\boldsymbol{\Sigma}_k \in \mathbb{R}^{3\times3}$ denotes the covariance matrix, σ_k denotes the opacity parameter, and $\mathbf{S}_k$ denotes spherical harmonic coefficients that model view-dependent color variation. The initial Gaussian parameters are typically estimated using Structure-from-Motion (SfM) (Schonberger and Frahm 2016; Pan et al. 2024) and are further refined in subsequent optimization.

During rendering, the three-dimensional Gaussian ellipsoids are projected onto the image plane. The color of each pixel p is then computed using point-based α-blending as follows:

$$\hat{C}(p) = \sum_{k\in\mathcal{K}} \alpha_k \mathbf{c}(\mathbf{v}_k; \mathbf{S}_k) \prod_{j=1}^{k-1} (1-\alpha_j), \tag{B}$$

α_k denotes the opacity of the k-th Gaussian primitive. More detailed formulations of the imaging model are provided in (Kerbl et al. 2023).

Compared with implicit volumetric rendering methods such as NeRF (Mildenhall et al. 2021), 3DGS employs explicit Gaussian ellipsoids with efficient rasterization, enabling fast and high-quality rendering. In this work, we leverage its geometric consistency to generate sharp images from the reconstructed scene, providing stable and precisely aligned supervision for real blurry images.

Discussion about Rendered Supervision

Unlike early works that use captured sharp images as supervision, our sharp references are rendered from reconstructed 3DGS representations. This design enables accurate spatial alignment but also introduces concerns regarding the domain gap and the quality of rendered supervision. We discuss the motivation, trade-off, and reliability of this design below.

Motivation of the Design

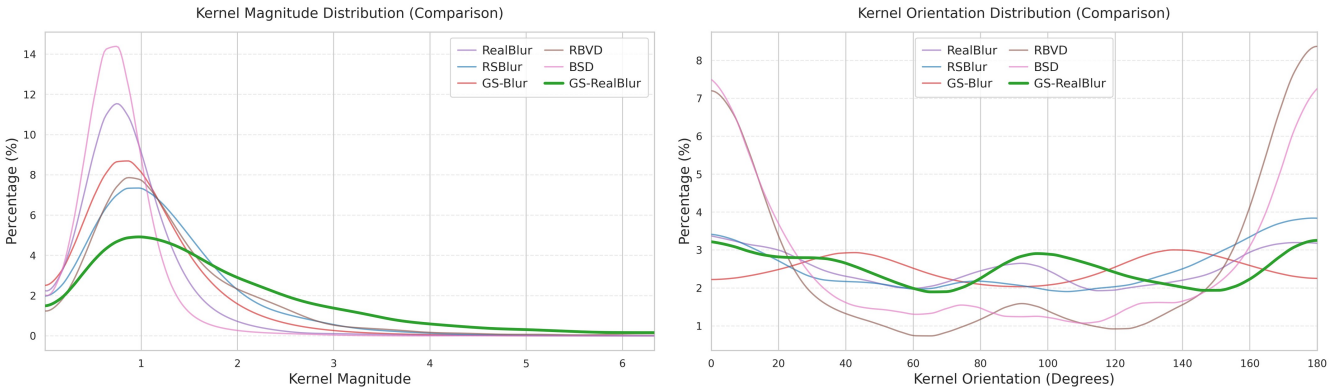
Obtaining accurately aligned blurry-sharp pairs in real-world scenes is challenging because the camera pose changes during exposure and may also vary between separate blurry and sharp captures. Consequently, directly capturing aligned blurry and sharp pairs is difficult. Post-processing methods like optical-flow alignment provide a possible solution, but they are often unreliable in heavily blurred regions and may introduce additional warping artifacts. In contrast, 3DGS provides a high-fidelity scene representation and allows sharp images to be rendered from any viewpoints that are adaptively aligned with the blurry observations. It therefore provides a practical way to preserve real captured blur while obtaining spatially aligned sharp supervision, which is difficult to achieve using conventional acquisition and alignment methods.

Trade-Off of the Design

This design relies on rendered sharp references. However, it preserves fully real captured blur and enables the construction of spatially aligned blurry-sharp pairs. It entails a clear trade-off between the authenticity of sharp supervision and the realism of input blur. We argue that realistic blurry inputs are more important for downstream deblurring than strictly real-captured sharp supervision. This argument is supported by two observations. First, as shown in Table 2 in the main paper, models trained on our dataset achieve the best generalization performance. Second, we further construct two training sets from the same scenes. One uses blurry images synthesized by GS-Blur, whereas the other uses real blurry images captured by our framework. In RealBlur (Rim et al. 2020) test set, the model trained with real blurry images achieves a PSNR/SSIM of 27.48/0.880, compared with 27.11/0.875 for the model trained with synthetic blurry images. These results indicate that the benefit of realistic blurry inputs outweighs the rendering gap.

Besides, to ensure the reliability of rendered supervision as much as possible, we employ a carefully designed framework. During data acquisition, we densely capture sharp-view videos to provide sufficient scene observations for 3DGS reconstruction. The blurry images are captured within the spatial range covered by sharp videos, reducing artifacts caused by large viewpoint deviations or insufficient observations. We further crop the images into patches and remove samples with evident artifacts by metric and strict manual inspection, ensuring that only reliable references are retained for training.

Overall, our design preserves the real-captured blur while providing spatially aligned high-quality supervision. Although the sharp supervisions are rendered, our experimental analyses show that the resulting trade-off is beneficial, supporting the reliability of rendering supervisions for deblurring.



(a) Blur Kernel Magnitude Distribution

(b) Blur Kernel Orientation Distribution

Figure A: **Blur Distribution of Deblurring Datasets**

Blur Distribution of GS-RealBlur Dataset.

To characterize the diversity of blur patterns in GS-RealBlur, we estimate the blur kernel of each blurry image using Equation (C) and analyze its magnitude and orientation.

$$\hat{\mathbf{K}} = \arg\min_{\mathbf{K}} \|\mathbf{K} * \nabla\mathbf{S} - \nabla\mathbf{B}\|_2^2 + \lambda\|\nabla\mathbf{K}\|_2^2, \quad \text{(C)}$$

where $\mathbf{K}$ donates the blur kernel, $*$ denotes convolution operation, ∇ denotes a gradient operator, and $\lambda = 10^3$ denotes a regularization weight.

As shown in Figure A, benefiting from our flexible blurry images collection, GS-RealBlur exhibits a wider range of blur magnitudes and a more uniform distribution of blur orientations.

Data Augmentation for Object Motion Blur

The primary limitation of using 3DGS is its restriction to rendering static scenes, meaning our method can only model motion blur caused by camera movement. However, in real-world blurry images, object motion blur and camera motion blur are independent, specifically manifested as inconsistent local blurring. To address the scarcity of dynamic blur in our datasets, we propose CutMix-Dynamic, a training-time data augmentation strategy inspired by CutMix (Yun et al. 2019). Unlike the original CutMit, which randomly extracts patches, CutMix-Dynamic specifically samples paired blurry and sharp patches containing dynamic objects. It therefore introduces realistic local motion blur while maintaining spatially aligned supervision. It could be done in 2 steps.

Dynamic Object Bank Construction As illustrated in Figure B(a), first, we define 9 categories of dynamic objects (Lin et al. 2014), including person, bicycle, car, motorcycle, bus, truck, cat, dog and sports ball. Then, we use a dynamic object detector (*i.e.*, YOLOv8 (Varghese and Sambath 2024)) to detect these objects and extract blurry-sharp patch pairs containing them from datasets. We collect these patch pairs to construct the Dynamic Objects Bank. These patches contain characteristic dynamic textures and real motion blur patterns, and we collect them into a dynamic object bank.

Cutmix-Dynamic As illustrated in Figure B(b), during training, we sample a patch pair from the bank, and paste the blurry patch on the input blurry images to simulate inconsistent local blurring. Simultaneously, the corresponding sharp patch is pasted on the ground truth sharp image to maintain consistent supervision. Consequently, training on the data enables the deblurring model to generalize effectively to real-world blurry images, even those with object motion blur.Specifically, we randomly sample a patch pair from the Dynamic Objects Bank and paste the blurry and sharp patches on the input blurry image and its sharp supervision, respectively. The augmented blurry images also exhibit locally inconsistent blur, similar to that observed in real-world images.

More Experiments

Additional Visual Comparison

Figure C presents additional visual comparisons. Figure C (a) presents comparisons on real-world datasets, and Figure C (b) shows visual results in in-the-wild data. In (b), rows 1 and 3 contain camera-motion blur, while rows 2 and 4 contain object-motion blur. The model trained on GS-RealBlur produces the clearest results.

Study of Artifact Filtering.

To investigate the effect of different filtering thresholds and determine an appropriate MUSIQ threshold, we manually select 500 samples from 100 scenes for evaluation. Half of the samples are manually identified as high-quality, while the other half contain noticeable artifacts. We then apply different MUSIQ thresholds to these samples and evaluate their ability to retain high-quality samples and reject samples with artifacts. Thresholds of 60, 63, and 65 retain 80%, 76%, and 63% of the high-quality patches, while rejecting 55%, 61%, and 64% of the patches with artifacts, respectively. We therefore set the MUSIQ threshold to 63, as it achieves a reasonable trade-off between retaining high-quality samples and reducing the burden of manual artifact inspection.

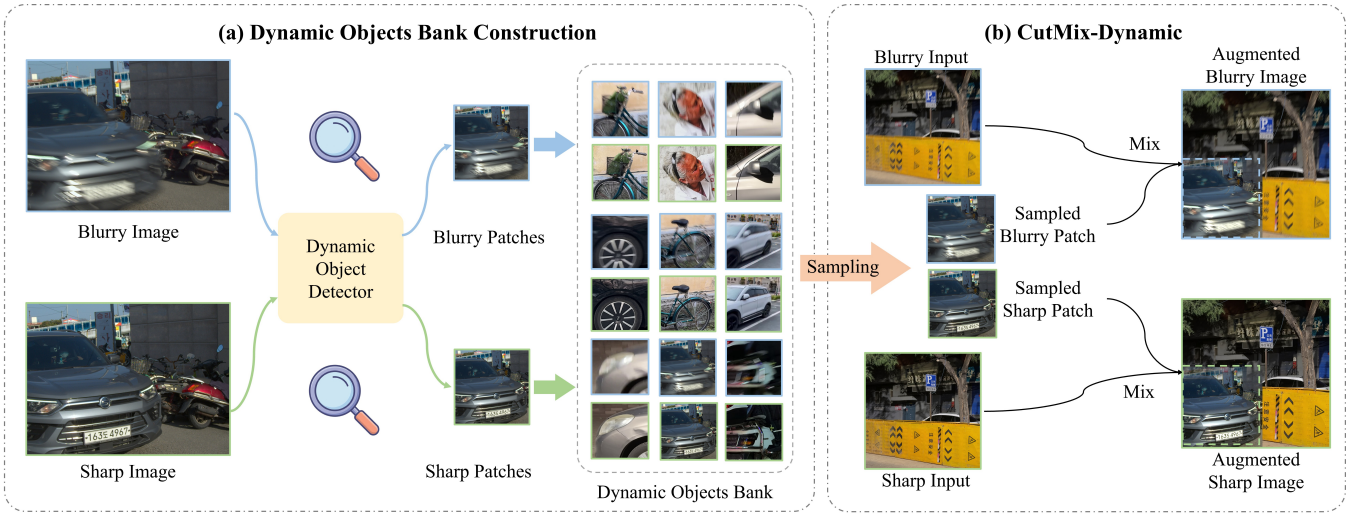


Figure B: **Overview of our CutMix-Dynamic**. We first construct a dynamic objects bank by extracting blurry-sharp patch pairs containing dynamic objects using a dynamic object detector. Subsequently, during the training process, CutMix-Dynamic randomly samples a patch pair from the Dynamic Objects Bank and pastes the blurry and sharp patches on the input blurry image and its sharp supervision, respectively.

Full Ablation Studies Results

Due to space limitations, we provide the complete ablation results in this section. Unless otherwise specified, all ablation experiments are trained on 5,000 samples using NAFNet (Chen et al. 2022), following the same training setting described in Training Details in the main paper.

Effect of BPR and Artifact Filtering. Table A presents the complete results for BPR and artifact filtering. Consistent with the observations in the main paper, each component independently improves deblurring performance by mitigating misalignment or excluding severely corrupted samples, while their combination achieves the best results.

Effect of Loss Terms in BPR. We analyze the contribution of each loss term in the BPR module. Directly measuring the alignment between real blurry–sharp image pairs is challenging. Since alignment in the 3DGS framework is determined by camera poses, we evaluate BPR using pose errors and downstream deblurring performance. For pose errors, as ground-truth poses are unavailable for real blurry images, we construct a synthetic evaluation set where blurry images are generated using GS-Blur and the calibrated poses of the corresponding sharp images are regard as ground-truth. We report Translation Error (TE) and Rotation Error (RE) before and after pose optimization. For debluring performance, we train deblurring models on data pairs generated under different loss configurations. As shown in Table B, without pose optimization, severe misalignment leads to poor deblurring performance. $\mathcal{L}_{\text{appearance}}$ mainly reduces rotation error and substantially improves deblurring performance, whereas $\mathcal{L}_{\text{centroid}}$ primarily reduces translation error. Their combination achieves the lowest pose errors and the best deblurring performance, demonstrating their complementarity.

Effect of CutMix-Dynamic. We evaluate our designed CutMix-Dynamic on RealBlur (Rim et al. 2020) and RBVD (Zhu et al. 2022) for camera-motion blur, and RS-Blur (Rim et al. 2022) and BSD (Zhong et al. 2020) for object motion blur. Static-scene results reflect camera-blur handling, while dynamic-scene results measure robustness to object-motion blur. Results are reported in Table C. The original CutMix (Yun et al. 2019) improves performance on dynamic scenes but slightly degrades results on static scenes due to unrealistic local replacements. In contrast, CutMix-Dynamic introduces realistic object-motion blur and achieves the best overall performance.

Effect of Dataset Scale. The effect of dataset scale is shown in Table D. Performance consistently improves with increasing data scale, with the full scale achieving the best results. This trend highlights the benefit of scaling up datasets using GS-RealBlur.

Limitations and Future Work

Real-world motion blur may arise from both camera motion and independently moving objects. Owing to the static-scene limitation of 3DGS and the still-limited reconstruction fidelity of current 4DGS methods for complex dynamic scenes, our current data construction framework primarily models camera-motion blur. Although CutMix-Dynamic partially compensates for this limitation by injecting real object-motion blur patterns during training, it does not explicitly reconstruct or render dynamic scenes. In future work, advances in the reconstruction fidelity of 4D Gaussian Splatting (Wu et al. 2024, 2026) may enable our framework to be extended to dynamic scenes. High-speed multi-view capture could then be used to reconstruct 4D scenes and render sharp references for moving objects.

Table A: **Ablation on BPR and Artifact Filtering**

BPR	Artifact Filtering	RealBlur PSNR↑	RealBlur SSIM↑	RealBlur LPIPS↓	RBVD PSNR↑	RBVD SSIM↑	RBVD LPIPS↓	RSBlur PSNR↑	RSBlur SSIM↑	RSBlur LPIPS↓	BSD PSNR↑	BSD SSIM↑	BSD LPIPS↓
✗	✗	26.36	0.848	0.244	26.06	0.892	0.198	30.89	0.821	0.326	29.92	0.917	0.165
✓	✗	27.13	0.863	0.229	26.33	0.897	0.190	32.41	0.853	0.319	30.86	0.928	0.126
✗	✓	27.38	0.880	0.154	26.46	0.899	0.189	32.61	0.854	0.315	31.42	0.930	0.109
✓	✓	**27.57**	**0.885**	**0.142**	**26.67**	**0.908**	**0.187**	**33.13**	**0.861**	**0.312**	**31.63**	**0.936**	**0.098**

Table B: **Ablation on Loss Terms in BPR.**

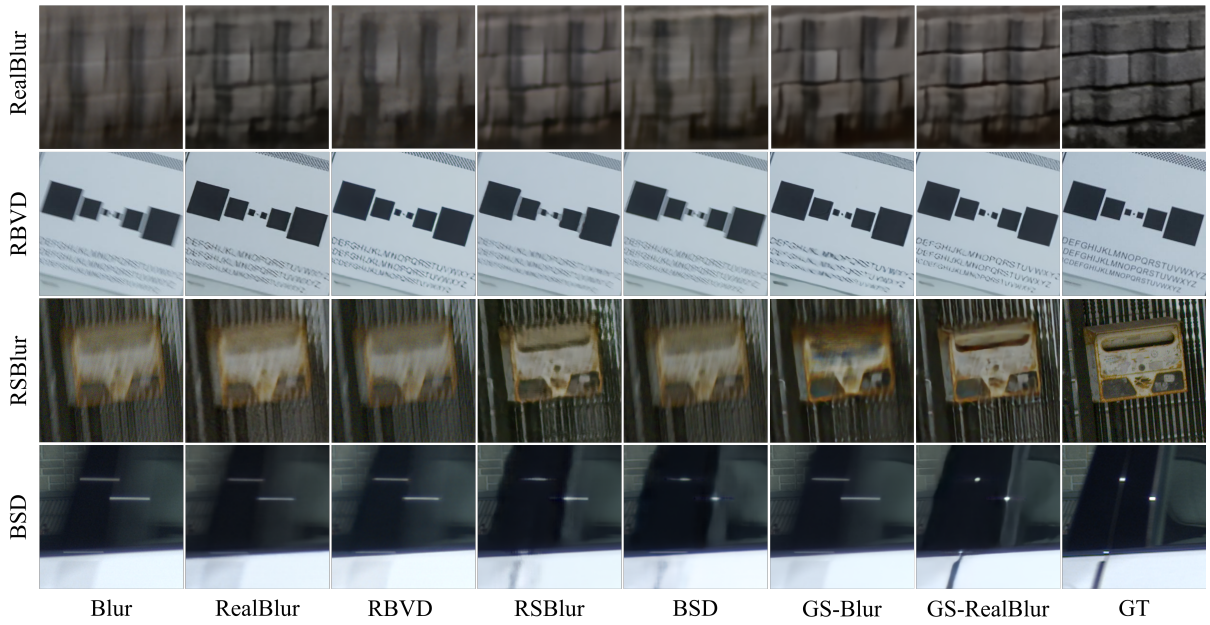
$\mathcal{L}_{appearance}$	$\mathcal{L}_{\text{centroid}}$	Pose Error TE↓ / RE↓	RealBlur PSNR↑	RealBlur SSIM↑	RealBlur LPIPS↓	RBVD PSNR↑	RBVD SSIM↑	RBVD LPIPS↓	RSBlur PSNR↑	RSBlur SSIM↑	RSBlur LPIPS↓	BSD PSNR↑	BSD SSIM↑	BSD LPIPS↓
✗	✗	1.036 / 1.102	27.38	0.880	0.154	26.46	0.899	0.189	32.61	0.854	0.315	31.42	0.930	0.109
✓	✗	0.702 / 0.503	27.52	0.884	0.143	26.63	0.907	**0.187**	32.95	0.858	0.314	31.59	0.935	0.102
✗	✓	0.452 / 0.841	27.43	0.883	0.146	26.52	0.903	0.188	32.75	0.856	0.315	31.51	0.934	0.105
✓	✓	**0.445 / 0.479**	**27.57**	**0.885**	**0.142**	**26.67**	**0.908**	**0.187**	**33.13**	**0.861**	**0.312**	**31.63**	**0.936**	**0.098**

Table C: **Ablation on CutMix-Dynamic**

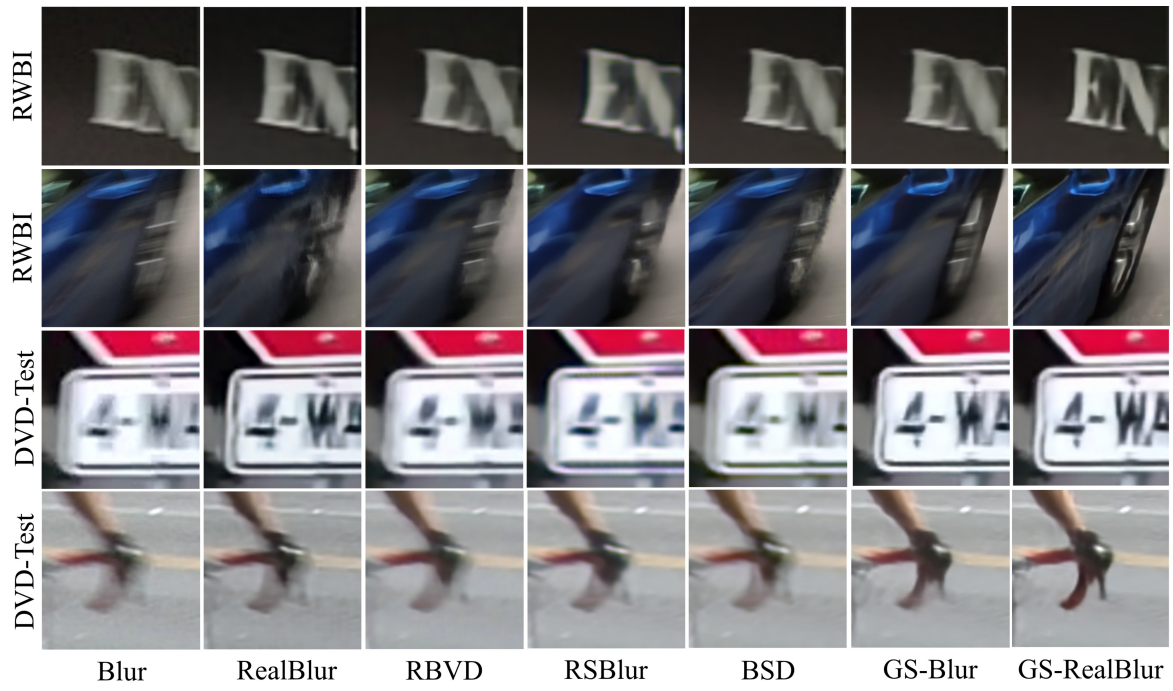
Methods	RealBlur PSNR↑	RealBlur SSIM↑	RealBlur LPIPS↓	RBVD PSNR↑	RBVD SSIM↑	RBVD LPIPS↓	RSBlur PSNR↑	RSBlur SSIM↑	RSBlur LPIPS↓	BSD PSNR↑	BSD SSIM↑	BSD LPIPS↓
Baseline	27.52	0.884	0.141	26.60	0.908	0.186	33.09	**0.860**	0.313	31.52	0.935	0.100
+ CutMix (Yun et al. 2019)	27.50	0.884	0.141	26.62	0.908	**0.185**	33.10	**0.860**	**0.312**	31.59	**0.936**	0.099
+ CutMix-Dynamic	**27.58**	**0.885**	**0.140**	**26.67**	**0.909**	**0.185**	**33.12**	**0.860**	**0.312**	**31.64**	**0.936**	**0.098**

Table D: **Ablation on Dataset Scale**

Training Data Proportions	RealBlur PSNR↑	RealBlur SSIM↑	RealBlur LPIPS↓	RBVD PSNR↑	RBVD SSIM↑	RBVD LPIPS↓	RSBlur PSNR↑	RSBlur SSIM↑	RSBlur LPIPS↓	BSD PSNR↑	BSD SSIM↑	BSD LPIPS↓
25%	27.52	0.883	0.143	26.47	0.907	0.190	33.08	0.862	0.314	31.58	0.935	0.102
50%	27.58	0.885	0.142	26.57	0.908	0.187	33.10	0.862	0.313	31.69	0.937	0.098
100%	**27.67**	**0.886**	**0.140**	**26.71**	**0.910**	**0.186**	**33.15**	**0.863**	**0.311**	**31.92**	**0.939**	**0.093**



(a) **Visual Comparison on Real-world Datasets**



(b) **Visual Comparison on In-the-wild Data**

Figure C: **Additional visual comparisons.**

References

Brooks, T.; and Barron, J. T. 2019. Learning to synthesize motion blur. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6840–6848.

Chen, H.; Wang, Y.; Guo, T.; Xu, C.; Deng, Y.; Liu, Z.; Ma, S.; Xu, C.; Xu, C.; and Gao, W. 2021a. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12299–12310.

Chen, L.; Chu, X.; Zhang, X.; and Sun, J. 2022. Simple baselines for image restoration. In *European conference on computer vision*, 17–33. Springer.

Chen, L.; Lu, X.; Zhang, J.; Chu, X.; and Chen, C. 2021b. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 182–192.

Cho, S.; and Lee, S. 2009. Fast motion deblurring. *ACM Transactions on Graphics*, 28(5): 1–8.

Cho, S.; and Lee, S. 2017. Convergence analysis of MAP based blur kernel estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 4808–4816.

Cho, S.-J.; Ji, S.-W.; Hong, J.-P.; Jung, S.-W.; and Ko, S.-J. 2021. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4641–4650.

Deng, S.; Ren, W.; Yan, Y.; Wang, T.; Song, F.; and Cao, X. 2021. Multi-scale separable network for ultra-high-definition video deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14030–14039.

Fergus, R.; Singh, B.; Hertzmann, A.; Roweis, S. T.; and Freeman, W. T. 2006. Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 25(3): 787–794.

Gao, H.; Tao, X.; Shen, X.; and Jia, J. 2019. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3848–3856.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Ke, J.; Wang, Q.; Wang, Y.; Milanfar, P.; and Yang, F. 2021. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5148–5157.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; Drettakis, G.; et al. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Köhler, R.; Hirsch, M.; Mohler, B.; Schölkopf, B.; and Harmeling, S. 2012. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *European conference on computer vision*, 27–40. Springer.

Krishnan, D.; and Fergus, R. 2009. Fast image deconvolution using hyper-Laplacian priors. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, 1033–1041.

Krishnan, D.; Tay, T.; and Fergus, R. 2011. Blind deconvolution using a normalized sparsity measure. In *CVPR 2011*, 233–240. IEEE.

Lee, D.; Park, J.; and Lee, K. M. 2024. GS-blur: A 3d scene-based dataset for realistic image deblurring. *Advances in Neural Information Processing Systems*, 37: 125394–125415.

Li, D.; Xu, C.; Zhang, K.; Yu, X.; Zhong, Y.; Ren, W.; Suominen, H.; and Li, H. 2021. Arvo: Learning all-range volumetric correspondence for video deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7721–7731.

Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; and Timofte, R. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1833–1844.

Lin, J.; Zhang, Z.; Li, W.; Pei, R.; Xu, H.; Zhang, H.; and Zuo, W. 2026. UniRestorer: Universal image restoration via adaptively estimating image degradation at proper granularity. In *ICLR*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Loshchilov, I.; and Hutter, F. 2017a. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Loshchilov, I.; and Hutter, F. 2017b. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*.

Lucy, L. B. 1974. An iterative technique for the rectification of observed distributions. *Astronomical Journal, Vol. 79, p. 745 (1974)*, 79: 745.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Nah, S.; Baik, S.; Hong, S.; Moon, G.; Son, S.; Timofte, R.; and Lee, K. M. 2019. NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study. In *32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2019*, 1996–2005. IEEE Computer Society.

Nah, S.; Hyun Kim, T.; and Mu Lee, K. 2017. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3883–3891.

Pan, L.; Baráth, D.; Pollefeys, M.; and Schönberger, J. L. 2024. Global structure-from-motion revisited. In *European Conference on Computer Vision*, 58–77. Springer.

Park, D.; Kang, D. U.; Kim, J.; and Chun, S. Y. 2020. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *European conference on computer vision*, 327–343. Springer.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 8026–8037.

Richardson, W. H. 1972. Bayesian-based iterative method of image restoration. *Journal of the optical society of America*, 62(1): 55–59.

Rim, J.; Kim, G.; Kim, J.; Lee, J.; Lee, S.; and Cho, S. 2022. Realistic blur synthesis for learning image deblurring. In *European conference on computer vision*, 487–503. Springer.

Rim, J.; Lee, H.; Won, J.; and Cho, S. 2020. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *European conference on computer vision*, 184–201. Springer.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Shen, Z.; Wang, W.; Lu, X.; Shen, J.; Ling, H.; Xu, T.; and Shao, L. 2019. Human-aware motion deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5572–5581.

Su, J.; Xu, B.; and Yin, H. 2022. A survey of deep learning approaches to image restoration. *Neurocomputing*, 487: 46–65.

Su, S.; Delbracio, M.; Wang, J.; Sapiro, G.; Heidrich, W.; and Wang, O. 2017. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1279–1288.

Tao, X.; Gao, H.; Shen, X.; Wang, J.; and Jia, J. 2018. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8174–8182.

Tsai, F.-J.; Peng, Y.-T.; Lin, Y.-Y.; Tsai, C.-C.; and Lin, C.-W. 2022. Stripformer: Strip transformer for fast image deblurring. In *European conference on computer vision*, 146–162. Springer.

Varghese, R.; and Sambath, M. 2024. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International conference on advances in data engineering and intelligent computing systems (ADICS)*, 1–6. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J.; Chan, K. C.; and Loy, C. C. 2023. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, 2555–2563.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wang, Z.; Cun, X.; Bao, J.; Zhou, W.; Liu, J.; and Li, H. 2022. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17683–17693.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 2024. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 20310–20320.

Wu, R.; Zhang, Z.; Chen, M.; Yan, Z.; and Zuo, W. 2026. Deblur4dgs: 4d gaussian splatting from blurry monocular video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 10727–10735.

Xu, H.; Zhang, Z.; Fan, J.; Wu, X.; and Zuo, W. 2025. Self-HVD: Self-Supervised Handheld Video Deblurring for Mobile Phones. *arXiv e-prints*, arXiv–2508.

Xu, L.; Zheng, S.; and Jia, J. 2013. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1107–1114.

Yang, S.; Wu, T.; Shi, S.; Lao, S.; Gong, Y.; Cao, M.; Wang, J.; and Yang, Y. 2022. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1191–1200.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6023–6032.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; and Yang, M.-H. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5728–5739.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; Yang, M.-H.; and Shao, L. 2021. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14821–14831.

Zhai, L.; Wang, Y.; Cui, S.; and Zhou, Y. 2023. A comprehensive review of deep learning-based real-world image restoration. *Ieee Access*, 11: 21049–21067.

Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Stenger, B.; Liu, W.; and Li, H. 2020. Deblurring by realistic blurring. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2737–2746.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhang, Z.; Wang, H.; Liu, M.; Wang, R.; Zhang, J.; and Zuo, W. 2021. Learning raw-to-srgb mappings with inaccurately aligned supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4348–4358.

Zhang, Z.; Zhang, S.; Wu, R.; Yan, Z.; and Zuo, W. 2025. Exposure bracketing is all you need for a high-quality image. In *International Conference on Learning Representations*, volume 2025, 95393–95418.

Zhong, Z.; Gao, Y.; Zheng, Y.; and Zheng, B. 2020. Efficient spatio-temporal recurrent neural network for video deblurring. In *European conference on computer vision*, 191–207. Springer.

Zhou, S.; Zhang, J.; Pan, J.; Xie, H.; Zuo, W.; and Ren, J. 2019a. Spatio-temporal filter adaptive network for video deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2482–2491.

Zhou, S.; Zhang, J.; Zuo, W.; Xie, H.; Pan, J.; and Ren, J. S. 2019b. Davanet: Stereo deblurring with view aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10996–11005.

Zhu, C.; Dong, H.; Pan, J.; Liang, B.; Huang, Y.; Fu, L.; and Wang, F. 2022. Deep recurrent neural network with multi-scale bi-directional propagation for video deblurring. In *Proceedings of the AAAI conference on artificial intelligence*, 3598–3607.