IMD-BLUR: IMPLICIT MOTION AND DEFOCUS BLUR AUGMENTATION FOR IMAGE DEBLURRING

Anonymous authors
Paper under double-blind review

ABSTRACT

The scarcity of paired sharp/blur training data for image deblurring makes realistic blur synthesis important for data augmentation and model generalization. However, existing blur augmentation methods typically focus on isolated blur types, such as defocus or motion blur, and often rely on auxiliary signals, including optical flow, aperture settings, or depth/defocus maps, which are rarely available in practice. To address these limitations, we propose IMD-Blur (Implicit Motion and Defocus Blur Augmentation), a unified framework that learns pixel-wise blur parameters directly from paired sharp/blur images without external signals. Specifically, using a parameterized kernel-based blurring pipeline and a blur parameter estimator, IMD-Blur constructs a continuous and interpretable parameter space for observed blur patterns. Based on the estimated blur parameters, we design a diffusion-based blur generator that produces diverse and controllable motion- and defocus-blurred images, thereby densifying the observed blur distribution and improving training-distribution coverage. Experiments show that IMD-Blur produces realistic blurred images and consistently improves state-of-the-art deblurring models across multiple benchmarks.

1 INTRODUCTION

Blur is a pervasive degradation in natural imaging and can be broadly categorized into motion blur and defocus blur. Motion blur arises from relative movement between the camera and the scene during exposure, leading to spatially varying and direction-dependent artifacts. Defocus blur occurs when scene points are imaged away from the camera's conjugate focal plane, producing depth-dependent blur governed by aperture and optical settings. Consequently, image deblurring is crucial for recovering sharp images from blurred observations; yet, it remains ill-posed in real scenes where blur can vary spatially and arise from mixed camera/object motion and depth-dependent defocus.

Image deblurring has advanced rapidly with deep learning. CNN-based methods (Abuolaim & Brown, 2020; Cho et al., 2021; Cui et al., 2023; 2024; Gao et al., 2019; Lee et al., 2021) first established impressive results, followed by Transformer-based approaches (Chen et al., 2025a; Tsai et al., 2022; Wang et al., 2022; Zamir et al., 2022; Zhang et al., 2025) that improved long-range dependency modeling. More recently, Mamba-based state-space models (Chen et al., 2025b; Kong et al., 2025; Li et al., 2025; Lin et al., 2025; Sun et al., 2025) have emerged as efficient alternatives to attention-based designs. Despite these architectural advances, deblurring performance is often limited by insufficient coverage of blur variations in the training distribution, which weakens model generalization. Blur synthesis methods (Ignatov et al., 2020; Kim et al., 2025; Seizinger et al., 2025; Wu et al., 2024; Zhu et al., 2025) have therefore been proposed for data augmentation, but their practical use remains constrained by several limitations. First, many methods address either motion or defocus blur, lacking a unified mechanism for both. Second, motion-blur synthesis often relies on simplistic linear trajectories, failing to model realistic non-linear motion patterns. Third, approaches, such as ID-Blau (Wu et al., 2024), BokehDiff (Zhu et al., 2025), and BokehLicious (Seizinger et al., 2025), require external signals, like video optical flow, depth/defocus maps, or camera/aperture metadata, which are often unavailable in practice. These limitations motivate a unified blur augmentation framework that supports motion and defocus blur, avoids auxiliary signals, and generates additional domain-specific blur samples that better cover the training distribution to improve generalization.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

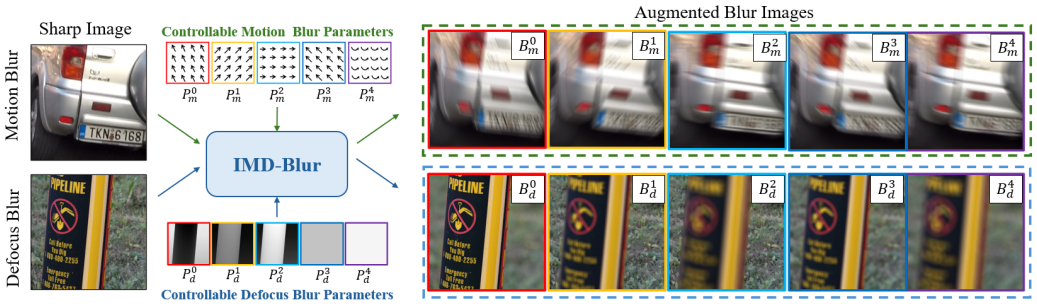


Figure 1: **Continuous blur synthesis via IMD-Blur.** IMD-Blur synthesizes motion and defocus blurred images (B_m^i and B_d^i) conditioned on the corresponding blur parameters (P_m^i and P_d^i), enabling **pixel-wise controllable** generation of diverse motion and defocus blur by varying these parameters.

In this paper, we propose IMD-Blur (Implicit Motion and Defocus Blur Augmentation), a unified framework for controllable synthesis of both motion and defocus blur. By implicitly estimating pixel-wise blur parameters from paired sharp/blur images, IMD-Blur operates without the need for auxiliary signals. It consists of two primary stages: (i) constructing a continuous and interpretable blur parameter space using a Parameterized Kernel-based Blurring Pipeline and a blur parameter estimator, and (ii) generating diverse and controllable motion- and defocus-blurred images from the estimated parameters using a Diffusion-based Blur Generator.

In the first stage, the parameter estimator infers interpretable, pixel-wise blur parameters that form a continuous blur representation. Specifically, motion blur is parameterized by magnitude, orientation, and curvature to support controllable non-linear trajectories, while defocus blur is represented by the circle-of-confusion (CoC) radius. Due to the absence of explicit annotations, these parameters are learned implicitly from paired sharp/blur images. They are subsequently mapped to motion and defocus kernels to synthesize blurred images, which are supervised by paired ground-truth blurred images. This reconstruction-based supervision enables implicit learning of a continuous blur-parameter space. Using the estimated blur parameters as guidance, we propose a Diffusion-based Blur Generator to produce realistic outputs aligned with the training blur distribution. By using ground-truth blurred images as supervision, the generator learns domain-specific characteristics, including color shifts, saturation variations, and sensor- or environment-induced artifacts. As shown in Fig. 1, IMD-Blur samples and adjusts blur parameters to generate diverse motion- and defocus-blurred images that better cover the underlying blur distribution, thereby providing effective augmented data for improving model generalization.

Our contributions are summarized as follows. First, we propose IMD-Blur, a unified framework for controllable augmentation of motion and defocus blur using solely paired sharp/blur images, without auxiliary signals. Second, we design a parameterized kernel-based pipeline to estimate pixel-wise blur parameters, including non-linear motion blur via curvature modeling. Third, we introduce a Diffusion-based Blur Generator that uses the estimated parameters as guidance to generate realistic, domain-aligned blurred images. Finally, extensive experiments show that IMD-Blur significantly improves representative deblurring models and achieves state-of-the-art performance on multiple motion and defocus benchmarks.

2 Related Work

Image Deblurring. With the rapid advancement of deep learning, CNN-based deblurring methods (Abuolaim & Brown, 2020; Cho et al., 2021; Cui et al., 2023; 2024; Gao et al., 2019; Lee et al., 2021; Nah et al., 2017; Park et al., 2020; Son et al., 2021; Tao et al., 2018; Zamir et al., 2021; Zhang et al., 2019) have achieved impressive progress. For instance, Cho et al. (2021) pioneered an efficient coarse-to-fine strategy by using multi-scale inputs and multi-scale outputs within a single network. Cui et al. (2023) introduced a dual-domain selection mechanism with multi-scale feature splitting to improve restoration quality and efficiency. Following the success of Transformers, numerous works (Chen et al., 2025a; 2021; Hu et al., 2025; Kong et al., 2023; Kuo et al., 2025; Mao et al., 2024; Tsai et al., 2022; Wang et al., 2022; Zamir et al., 2022; Zhang et al., 2025) have adopted attention-based architectures to model long-range dependencies and global context for deblurring. For example, Tsai et al. (2022) designed a strip-based attention scheme to handle blur patterns with

diverse orientations and strengths. Chen et al. (2025a) proposed a motion-decomposition Transformer, which separates translational and rotational motion to better manipulate complex blur. More recently, Mamba-based models (Chen et al., 2025b; Kong et al., 2025; Li et al., 2025; Lin et al., 2025; Sun et al., 2025) have emerged as an efficient alternative to Transformers. By using selective state-space modeling, these models capture long-range features with lower computational cost. For instance, Lin et al. (2025) developed a multi-head selective scan with an all-around scanning mechanism. Kong et al. (2025) presented an efficient visual scan block based on geometric transformations.

Despite these architectural innovations, deblurring performance is frequently bottlenecked by the limited coverage of blur variations in the underlying distribution, which weakens model generalization. This limitation stems from the scarcity of paired sharp/blur training data, which captures only a small fraction of real-world blur patterns, scene complexities, and imaging conditions. Moreover, acquiring large-scale paired sharp/blur data remains labor-intensive and often requires specialized dual-camera systems with precise hardware synchronization and spatial alignment. Consequently, synthesizing diverse and realistic blur patterns for training has become an important direction for improving the generalization of modern deblurring models.

Blur Synthesis and Augmentation. Beyond network architecture innovations, studies (Chi et al., 2021; He et al., 2024; Lee et al., 2021; Li et al., 2022; Nah et al., 2022; Yang et al., 2023; Zhang et al., 2020) have improved deblurring through reblurring models. Zhang et al. (2020) proposed a GAN-based approach to generate blurred images for training deblurring models. Nah et al. (2022), Lee et al. (2021), and Yang et al. (2023) improved deblurring by reblurring restored images and enforcing consistency with the original blurred counterparts. However, these approaches are not designed as controllable frameworks, limiting their ability to systematically generate diverse training data.

In parallel, recent advancements in bokeh rendering (Ignatov et al., 2020; Kaneko, 2021; Peng et al., 2022a; 2024; 2022b; Seizinger et al., 2025; Sheng et al., 2024; Zhu et al., 2025) aim to synthesize shallow depth-of-field effects with realistic defocus appearance. For example, Seizinger et al. (2025) proposed an aperture-aware attention mechanism that uses aperture annotations to simulate defocus effects. Zhu et al. (2025) introduced a physics-inspired self-attention module that incorporates depth and defocus maps to synthesize defocus effects. However, these methods often rely on auxiliary signals, such as aperture settings, depth maps, or defocus maps, which are not consistently available in deblurring datasets. Furthermore, they mainly target defocus rendering and do not directly support motion blur synthesis.

More recently, motion blur augmentation (Kim et al., 2025; Wu et al., 2024) has attracted increasing attention for controllable motion blur synthesis. Wu et al. (2024) parameterized blur patterns as pixel-wise blur condition maps and synthesized blurred images conditioned on different maps. Kim et al. (2025) estimated 3D camera poses over the exposure interval, rendered the corresponding views, and aggregated them to generate motion blur. However, Wu et al. (2024) relied on video sequences and optical flow estimation to derive blur conditions, which are usually unavailable in real-world deblurring datasets. Moreover, both methods focus on motion blur augmentation and do not jointly support defocus synthesis or non-linear motion blur generation. In contrast, IMD-Blur provides a unified blur augmentation framework that addresses three limitations of prior methods: (i) it jointly supports motion and defocus blur; (ii) it learns solely from paired sharp/blur images without auxiliary signals; and (iii) it synthesizes diverse blur variations, including non-linear motion blur, by sampling a learned, continuous blur-parameter space.

3 Proposed Method

We propose Implicit Motion and Defocus Blur Augmentation (IMD-Blur), a unified framework for controllable synthesis of motion and defocus blur by learning a continuous, pixel-wise blur-parameter space from paired sharp/blur images. As illustrated in Fig. 2, IMD-Blur is trained using only paired sharp/blur images and comprises two primary stages: (i) blur-parameter estimation via a Parameterized Kernel-based Blurring Pipeline with a blur parameter estimator, and (ii) parameter-guided blur generation via a Diffusion-based Blur Generator.

In the first stage, the blur parameter estimator predicts pixel-wise blur parameters in a continuous space, forming a data-driven blur-parameter field. To optimize these parameters, they are mapped to spatially varying motion or defocus kernels and applied to the sharp image to synthesize a kernel-based blurred image. The synthesis is supervised by the paired ground-truth blurred image, en-

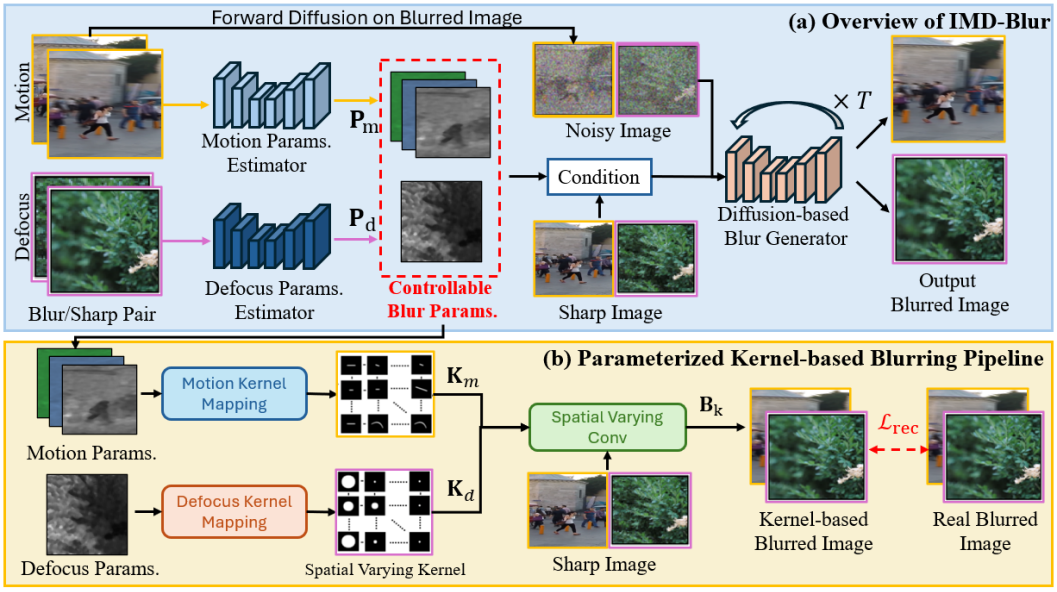


Figure 2: **Overview of the IMD-Blur pipeline.** (a) Given a blur/sharp pair $(\mathbf{B}, \mathbf{S})$, we estimate per-pixel blur parameters. Guided by these parameters and the sharp image, the Diffusion-based Blur Generator learns to generate realistic blurred images. (b) *Parameterized Kernel-based Blurring Pipeline*: Given a blur/sharp pair $(\mathbf{B}, \mathbf{S})$, we estimate per-pixel blur parameters, map them to spatially varying kernels $\mathbf{K}$, and synthesize a kernel-based blurred image $\mathbf{B}_k$ by reconstructing $\mathbf{B}$ to implicitly optimize the blur parameter space.

abling implicit parameter learning without explicit annotations. In the second stage, the Diffusion-based Blur Generator uses the estimated parameters as guidance to generate realistic outputs aligned with the training blur distribution, including domain-specific color shifts, saturation variations, and sensor- or environment-induced artifacts. During inference, IMD-Blur samples or perturbs blur parameters to generate diverse blurred images that densify the observed blur distribution and improve training-distribution coverage for deblurring model training.

3.1 Parameterized Kernel-based Blurring Pipeline

Our objective is to develop a controllable blur synthesis framework for both motion and defocus blur using only paired sharp/blur images, without any auxiliary signals (e.g., optical flow, depth/defocus maps, or camera metadata). To this end, we propose a *Parameterized Kernel-based Blurring Pipeline (PKBP)*, which learns a continuous and interpretable pixel-wise blur-parameter field to guide the subsequent Diffusion-based Blur Generator.

As shown in Fig. 2, given a paired training sample $(\mathbf{S}, \mathbf{B})$, where $\mathbf{S} \in \mathbb{R}^{H\times W\times 3}$ is the sharp image and $\mathbf{B} \in \mathbb{R}^{H\times W\times 3}$ is the corresponding blurred observation, we first estimate blur parameters using a dedicated estimator and then convert them into a spatially varying kernel field through our kernel mapping module. We consider two distinct blur parameterizations.

Motion Blur. Inspired by (Fang et al., 2025; Wu et al., 2024), we represent motion blur with a pixel-wise motion parameter field $\mathbf{P}_m$:

$$\mathbf{P}_m = [\mathbf{M}, \mathbf{O}, \mathbf{C}] = E_m(\mathbf{S}, \mathbf{B}) \in \mathbb{R}^{H\times W\times 3}, \tag{1}$$

where $\mathbf{P}_m$ comprises the pixel-wise motion magnitude $\mathbf{M}$, motion orientation $\mathbf{O}$, and curvature $\mathbf{C}$, all generated by applying the motion-branch parameter estimator $E_m(\cdot)$ to the image pair. By introducing the curvature $\mathbf{C}$ to account for trajectory non-linearity, this parameterization enables controllable modeling of both linear and non-linear motion patterns.

Defocus Blur. Inspired by Circle-of-Confusion (CoC) modeling (Abuolaim et al., 2021), we represent defocus blur using a pixel-wise CoC radius field $\mathbf{P}_d$:

$$\mathbf{P}_d = \mathbf{R} = E_d(\mathbf{S}, \mathbf{B}) \in \mathbb{R}^{H\times W\times 1}, \tag{2}$$

where $E_d(\cdot)$ denotes the defocus-branch parameter estimator and $\mathbf{R}$ is the predicted CoC radius. Given the estimated parameter fields, we construct a spatially varying kernel field $\mathbf{K}$ through the following kernel mappings:

$$\mathbf{K} \in \mathbb{R}^{H\times W\times k\times k} = \begin{cases} \mathbf{K}_m = \mathcal{M}(\mathbf{P}_m), & \text{for motion blur,} \\ \mathbf{K}_d = \mathcal{D}(\mathbf{P}_d), & \text{for defocus blur,} \end{cases} \tag{3}$$

where k denotes the kernel size, $\mathcal{M}(\cdot)$ and $\mathcal{D}(\cdot)$ are the motion and defocus kernel mappings, respectively, and $\mathbf{K}_m$ and $\mathbf{K}_d$ represent the corresponding kernel fields (elaborated in Sec. 3.2). In our setting, the blur type is determined by the data acquisition protocol and is treated as known during training and augmentation. We then synthesize a kernel-based blurred image, $\mathbf{B}_k = \mathcal{G}(\mathbf{S};\mathbf{K}) \in \mathbb{R}^{H\times W\times 3}$, where $\mathcal{G}(\cdot)$ denotes a differentiable, spatially varying convolution operator that applies the per-pixel kernels in $\mathbf{K}$ to $\mathbf{S}$.

Training objective. The PKBP is optimized using only paired images by minimizing the reconstruction discrepancy between $\mathbf{B}_k$ and the observed blur $\mathbf{B}$ in both spatial and frequency domains:

$$\mathcal{L}_{\text{rec}} = \|\mathbf{B}_k - \mathbf{B}\|_1 + \lambda \|\mathcal{F}(\mathbf{B}_k) - \mathcal{F}(\mathbf{B})\|_1 , \tag{4}$$

where $\mathcal{F}(\cdot)$ denotes the Fourier transform (implemented with FFT), and λ is empirically set to 0.1. This reconstruction-based supervision encourages the learned blur parameters to faithfully reproduce the observed blur, enabling implicit learning of a continuous blur-parameter space.

After optimization, PKBP establishes a controllable and continuous parameter space represented by the motion parameter field $\mathbf{P}_m$ or the defocus parameter field $\mathbf{P}_d$. By sampling or perturbing this space, the resulting parameters guide the Diffusion-based Blur Generator to produce diverse, domain-specific blurred images for data augmentation (Sec. 3.4). Next, we detail the motion and defocus kernel mappings.

3.2 Motion and Defocus Kernel Mappings

Following the prediction of the blur-parameter fields, we construct spatially varying kernel fields through two differentiable mapping functions: the motion kernel mapping $\mathcal{M}(\cdot)$ and the defocus kernel mapping $\mathcal{D}(\cdot)$.

Motion Blur Kernel Mapping. Given the motion parameter field $\mathbf{P}_m = [\mathbf{M}, \mathbf{O}, \mathbf{C}]$, we construct a spatially varying motion kernel field $\mathbf{K}_m = \mathcal{M}(\mathbf{P}_m) \in \mathbb{R}^{H\times W\times k\times k}$. We employ a fixed odd kernel size k to ensure a well-defined kernel center. For each pixel (i, j), the mapping $\mathcal{M}(\cdot)$ generates a discrete kernel $K_{ij} \in \mathbb{R}^{k\times k}$ by rasterizing the corresponding motion trajectory within the kernel support, followed by unit-sum normalization to preserve average intensity.

Specifically, we adopt a kernel-centered coordinate system with the origin at $(0, 0)$ and parameterize a reference trajectory by a scalar variable t. For each pixel (i, j), the predicted magnitude $M_{ij} \geq 0$ defines N uniformly spaced samples $\{t_n\}_{n=0}^{N-1}$ over the interval $\left[-\frac{M_{ij}}{2}, \frac{M_{ij}}{2}\right]$ with $N \geq 2$: $t_n = -\frac{M_{ij}}{2} + \frac{n}{N-1} M_{ij},$ for $n = 0, 1, \ldots, N-1$, where t_n represents a 1D coordinate along the reference path. Given the predicted curvature C_{ij}, we map each scalar sample t_n to a 2D point on a reference trajectory in the kernel-centered coordinate system, i.e., $x_n = t_n$ and $y_n = C_{ij}\, t_n^2$, where (x_n, y_n) denotes the 2D coordinate of the n-th sample on the reference path. To account for the predicted orientation O_{ij}, we rotate the reference trajectory by θ_{ij} and obtain the oriented 2D displacements:

$$\begin{bmatrix} u_n \\ v_n \end{bmatrix} = \begin{bmatrix} \cos\theta_{ij} & -\sin\theta_{ij} \\ \sin\theta_{ij} & \cos\theta_{ij} \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}, \text{ where } \theta_{ij} = O_{ij} \cdot \frac{\pi}{180}, \tag{5}$$

and (u_n, v_n) denotes the rotated 2D displacement of the n-th sample in the kernel coordinate system. To discretize these continuous samples $\{(u_n, v_n)\}_{n=0}^{N-1}$ onto the $k \times k$ kernel grid, we adopt differentiable bilinear splatting. Let $r = \lfloor k/2 \rfloor$ and $(p, q) \in \{-r, \ldots, r\}^2$ denote discrete kernel coordinates. Each sample contributes to the kernel bins via the bilinear basis:

$$\tilde{K}_{ij}(p, q) = \sum_{n=0}^{N-1} \phi(u_n - p)\, \phi(v_n - q), \text{ where } \phi(x) = \max(0, 1 - |x|), \tag{6}$$

and $\phi(\cdot)$ is the 1D tent function and makes samples outside the kernel support contribute zero. Finally, we enforce unit-sum normalization:

$$K_{ij}(p, q) = \frac{\tilde{K}_{ij}(p, q)}{\sum_{p,q} \tilde{K}_{ij}(p, q) + \epsilon}. \tag{7}$$

In practice, we set ϵ to a small value for numerical stability. Stacking K_{ij} over all pixels yields the spatially varying motion kernel field $\mathbf{K}_m$.

Defocus Blur Kernel Mapping. For defocus blur, we construct a spatially varying defocus kernel field $\mathbf{K}_d = \mathcal{D}(\mathbf{P}_d) \in \mathbb{R}^{H\times W\times k\times k}$ from the CoC radius field $\mathbf{P}_d = \mathbf{R}$. For each pixel (i, j), the mapping $\mathcal{D}(\cdot)$ generates a discrete defocus kernel $K_{ij} \in \mathbb{R}^{k\times k}$ by rasterizing a disk point-spread function (PSF) of radius R_{ij}, where R_{ij} is measured in pixels in the kernel coordinate system. We adopt the same kernel-centered coordinate system and define the kernel half-size as $r = \lfloor k/2 \rfloor$ with discrete kernel coordinates $(p, q) \in \{-r, \dots, r\}^2$, and $R_{ij} \le r$. While an ideal defocus PSF is a binary disk, discretizing its hard boundary is non-differentiable with respect to R_{ij}. For gradient-based optimization, we employ a differentiable soft-disk formulation using a sigmoid transition:

$$\tilde{K}_{ij}(p, q) = \sigma\Big(\alpha\left(R_{ij} - \sqrt{p^2 + q^2}\right)\Big), \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function and α controls the edge softness (larger α yields a sharper boundary). We normalize each kernel using Eq. 7, and stacking K_{ij} over all pixels yields the spatially varying defocus kernel field $\mathbf{K}_d$. In summary, $\mathcal{M}(\cdot)$ and $\mathcal{D}(\cdot)$ map continuous interpretable blur-parameter fields to spatially varying kernels, enabling end-to-end learning of implicit blur parameter estimation within PKBP.

3.3 Diffusion-based Blur Generator

Building on the continuous and controllable blur parameter space learned by PKBP, we further introduce a *Diffusion-based Blur Generator (DBG)* to generate realistic, domain-specific motion- and defocus-blurred images, as shown in Fig. 2 (a). Guided by the estimated blur parameters $\mathbf{P} \in \{\mathbf{P}_m, \mathbf{P}_d\}$, DBG learns the reverse diffusion process under the supervision of real blurred images. This enables it to reproduce domain-specific characteristics, including ISP-induced color shifts, saturated light trails, and saturation artifacts. Given a sharp image and its blur parameters, DBG produces a blurred image $\mathbf{B}_d \in \mathbb{R}^{H\times W\times 3}$, as detailed below.

Training Stage. We adopt the DDPM formulation (Ho et al., 2020) and train DBG via conditional noise prediction. Let $\mathbf{x}_0 = \mathbf{B}$ denote the target blurred image in each training pair. The forward process progressively perturbs $\mathbf{x}_0$ with Gaussian noise to obtain a noisy sample $\mathbf{x}_t$ at timestep t: $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0,\ (1 - \bar{\alpha}_t)\mathbf{I})$, where $\{\alpha_t\}_{t=1}^T$ is a predefined noise schedule and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. We optimize DBG with the standard objective (Ho et al., 2020): $\mathcal{L}_{\text{diff}} = \big\|\epsilon - \epsilon_\phi(\mathbf{x}_t, t \mid \mathbf{P}, \mathbf{S}, \mathbf{B}_k)\big\|_1$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{B} + \sqrt{1 - \bar{\alpha}_t}\epsilon$, and $\epsilon_\phi(\cdot)$ is the conditional noise predictor. Conditioning on $\mathbf{P}$ guides controllable blur generation, conditioning on $\mathbf{B}_k$ preserves the blur structure obtained from kernel-based synthesis, and conditioning on $\mathbf{S}$ provides scene content, allowing DBG to learn domain-specific blur characteristics from paired blurred images.

Inference Stage. During inference, we employ deterministic DDIM sampling (Song et al., 2021) conditioned on $(\mathbf{P}, \mathbf{S}, \mathbf{B}_k)$. Starting from Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we iteratively update $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$ via

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\,\epsilon_\phi(\mathbf{x}_t, t \mid \mathbf{P}, \mathbf{S}, \mathbf{B}_k)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1 - \bar{\alpha}_{t-1}}\,\epsilon_\phi(\mathbf{x}_t, t \mid \mathbf{P}, \mathbf{S}, \mathbf{B}_k). \tag{9}$$

After the last iteration, we obtain the generated blurred image $\mathbf{B}_d = \mathbf{x}_0$, which serves as the synthesized blur used for augmentation.

3.4 Controllable Blur Data Augmentation

After training, IMD-Blur serves as a controllable blur generator for synthesizing diverse training pairs. To ensure that the generated samples remain aligned with the observed blur distribution, we implement a distribution-aware sampling strategy. We first run the trained blur parameter estimator on all training pairs to extract pixel-wise parameter fields, and then aggregate these parameter values over the entire training set to estimate the data-driven blur-parameter distributions of blur parameters, e.g., the magnitude, orientation, curvature, and CoC radius histograms shown in Fig. 6. During augmentation, we sample sharp images from the training set together with blur parameters from the estimated distributions, and then synthesize diverse blurred images using IMD-Blur. These augmented samples densify the observed blur distribution and improve generalization in deblurring.

Table 1: Evaluation results on the RealDOF, RTF, and DPDD (Indoor/Outdoor/Combined) datasets, where "Baseline" and "+IMD-Blur" denote the deblurring performances without and with IMD-Blur, respectively.

Model		RealDOF		RTF		DPDD-Indoor		DPDD-Outdoor		DPDD-Combined	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MIMO-UNet	Baseline	24.24	0.770	22.78	0.742	28.07	0.871	22.82	0.749	25.38	0.808
	+IMD-Blur	**24.73 (+0.49)**	**0.787**	**24.22 (+1.44)**	**0.802**	**28.51 (+0.44)**	**0.884**	**23.33 (+0.51)**	**0.763**	**25.85 (+0.47)**	**0.822**
NRKNet	Baseline	25.14	0.768	25.89	0.830	28.77	0.878	23.43	0.754	26.03	0.815
	+IMD-Blur	**25.46 (+0.32)**	**0.788**	**26.08 (+0.19)**	**0.843**	**29.01 (+0.24)**	**0.881**	**23.54 (+0.11)**	**0.756**	**26.21 (+0.18)**	**0.819**
EVSSM	Baseline	25.00	0.788	24.88	0.846	29.17	0.888	23.32	0.758	26.17	0.821
	+IMD-Blur	**25.76 (+0.76)**	**0.804**	**26.13 (+1.25)**	**0.867**	**29.54 (+0.37)**	**0.894**	**23.70 (+0.38)**	**0.776**	**26.55 (+0.38)**	**0.833**
Avg Gains		**+0.52**	**+0.018**	**+0.96**	**+0.031**	**+0.35**	**+0.007**	**+0.33**	**+0.011**	**+0.34**	**+0.010**

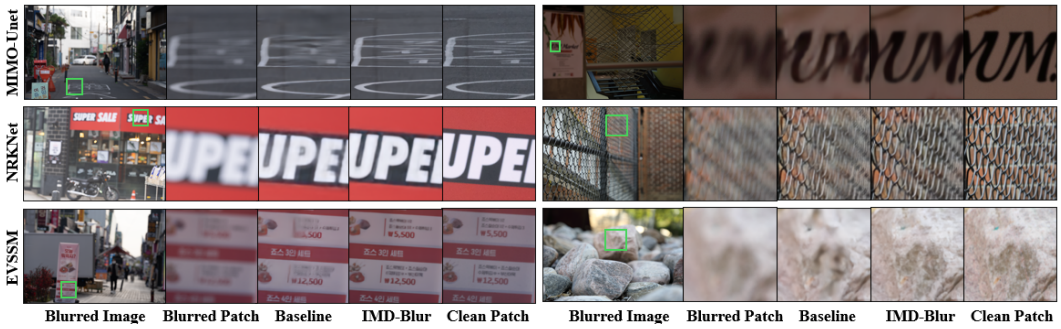


Figure 3: Qualitative results on the RealDOF (left) and DPDD (right) dataset.

4 Experiments

4.1 Experimental Setup

Deblurring Models and Datasets. We adopt four representative deblurring models, MIMO-UNet (Cho et al., 2021), Stripformer (Tsai et al., 2022), NRKNet (Quan et al., 2023), and EVSSM (Kong et al., 2025), to validate the effectiveness of IMD-Blur. Following prior work, we evaluate IMD-Blur on both defocus and motion deblurring benchmarks. For defocus deblurring, we adopt three real-world datasets, DPDD (Abuolaim & Brown, 2020), RealDOF (Lee et al., 2021), and RTF (D'Andrès et al., 2016); RealDOF and RTF are used only for cross-dataset evaluation. For motion deblurring, we evaluate on two synthetic datasets, GoPro (Nah et al., 2017) and HIDE (Shen et al., 2019), as well as two real-world datasets, RealBlur-J (Rim et al., 2020) and RealBlur-R (Rim et al., 2020). HIDE is used only for testing.

Implementation Details. We implement both the blur parameter estimator and the diffusion noise estimator using the same U-Net architecture (Wu et al., 2024). For motion blur synthesis, IMD-Blur is trained on the GoPro training set for 3000 epochs. For defocus blur synthesis, IMD-Blur is trained on the DPDD training set for 300 epochs, with preprocessing following (Abuolaim & Brown, 2020). Both blur types share the same training configuration: batch size 8, random cropping to 256×256, the Adam optimizer, and a cosine-annealed learning rate from 1×10^{-4} to 1×10^{-6}. After optimizing IMD-Blur, we generate an additional augmented training set that is five times larger than the original training data using our data-driven blur-parameter sampling strategy. Each deblurring model is pre-trained on these pairs for 500 epochs and then fine-tuned with its default protocol (see supplementary material).

4.2 Experimental Results

Image Defocus Deblurring Results. We compare single-image defocus deblurring performance of three deblurring models and their IMD-Blur-enhanced variants in Tab. 1, where *Baseline* and *+IMD-Blur* denote training without and with IMD-Blur augmentation, respectively. As shown in Tab. 1, IMD-Blur yields consistent and substantial improvements across all models and all datasets, achieving average PSNR gains of 0.52 dB on RealDOF, 0.96 dB on RTF, and 0.34 dB on DPDD. Notably, when equipped with IMD-Blur, EVSSM establishes new state-of-the-art results on all three benchmarks, reaching 25.76 dB on RealDOF, 26.13 dB on RTF, and 26.55 dB on DPDD. We also provide qualitative comparisons on RealDOF and DPDD in Fig. 3, where IMD-Blur significantly improves visual sharpness and restoration fidelity over the baselines.

Table 2: Evaluation results on the GoPro, HIDE, RealBlur-J, and RealBlur-R datasets.

Model		GoPro		HIDE		RealBlur-J		RealBlur-R	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MIMO-Unet	Baseline	32.44	0.957	30.00	0.930	31.92	0.919	39.19	0.969
	+IMD-Blur	**33.17 (+0.73)**	**0.963**	**31.01 (+1.01)**	**0.942**	**32.18 (+0.26)**	**0.925**	**39.56 (+0.37)**	**0.971**
Stripformer	Baseline	33.08	0.962	31.03	0.940	32.48	0.929	39.84	0.974
	+IMD-Blur	**33.61 (+0.53)**	**0.966**	**31.49 (+0.46)**	**0.945**	**33.52 (+1.04)**	**0.938**	**40.82 (+0.98)**	**0.977**
EVSSM	Baseline	34.26	0.970	31.91	0.949	34.21	0.942	41.00	0.976
	+IMD-Blur	**34.50 (+0.24)**	**0.971**	**32.14 (+0.23)**	**0.951**	**34.52 (+0.31)**	**0.947**	**41.35 (+0.35)**	**0.977**
Avg Gains		**+0.51**	**+0.004**	**+0.57**	**+0.006**	**+0.54**	**+0.007**	**+0.57**	**+0.002**

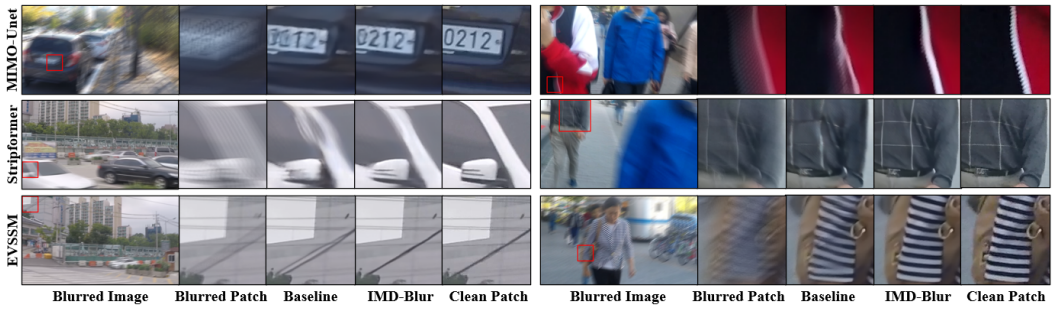


Figure 4: Qualitative results on the GoPro (left) and HIDE (right) datasets.

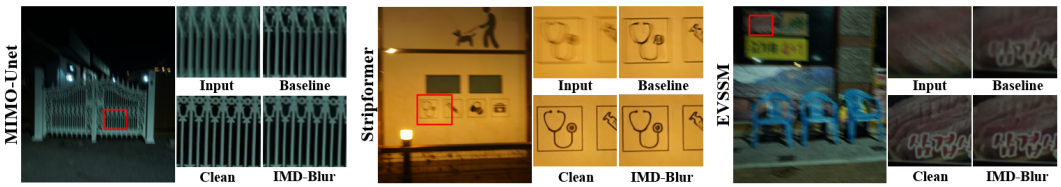


Figure 5: Qualitative results on the RealBlur-J dataset.

Image Motion Deblurring Results. We compare single-image motion deblurring performance of three deblurring models and their IMD-Blur-enhanced variants in Tab. 2, where *Baseline* and *+IMD-Blur* denote training without and with IMD-Blur augmentation, respectively. As shown in Tab. 2, IMD-Blur delivers consistent and notable improvements across all models and all datasets, with average PSNR gains of 0.51 dB on GoPro, 0.57 dB on HIDE, 0.54 dB on RealBlur-J, and 0.57 dB on RealBlur-R. We further provide qualitative comparisons on GoPro, HIDE, and RealBlur-J in Fig. 4 5, showing that IMD-Blur substantially improves visual sharpness and restoration quality over the baselines.

4.3 Ablation Study

We analyze the impact of IMD-Blur on deblurring performance using MIMO-UNet as the backbone. Specifically, we compare IMD-Blur with alternative blur synthesis methods for data augmentation, ablate individual designed components to verify their contributions, examine the effect of parameter-space sampling for augmentation and non-linearity modeling, and assess the effect of the diffusion-based blur generator.

Comparison with Defocus Blur Synthesis Methods. We compare IMD-Blur with representative defocus blur synthesis methods, including PyNet (Ignatov et al., 2020), BokehDiff (Zhu et al., 2025), and Bokehlicious (Seizinger et al., 2025), for blur data augmentation. For each method, we generate the same number of additional training pairs for pre-training. As shown in Tab. 3, IMD-Blur consistently achieves the best performance across all datasets. Unlike prior methods with limited controllability or reliance on synthetic data and auxiliary signals, IMD-Blur learns a controllable, data-driven blur-parameter space from paired sharp/blur images, generating diverse samples that densify the observed blur distribution.

Comparison with Motion Blur Synthesis Methods. We compare IMD-Blur with representative motion blur synthesis methods, including ID-Blau (Wu et al., 2024) and GeoSyn (Kim et al., 2025), for blur data augmentation. For each method, we generate the same number of additional training pairs for pre-training. As shown in Tab. 4, IMD-Blur consistently achieves the best performance.

Table 3: Comparison among different defocus blur synthesis methods.

Method	RealDOF		RTF		DPDD-Indoor		DPDD-Outdoor		DPDD-Combined	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Baseline	24.24	0.770	22.78	0.742	28.07	0.871	22.82	0.749	25.38	0.808
+PyNet	24.23	0.771	23.87	0.768	28.05	0.876	22.88	0.752	25.39	0.812
+BokehDiff	24.43	0.764	24.02	0.772	28.04	0.868	22.92	0.743	25.41	0.804
+Bokehlicious	24.17	0.764	23.98	0.771	28.08	0.873	22.92	0.747	25.43	0.808
+IMD-Blur	**24.73**	**0.787**	**24.22**	**0.802**	**28.51**	**0.884**	**23.33**	**0.763**	**25.85**	**0.822**

Table 4: Comparison among different motion blur synthesis methods.

Method	GoPro		HIDE	
	PSNR	SSIM	PSNR	SSIM
Baseline	32.44	0.957	30.00	0.930
+ID-Blau	32.93	0.961	30.68	0.938
+GeoSyn	33.01	0.962	30.86	0.940
+IMD-Blur	**33.17**	**0.963**	**31.01**	**0.942**

Table 5: Effectiveness of each component on the GoPro dataset.

	PKBP	Blur Parameter	DBG	GoPro	RealDOF
Baseline				32.44	24.24
Net 1	✓			32.43	24.07
Net 2	✓	✓		32.81	24.55
Net 3	✓	✓	✓	**33.17**	**24.73**

Table 6: Effect of sampling strategies for motion blur parameters.

	Magnitude	Orientation	Curvature	PSNR
Setting 1	Random	Random	Random	32.32
Setting 2	DSS	DSS	Random	32.41
Setting 3	Random	DSS	DSS	32.38
Setting 4	DSS	Random	DSS	32.50
Setting 5	DSS	DSS	Zero	32.40
Setting 6	DSS	DSS	DSS	**32.52**

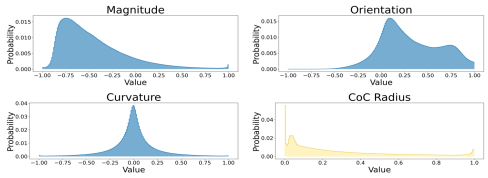


Figure 6: Distribution of blur parameters within GoPro and DPDD training set.

These gains primarily arise from the modeling of controllable nonlinear motion. Compared with prior methods, IMD-Blur synthesizes diverse non-linear motion blur by sampling and perturbing motion parameters, improving deblurring generalization.

Effectiveness of Each Component. We evaluate each component of IMD-Blur through the ablation study in Tab. 5. *Net1* directly synthesizes blurred images from sharp inputs for augmentation, without the blur parameter space or DBG. *Net2* uses our PKBP to synthesize augmented images but excludes DBG, while *Net3* represents the full framework. As shown in Tab. 5, the lack of a controllable parameter space in *Net1* limits sample diversity and deblurring performance. Introducing controllable blur parameters in *Net2* improves accuracy, while adding DBG in *Net3* achieves the best results, demonstrating the complementary benefits of continuous blur parameters and diffusion-based blur generation.

Effect of Sampling Strategy and Non-linearity Modeling. We analyze the effects of data-driven blur-parameter sampling (DSS) and non-linearity modeling in Tab. 6. Following Sec. 3.4, we estimate the distributions of motion magnitude, orientation, and curvature from the GoPro training set (Fig. 6). In Tab. 6, *Random* samples parameters uniformly, whereas *DSS* samples from the estimated distributions. Applying DSS to all three parameters achieves the best performance (*Setting 6*). To evaluate non-linearity modeling, *Setting 5* sets curvature to zero and generates only linear motion blur. Its comparison with *Setting 6* shows that modeling non-linear motion through curvature further improves deblurring performance.

5 Conclusion

In this paper, we presented IMD-Blur (Implicit Motion and Defocus Blur Augmentation), a unified framework for controllable blur augmentation that supports both motion and defocus blur using only paired sharp/blur images, without requiring auxiliary signals. In IMD-Blur, the proposed Parameterized Kernel-based Blurring Pipeline estimates interpretable pixel-wise blur parameters in a continuous space, including curvature for non-linear motion trajectories. Guided by these parameters, the Diffusion-based Blur Generator produces realistic, domain-specific motion- and defocus-blurred images for data augmentation. Extensive experiments on multiple motion and defocus benchmarks have demonstrated that IMD-Blur consistently improves representative deblurring models, achieving state-of-the-art performance across diverse datasets.

REFERENCES

Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *ECCV*, 2020.

Abdullah Abuolaim, Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Learning to reduce defocus blur by realistically modeling dual-pixel data. In *ICCV*, 2021.

Duosheng Chen, Shihao Zhou, Jinshan Pan, Jinglei Shi, Lishen Qu, and Jufeng Yang. A polarization-aided transformer for image deblurring via motion vector decomposition. In *CVPR*, 2025a.

Hang Chen, Yin Xie, Xiaoxiu Peng, Lihu Sun, Wenkai Su, Xiaodong Yang, and Chengming Liu. Quad-pixel image defocus deblurring: A new benchmark and model. In *CVPR*, 2025b.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021.

Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *CVPR*, 2021.

Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *ICCV*, 2021.

Yuning Cui, Wenqi Ren, Xiaochun Cao, and Alois Knoll. Focal network for image restoration. In *ICCV*, 2023.

Yuning Cui, Wenqi Ren, Xiaochun Cao, and Alois Knoll. Revitalizing convolutional network for image restoration. *TPAMI*, 2024.

Laurent D'Andrès, Jordi Salvador, Axel Kochale, and Sabine Süsstrunk. Non-parametric blur map regression for depth of field extension. *TIP*, 2016.

Zhenxuan Fang, Fangfang Wu, Tao Huang, Le Dong, Weisheng Dong, Xin Li, and Guangming Shi. Parameterized blur kernel prior learning for local motion deblurring. In *CVPR*, 2025.

Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *CVPR*, 2019.

Jin-Ting He, Fu-Jen Tsai, Jia-Hao Wu, Yan-Tsung Peng, Chung-Chi Tsai, Chia-Wen Lin, and Yen-Yu Lin. Domain-adaptive video deblurring via test-time blurring. In *ECCV*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

JiaKui Hu, Zhengjian Yao, Lujia Jin, Hangzhou He, and Yanye Lu. Enhancing image restoration transformer via adaptive translation equivariance. In *ICCV*, 2025.

Andrey Ignatov, Jagruti Patel, and Radu Timofte. Rendering natural camera bokeh effect with deep learning. In *CVPRW*, 2020.

Takuhiro Kaneko. Unsupervised learning of depth and depth-of-field effect from natural images with aperture rendering generative adversarial networks. In *CVPR*, 2021.

Insoo Kim, Hana Lee, Hyong-Euk Lee, and Jinwoo Shin. Controllable blur data augmentation using 3d-aware motion estimation. In *ICLR*, 2025.

Lingshun Kong, Jiangxin Dong, Jianjun Ge, Mingqiang Li, and Jinshan Pan. Efficient frequency domain-based transformers for high-quality image deblurring. In *CVPR*, 2023.

Lingshun Kong, Jiangxin Dong, Jinhui Tang, Ming-Hsuan Yang, and Jinshan Pan. Efficient visual state space model for image deblurring. In *CVPR*, 2025.

Pin-Hung Kuo, Jinshan Pan, Shao-Yi Chien, and Ming-Hsuan Yang. Efficient concertormer for image deblurring and beyond. In *ICCV*, 2025.

Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *CVPR*, 2021.

Boyun Li, Haiyu Zhao, Wenxin Wang, Peng Hu, Yuanbiao Gou, and Xi Peng. Mair: A locality-and continuity-preserving mamba for image restoration. In *CVPR*, 2025.

Dasong Li, Yi Zhang, Ka Chun Cheung, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. Learning degradation representations for image deblurring. In *ECCV*, 2022.

Yu-Cheng Lin, Yu-Syuan Xu, Hao-Wei Chen, Hsien-Kai Kuo, and Chun-Yi Lee. Eamamba: Efficient all-around vision state space model for image restoration. In *ICCV*, 2025.

Xintian Mao, Jiansheng Wang, Xingran Xie, Qingli Li, and Yan Wang. Loformer: Local frequency transformer for image deblurring. In *ACM MM*, 2024.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017.

Seungjun Nah, Sanghyun Son, Jaerin Lee, and Kyoung Mu Lee. Clean images are hard to reblur: Exploiting the ill-posed inverse task for dynamic scene deblurring. In *ICLR*, 2022.

Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *ECCV*, 2020.

Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. Bokehme: When neural rendering meets classical rendering. In *CVPR*, 2022a.

Juewen Peng, Jianming Zhang, Xianrui Luo, Hao Lu, Ke Xian, and Zhiguo Cao. Mpib: An mpi-based bokeh rendering framework for realistic partial occlusion effects. In *ECCV*, 2022b.

Juewen Peng, Zhiguo Cao, Xianrui Luo, Ke Xian, Wenfeng Tang, Jianming Zhang, and Guosheng Lin. Bokehme++: Harmonious fusion of classical and neural rendering for versatile bokeh creation. *TPAMI*, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Yuhui Quan, Zicong Wu, and Hui Ji. Neumann network with recursive kernels for single image defocus deblurring. In *CVPR*, 2023.

Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, pp. 184–201, 2020.

Tim Seizinger, Florin-Alexandru Vasluianu, Marcos V Conde, Zongwei Wu, and Radu Timofte. Bokehlicious: Photorealistic bokeh rendering with controllable apertures. In *ICCV*, 2025.

Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, 2019.

Yichen Sheng, Zixun Yu, Lu Ling, Zhiwen Cao, Xuaner Zhang, Xin Lu, Ke Xian, Haiting Lin, and Bedrich Benes. Dr. bokeh: Differentiable occlusion-aware bokeh rendering. In *CVPR*, 2024.

Hyeongseok Son, Junyong Lee, Sunghyun Cho, and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *ICCV*, 2021.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Zhijing Sun, Senyan Xu, Kean Liu, Runze Tian, Xueyang Fu, and Zheng-Jun Zha. Evdm: Event-based real-world video deblurring with mamba. In *ICCV*, 2025.

Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018.

Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, and Chia-Wen Lin. Stripformer: Strip transformer for fast image deblurring. In *ECCV*, 2022.

Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, 2022.

Jia-Hao Wu, Fu-Jen Tsai, Yan-Tsung Peng, Chung-Chi Tsai, Chia-Wen Lin, and Yen-Yu Lin. Id-blau: Image deblurring by implicit diffusion-based reblurring augmentation. In *CVPR*, 2024.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NeurIPS*, 2024.

Yan Yang, Liyuan Pan, Liu Liu, and Miaomiao Liu. K3dn: Disparity-aware kernel estimation for dual-pixel defocus deblurring. In *CVPR*, 2023.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*, 2019.

Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *CVPR*, 2020.

Zhengbo Zhang, Lin Geng Foo, Hossein Rahmani, Jun Liu, and De Wen Soh. Performing defocus deblurring by modeling its formation process. In *ICCV*, 2025.

Chengxuan Zhu, Qingnan Fan, Qi Zhang, Jinwei Chen, Huaqi Zhang, Chao Xu, and Boxin Shi. Bokehdiff: Neural lens blur with one-step diffusion. In *ICCV*, 2025.

A Implementation Details

To construct augmented training data, we employ the proposed **IMD-Blur** together with the **distribution-matched sampling strategy** to synthesize additional blurred images. In practice, we generate a training set five times larger than the original dataset for blur data augmentation.

Motion Blur Augmentation. For motion deblurring, we use the GoPro (Nah et al., 2017) dataset, which contains 2,103 training image pairs. Using the 2,103 sharp training images, IMD-Blur synthesizes 10,515 additional blurred images to enlarge the training set. The synthesized samples are used to pre-train each deblurring model for 500 epochs, following the same protocol of ID-Blau (Wu et al., 2024) for a fair comparison. All other training configurations, including batch size, optimizer, and learning rate, follow the default settings of the corresponding methods.

For blur synthesis, we sample three motion parameters—magnitude, orientation, and curvature—according to the proposed distribution-matched sampling strategy. For simplicity and a fair comparison with ID-Blau (Wu et al., 2024) and GeoSyn (Kim et al., 2025), the parameters are sampled once per image and applied uniformly to all pixels during blur generation. IMD-Blur then synthesizes the corresponding motion-blurred image.

After pre-training, each model is further fine-tuned following its default training protocol. We adopt three representative architectures: MIMO-Unet (Cho et al., 2021), Stripformer (Tsai et al., 2022), and EVSSM (Kong et al., 2025). For the synthetic benchmarks, the models are fine-tuned on the GoPro training set and evaluated on the GoPro and HIDE (Shen et al., 2019) testing sets. Following prior works, we further assess performance on real-world data by fine-tuning the GoPro-pretrained models on the RealBlur (Rim et al., 2020) training set and evaluating them on the corresponding testing set.

Defocus Blur Augmentation. For defocus deblurring, we use the DPDD (Abuolaim & Brown, 2020) dataset, which contains 350 training image pairs. Following previous works, each image is randomly cropped into 512×512 patches, resulting in 7,000 training patches. Using IMD-Blur, we further synthesize 35,000 blurred samples from these patches to augment the training data. The augmented dataset is used to pre-train each model for 150 epochs, while other training configurations follow the default settings of the corresponding methods.

For defocus blur synthesis, the blur parameter is defined by the circle-of-confusion (CoC) radius, which is sampled using the proposed distribution-matched strategy. Similar to the motion blur setting, one CoC radius is sampled per image and applied uniformly to all pixels during blur generation.

After pre-training, the models are fine-tuned according to their default training protocols. We adopt MIMO-Unet (Cho et al., 2021), NRKNet (Quan et al., 2023), and EVSSM (Kong et al., 2025) as the defocus deblurring backbones. The models are fine-tuned on the DPDD training set and evaluated on the DPDD testing set. In addition, following prior works, we directly evaluate the DPDD-pretrained models on the RealDOF (Lee et al., 2021) and RTF (D'Andrès et al., 2016) testing sets to assess their generalization to real-world data.

Hyperparameters. In PKBP, we set the spatially varying kernel size to $k = 61$ for both motion and defocus blur synthesis, providing sufficient support to model large blur patterns while keeping the computation manageable. For the differentiable soft-disk defocus kernel, we set $\alpha = 8$, which offers a practical trade-off between approximating a sharp disk-like PSF and maintaining stable gradients for optimizing the predicted CoC radius. For DBG inference, we adopt DDIM sampling with 20 timesteps to balance generation quality and inference efficiency.

B Effect of Diffusion-based Blur Generator

We evaluate DBG by measuring the PSNR between synthesized and ground-truth blur on the GoPro and RealBlur-J test sets in Tab. 7. DBG consistently improves blur synthesis, with a notably large gain on the real-world RealBlur-J benchmark of nearly 5 dB, indicating that DBG effectively reduces domain-specific artifacts. Qualitative results in Fig. 7 further show that, while kernel-based synthesis preserves blur structure, DBG better matches real blur appearance and mitigates color shifts and saturated light trails induced by the imaging pipeline.

Table 7: Effect of DBG on GoPro and RealBlur-J.

	GoPro	RealBlur-J
w/o DBG	45.06	34.51
w/ DBG	**46.07**	**39.48**



Real Sharp Real Blur w/o DBG w/ DBG

Figure 7: Visual comparison of blurring results with and without DBG on RealBlur-J dataset.

Table 8: Comparison on GoPro under matched training iterations. The training budget of IMD-Blur includes both pre-training and fine-tuning.

Method	197,160	394,321	788,625	1,445,812
Baseline	32.11	32.35	32.44	32.48
IMD-Blur	**32.41**	**32.87**	**32.96**	**33.17**

C Effectiveness under Matched Training Iterations

To determine whether the performance gains arise from IMD-Blur augmentation rather than additional training, we conduct controlled experiments on GoPro using MIMO-UNet+ under matched training budgets. Specifically, we evaluate the baseline and IMD-Blur at 197,160, 394,321, 788,625, and 1,445,812 iterations. For each budget, the baseline and IMD-Blur use the same total number of training iterations, where the IMD-Blur budget includes both pre-training and fine-tuning. As shown in Tab. 8, IMD-Blur consistently outperforms the iteration-matched baseline across all training budgets, yielding PSNR gains of 0.30–0.69 dB. These results confirm that the improvements primarily stem from the proposed IMD-Blur augmentation rather than a longer training schedule.

D Effect of Augmentation Scale

We evaluate augmentation scales from $1\times$ to $6\times$ on GoPro using MIMO-UNet+, where $0\times$ denotes training without augmented data. As shown in Tab. 9, increasing the augmentation scale consistently improves performance up to $5\times$. Further increasing it to $6\times$ yields only a marginal gain of 0.02 dB at additional training cost. We therefore adopt $5\times$ augmentation as a favorable trade-off between deblurring performance and computational cost.

E Effect of Spatially Varying Blur Estimation

This experiment examines whether pixel-wise blur estimation is necessary for learning an accurate blur representation and generating effective augmented data. We construct a *spatially uniform blur augmentation* baseline by restricting PKBP to estimate a single global blur parameter for each image. The estimated parameter is shared across all pixels, resulting in the same blur kernel throughout the image. In contrast, IMD-Blur estimates a pixel-wise parameter field, allowing the blur kernel to vary spatially and better represent non-uniform motion and defocus blur.

For a fair comparison, all methods follow the same augmentation and training protocols. As shown in Tab. 10, spatially uniform augmentation performs worse than no augmentation, reducing PSNR by 0.23 dB on GoPro and 0.27 dB on DPDD. This suggests that a single global parameter cannot accurately model spatially varying blur and may produce ineffective augmented samples. In contrast, IMD-Blur achieves 33.17 dB on GoPro and 25.85 dB on DPDD, outperforming the spatially uniform baseline by 0.96 dB and 0.74 dB, respectively. These results demonstrate the importance of spatially varying blur estimation for both motion and defocus deblurring.

Table 9: Effect of augmentation scale on GoPro using MIMO-UNet+. The $0\times$ setting uses no augmented data.

Scale	$0\times$	$1\times$	$2\times$	$3\times$	$4\times$	$5\times$	$6\times$
PSNR (dB)	32.44	32.85	33.00	33.08	33.12	<u>33.17</u>	**33.19**

Table 10: Comparison with spatially uniform blur augmentation using MIMO-UNet+. Results are reported in PSNR (dB).

Dataset	No Aug.	Spatially Uniform	IMD-Blur
GoPro (motion)	32.44	32.21	**33.17**
DPDD (defocus)	25.38	25.11	**25.85**

F Computational Overhead

Computational overhead of IMD-Blur. We evaluate the computational overhead of the proposed IMD-Blur during inference on images of size 256×256. The run time and parameter count are measured on a single NVIDIA RTX 4090 GPU, and the results are summarized in Tab.11. For the parameterized kernel-based blurring pipeline (PKBP), inference only involves sampling a set of blur parameters, followed by kernel mapping and spatially varying convolution. Therefore, PKBP introduces no parameters. Its run time is 0.037 s for defocus blur and 0.043 s for motion blur. The slight difference comes from the different kernel mapping procedures used for the two blur types. For the Diffusion-based Blur Generator (DBG), both blur types share the same generation network. DBG contains 9.49 M parameters and requires 0.182 s per image during inference. Overall, the total inference time of IMD-Blur is 0.219 s for defocus blur and 0.225 s for motion blur, with a total parameter count of 9.49 M for both settings.

Comparison with different blur synthesis methods. We compare the computational overhead of different blur synthesis methods during inference on 256×256 images using an NVIDIA RTX 4090 GPU. As shown in Tab. 12, BokehDiff (Zhu et al., 2025) has an extremely large number of parameters, reaching 3807M, because it relies on multiple large pretrained models, including Depth Anything V2 (Yang et al., 2024) and an SDXL-based architecture (Podell et al., 2023). In contrast, our IMD-Blur has the same parameter count as ID-Blau (Wu et al., 2024), since both methods adopt the same U-Net architecture as the noise estimator, resulting in a much more lightweight design than other methods. In terms of run time, our method is not the fastest due to the diffusion-based design. Nevertheless, its inference cost remains comparable to prior methods. These results show that IMD-Blur not only provides three key advantages: (i) support for both defocus and motion blur, (ii) requiring only paired training data without additional auxiliary data, and (iii) support for controllable non-linear blur synthesis, but also achieves competitive efficiency in terms of both parameter count and run time. We do not report the results of GeoSyn here because its official source code has not been released.

G Deblurring Results on Real-world Dataset

We provide additional qualitative comparisons on real-world datasets to further evaluate the effectiveness of IMD-Blur. Specifically, we consider four deblurring models, namely MIMO-Unet (Cho et al., 2021), Stripformer (Tsai et al., 2022), NRKNet (Quan et al., 2023), and EVSSM (Kong et al., 2025), and compare their performance with and without our IMD-Blur augmentation. For motion deblurring, qualitative results on the RealBlur-J (Rim et al., 2020) dataset are shown in Fig. 8 9 10. For defocus deblurring, qualitative results on the RealDOF (Lee et al., 2021) dataset are shown in Fig. 11 12 13. As can be observed, models trained with IMD-Blur consistently produce clearer restored images with sharper structures and visual appearances closer to the ground-truth sharp images than the corresponding baselines. These results further validate the effectiveness of our method in real-world scenarios.

Table 11: Computational overhead of the proposed IMD-Blur measured during inference on 256×256 images using an NVIDIA RTX 4090 GPU. PKBP denotes the parameterized kernel-based blurring pipeline, which only involves kernel mapping and convolution operations.

	PKBP		DBG		Total	
Blur Type	**Run time (s)**	**Params (M)**	**Run time (s)**	**Params (M)**	**Run time (s)**	**Params (M)**
Defocus Blur	0.037	0.00	0.182	9.49	0.219	9.49
Motion Blur	0.043	0.00	0.182	9.49	0.225	9.49

Table 12: Comparison of the computational overhead of different blur synthesis methods measured during inference on 256×256 images using an NVIDIA RTX 4090 GPU.

Metric	**BokehDiff** (Zhu et al., 2025)	**Bokehlicious** (Seizinger et al., 2025)	**ID-Blau** (Wu et al., 2024)	**IMD-Blur (Ours)**
Run time (s)	0.424	0.076	0.182	0.225
Params (M)	3807.65	13.95	9.49	9.49

H Controllable Motion and Defocus Blurring Results

We provide additional qualitative results on the GoPro, RealBlur-J, and DPDD datasets to demonstrate the controllability of IMD-Blur in synthesizing both motion and defocus blur patterns.

Controllable Motion Blur. For motion blur, the blur pattern is governed by three parameters: magnitude, orientation, and curvature. By explicitly manipulating these parameters, IMD-Blur can generate diverse motion blur patterns, as illustrated in Fig. 14 15. Specifically, given a sharp image S and a predefined motion parameter set $\mathbf{P}_m^i$, IMD-Blur synthesizes the corresponding blurred image as $B = \text{IMD-Blur}(S, \mathbf{P}_m^i)$.

From Fig. 14 15, three groups of parameter variations are presented. For $i = 1$ to 4, the orientation and curvature are fixed while the magnitude is gradually increased, resulting in progressively stronger motion blur. For $i = 5$ to 9, the magnitude and curvature are fixed while the orientation is gradually rotated, producing motion blur with different directions. For $i = 10$ to 14, the magnitude and orientation are fixed while the curvature is gradually adjusted, generating non-linear motion trajectories. These results demonstrate that IMD-Blur enables precise and interpretable control over motion blur patterns.

Controllable Defocus Blur. For defocus blur, the blur pattern is controlled by the circle-of-confusion (CoC) radius. To better illustrate the pixel-wise controllability of IMD-Blur, we further incorporate a pretrained depth estimator (Yang et al., 2024) to obtain a depth map and modulate the blur parameters accordingly, as shown in Fig. 16. Given a sharp image S and the corresponding defocus parameter field $\mathbf{P}_d^i$, the synthesized blurred image is obtained as $B = \text{IMD-Blur}(S, \mathbf{P}_d^i)$.

From Fig. 16, three representative settings are presented. For $i = 1$ to 4, a uniform CoC radius is applied to the entire image and gradually increased, resulting in globally stronger defocus blur. For $i = 5$ to 9, the blur radius is modulated based on the estimated depth map, producing spatially varying blur effects. For $i = 10$ to 14, we additionally specify different focus depths to shift the focal plane, generating blur patterns corresponding to different focus positions. These results demonstrate that IMD-Blur can effectively generate controllable defocus blur patterns and, when combined with depth information, produce more diverse and realistic blur effects.

I Limitation

Although our experiments demonstrate the effectiveness of IMD-Blur across diverse blur domains, future work could further extend its motion parameterization to capture more complex real-world trajectories, such as large-scale camera shake, direction reversals, and direction-sensitive blur induced by rolling-shutter effects or asymmetric exposure. In addition, extending this implicit parameterization to other low-level vision tasks, such as deraining, desnowing, and dehazing, requires further investigation into task-specific degradation spaces.

J AI USE STATEMENT

In this work, we used generative AI tools solely for language editing, including grammar correction, sentence restructuring, and improving the clarity of presentation. Generative AI was not used to develop research ideas or methods, conduct experiments, generate or analyze results, or identify citations. All AI-assisted revisions were carefully reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

K REPRODUCIBILITY STATEMENT

We provide the methodological, architectural, implementation, training, and evaluation details necessary to reproduce our results in the main paper and appendix. These include the model configurations, optimization settings, data-processing procedures, augmentation protocols, and experimental setups. To further support reproducibility, we commit to publicly releasing the complete source code, pretrained models, and configuration files immediately upon acceptance.

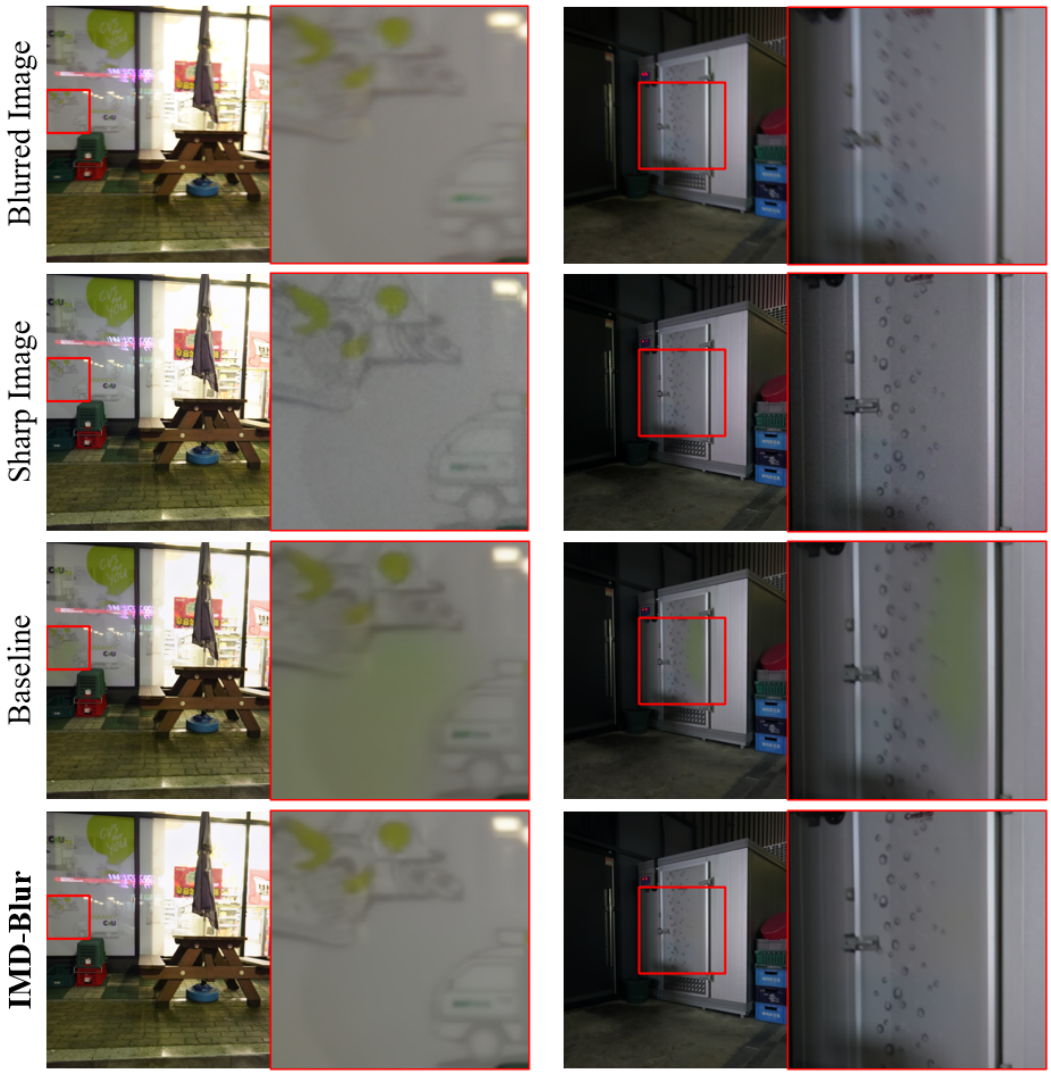


Figure 8: Qualitative results of MIMO-Unet (Cho et al., 2021) on the RealBlur-J (Rim et al., 2020) dataset.

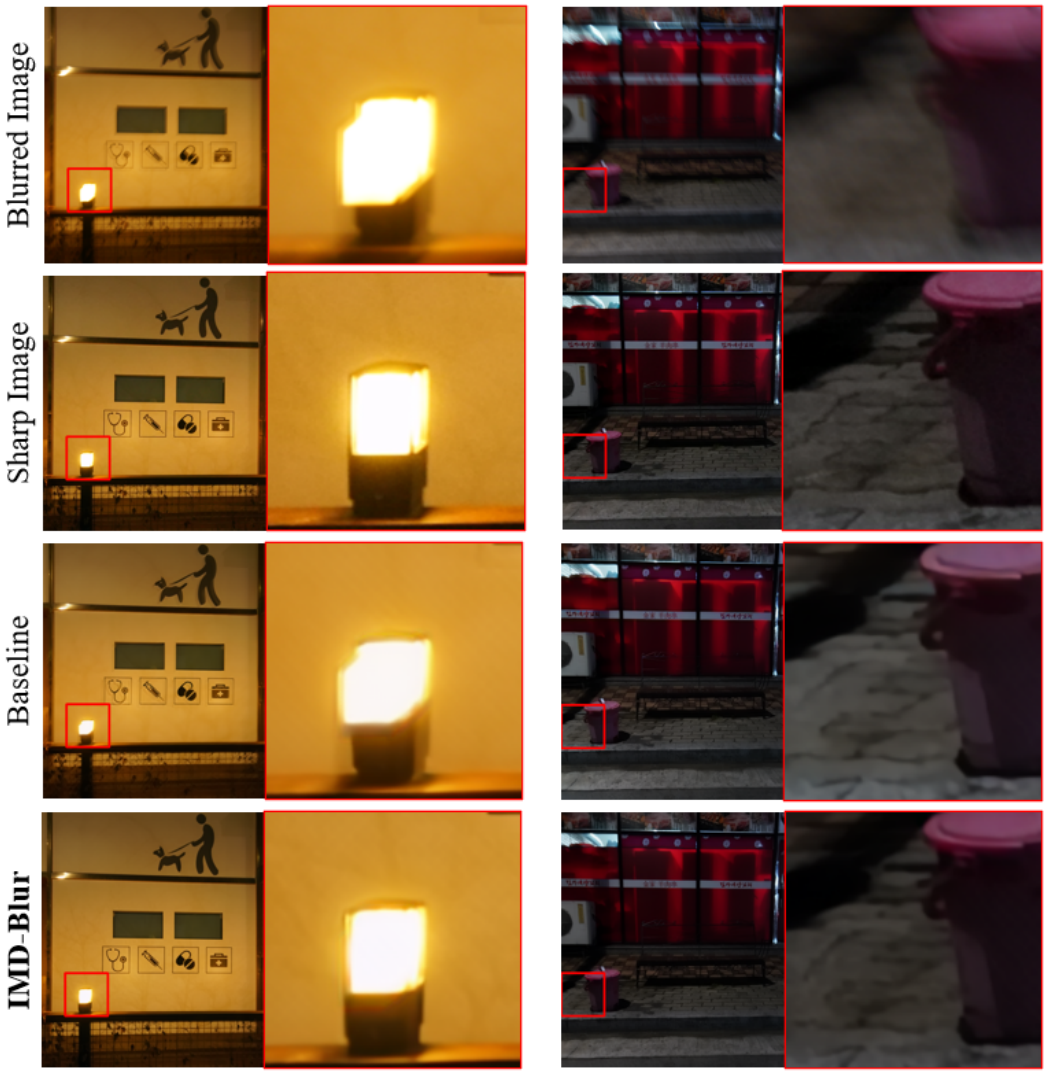


Figure 9: Qualitative results of Stripformer (Tsai et al., 2022) on the RealBlur-J (Rim et al., 2020) dataset.

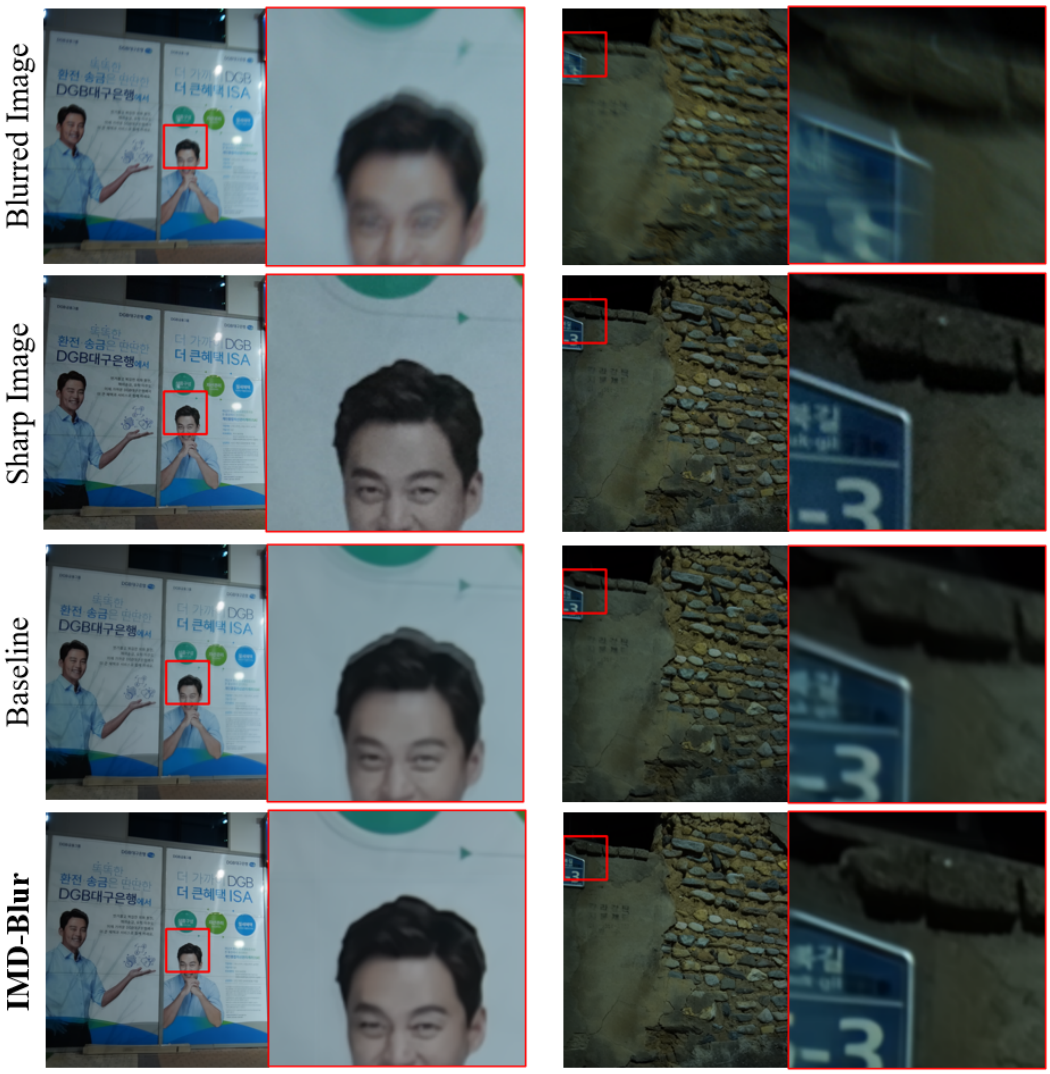


Figure 10: Qualitative results of EVSSM (Kong et al., 2025) on the RealBlur-J (Rim et al., 2020) dataset.

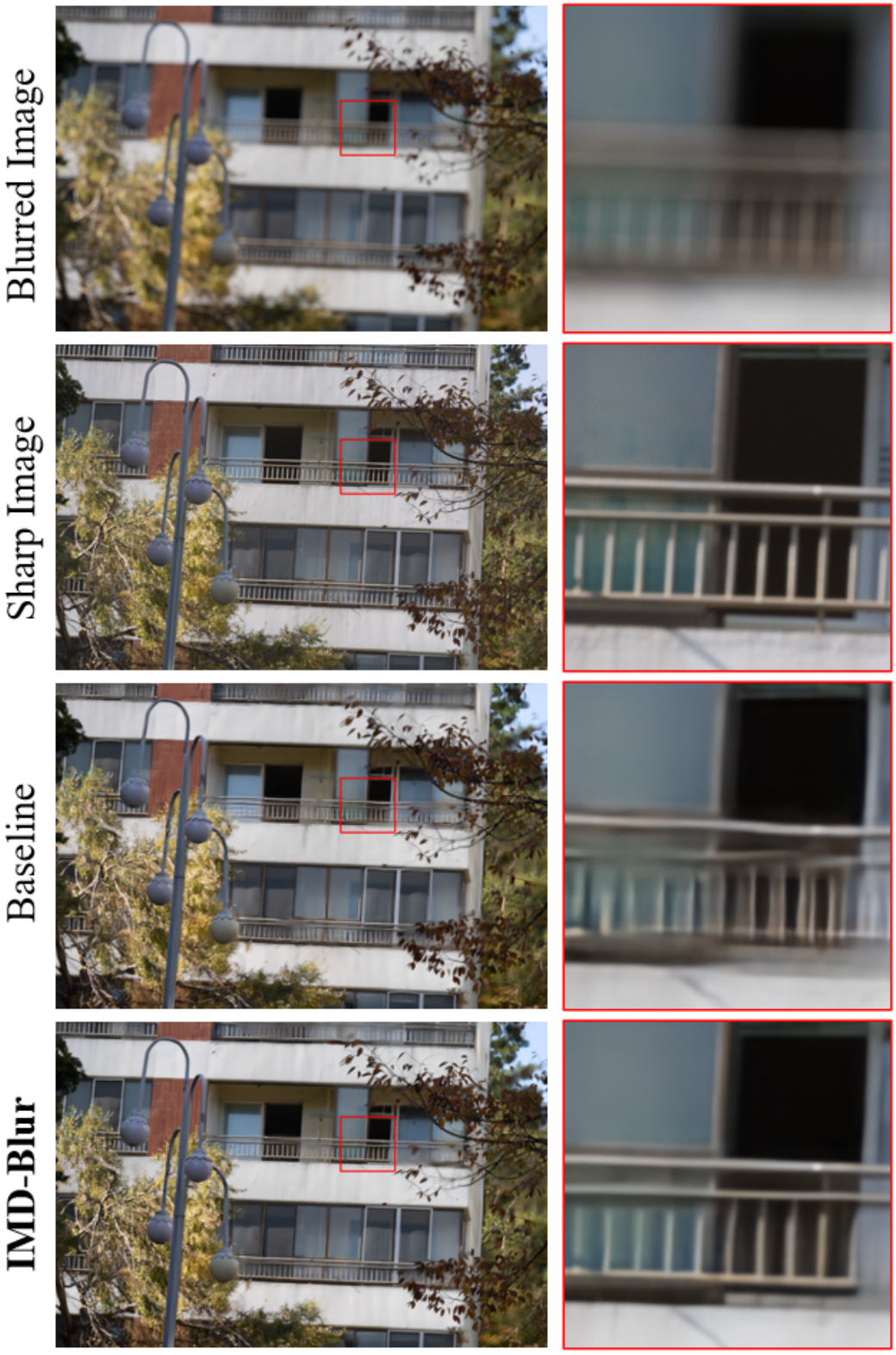


Figure 11: Qualitative results of MIMO-Unet (Cho et al., 2021) on the RealDOF (Lee et al., 2021) dataset.

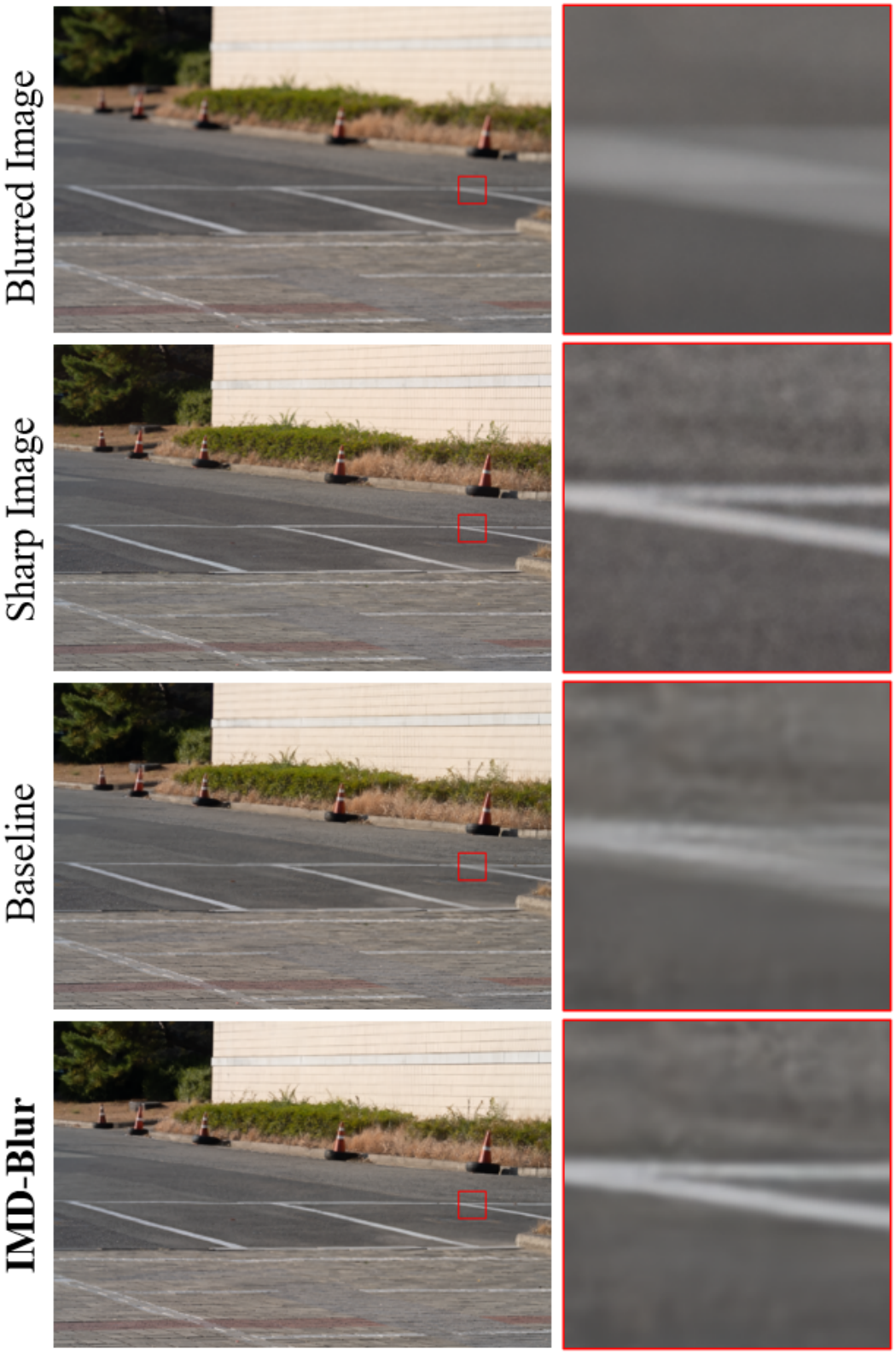


Figure 12: Qualitative results of NRKNet (Quan et al., 2023) on the RealDOF (Lee et al., 2021) dataset.



Figure 13: Qualitative results of EVSSM (Kong et al., 2025) on the RealDOF (Lee et al., 2021) dataset.

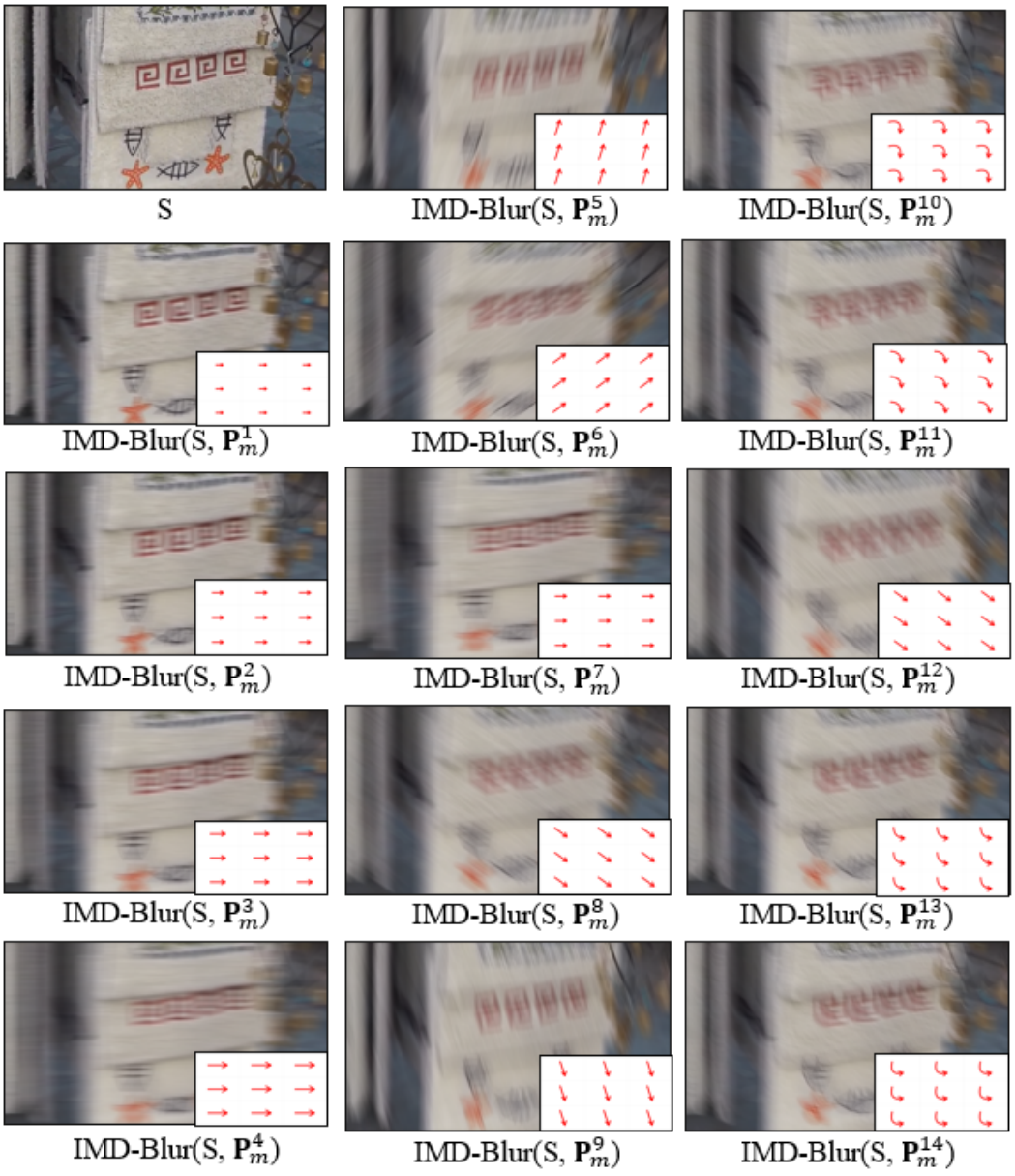


Figure 14: Qualitative blurred results of IMD-Blur on the GoPro dataset

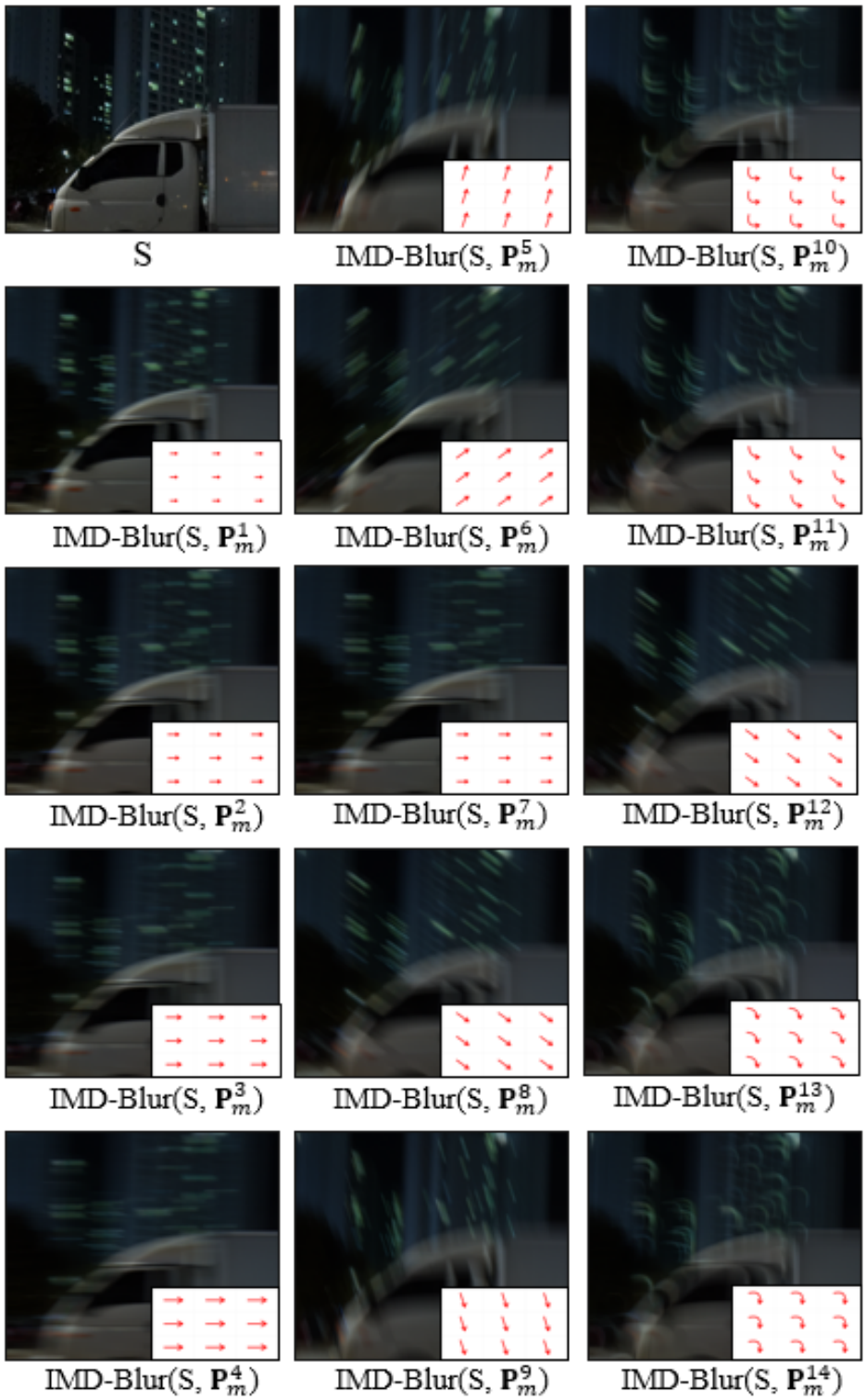


Figure 15: Qualitative blurred results of IMD-Blur on the RealBlur dataset

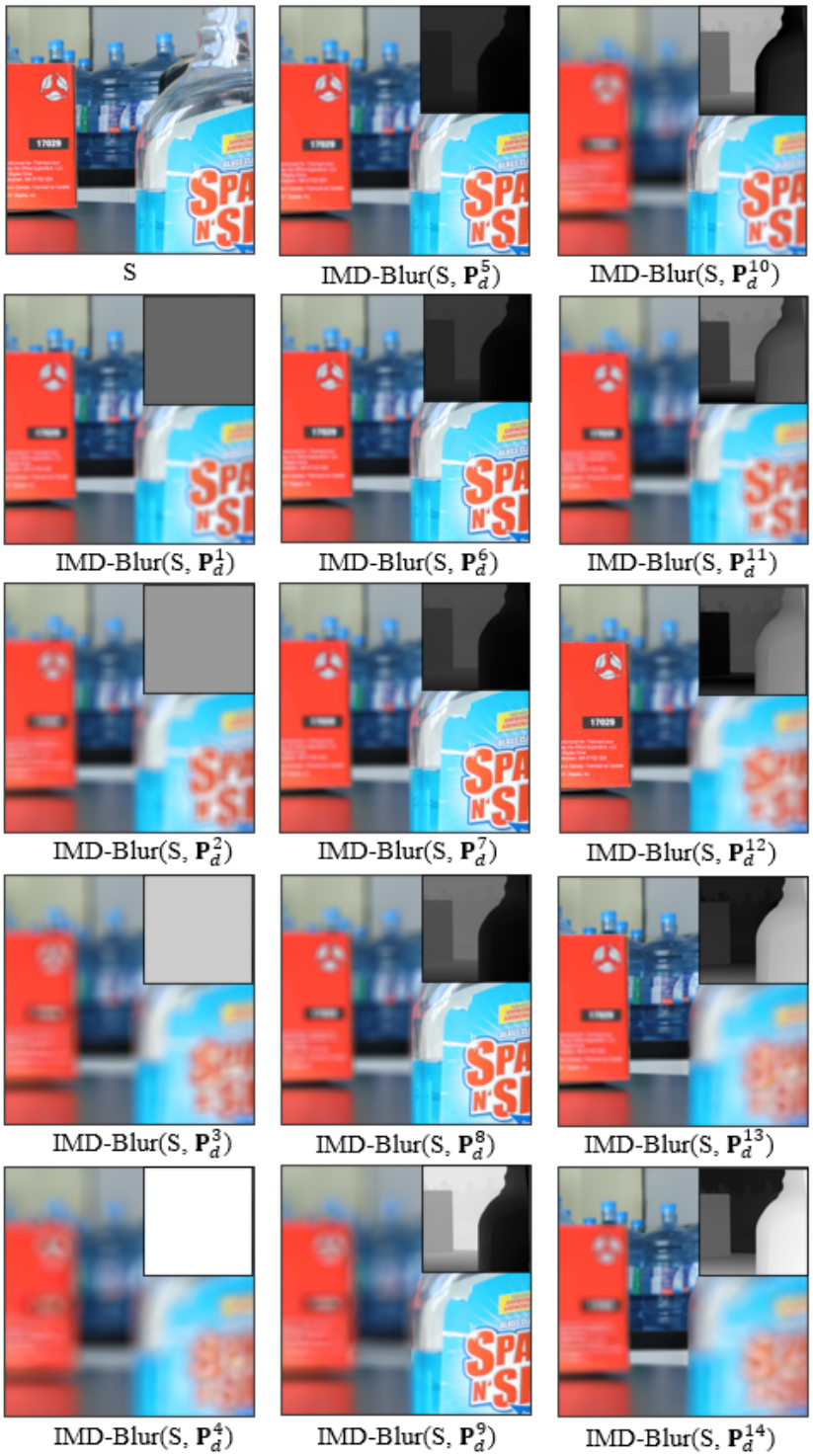


Figure 16: Qualitative blurred results of IMD-Blur on the DPDD dataset