OMoBlur: An Object Motion Blur Dataset and Benchmark for Real-World Local Motion Deblurring

Supplementary Material

A. Image Acquisition and Dataset Construction

A.1. System Calibration

Since the defective pixels and black level have been corrected in the camera, we only need to test the sensor's response linearity and calibrate the remaining parameters in the ISP for our dataset construction.

Linear Response Test Before data acquisition, we characterize the sensor's linear operating region. Across scenes, we adjust the aperture and gain so that the irradiance of moving subjects remains within the linear range for all Bayer channels. Specifically, we capture an FL Pattern BOX (a standard 5100 K transmissive lightbox) in a dark room. For each illumination level, we increase the exposure in 100 μs steps starting from 100 μs until the mean value in any two Bayer channels reaches saturation. We plot the per-channel means for illumination levels 1, 5, and 10 in Fig. 3.

Lens Shading Correction A region with the most uniform illumination is selected at the center of the lightbox as the field of view, and the image is slightly defocused to achieve nearly ideal, perfectly uniform lighting. The corresponding LSC parameters are then calibrated at different aperture settings. Specifically, for each aperture, multiple images are captured within the linear response range, and their averages are computed to reduce noise, then a gain map for LSC is fitted for each RGB channel. The impact difference of the largest aperture (F2.4) and the smallest aperture (F16) on the gain requirement is found to be less than 1%. For simplicity, the average gain map is used as the ISP's LSC gain map. Fig. 1 shows the images before and after correction along with the per-channel histograms of the RAW files.

White Balance and Color Correction We calibrate the color correction matrix (CCM) under the correct white balance (WB) and save it for use by the ISP. For WB, we perform scene-specific calibration using a standard diffuse white target.

A.2. Hardware Configuration and Capture Setting

Our imaging setup comprises a BASLER a2A1920-160ucBAS camera equipped with a Sony IMX392 CMOS sensor (pixel size: $3.45\mu m \times 3.45\mu m$), and a Basler C23-0824-5M lens with an 8 mm focal length.

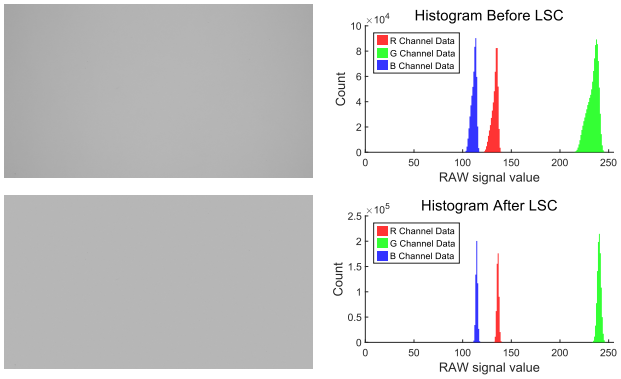

Figure 1. Comparison of an image and its histogram before and after Lens Shading Correction (LSC) under uniform lighting.

Capture-side Optimization for Residue Exposure Gap According to the imaging-geometry relationship[1], the relation among sensor pixel size c, blurry pixel count n (representing the relative displacement between the image and sensor), object distance d, image distance l', object speed v and exposure time Δt can be expressed as:

$$\frac{v\Delta t}{nc} = \frac{l'}{d} \tag{1}$$

In our configuration, we have $l' = 8mm$, $c = 3.45\mu m$, exposure time $\Delta t = 1960\mu s$, and exposure gap $\delta t = 40\mu s$. When $\frac{d}{v} > 0.927s$, the image displacement during the exposure gap is limited to less than 0.1 pixel ($n < 0.1$), and the displacement during exposure time remains within 4.9 pixel.

In practical scenarios, the typical moving speed of running individuals or vehicles on campus does not exceed $5.5m/s$. Maintaining an object distance of around $5m$ satisfies the condition of $\frac{d}{v} > 0.927s$. For pedestrians walking at approximately $1m/s$, a closer distance of about $1m$ can be adopted to capture finer details.

Under these settings, our system achieves quasi-continuous frame acquisition and provides effectively sharp ground-truth images.

Data Balance for Aberration To reduce the absolute readout time, we limit the Region of Interest (ROI) to

1920×360. To enhance the network's robustness to various combinations of aberration and object motion-induced image degradation, we shift the ROI vertically across different scenes, thus increasing the field of view. Specifically, we focus on the central 1920×1080 area of the CMOS sensor, allocating more captures from the center to the sides (Fig. 2), as the central region typically contains more critical information.

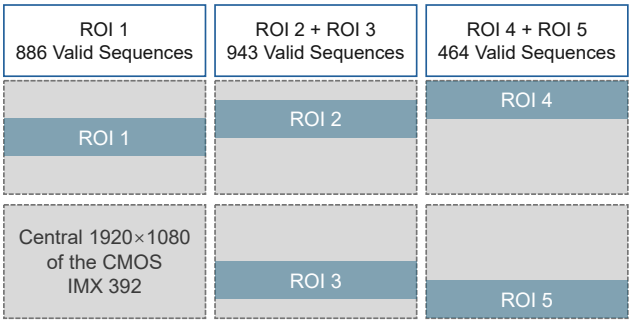

Figure 2. The illustration of capture ROI, with valid sequence count above.

A.3. Dataset Construction

Blur-Sharp Pairs In total, we record over 2600 sequences, each containing 50 frames. We use the 8th, 24th, and 40th frame as ground truth middle frames, and average 7, 11, and 15 consecutive frames around each to generate blurred images. For ISP, we apply pre-calibrated LSC, WB, CCM, with Menon's demosaic method[2] and simple gamma correction ($\gamma = 2.2$).

We manually discard the following cases: (i) frames with strong flickering light sources on moving objects, (ii) scenes with a single moving object whose displacement is excessively large (blur >100 pixels or $> 2\times$ the object's size), and (iii) trajectories with abrupt changes (e.g., a ping-pong ball bouncing off a racket). After filtering, over 20,000 blur-sharp pairs remain.

Mask Generator We generate the motion mask in two stages to suppress noise and retain temporally consistent motion. First, for each optical flow field we cluster pixel vectors $[u, v]$ to separate motion patterns and then apply percentile based IQR gating on the magnitude $f = \sqrt{u^2 + v^2}$ to remove low energy background responses and within cluster outliers, producing a per frame support map while accumulating raw vectors. Next, we temporally fuse the flows across blur window and normalize by the accumulated support to obtain $\hat{f} = \sqrt{A_u^2 + A_v^2} / \max(T, 1)$, which emphasizes motions that are strong and repeatedly observed while down weighting sporadic artifacts. We then perform sequence level clustering on the accumulated vectors and apply a second IQR gate on $\hat{f}$ to refine coherent regions and eliminate residual outliers. Finally, a light morphological closing fills small holes and smooths boundaries to yield the binary mask. See Algorithm 1 for details.

Algorithm 1: MeFlow-based Motion Region Mask Generation

Input: Optical-flow model $\mathcal{F}$; frames $\{\mathbf{I}_t\}$; window lengths $L = \{7, 11, 15\}$; clusters K; radius r
Output: Binary motion mask $\mathbf{M}^{(l)}$ for each $l \in L$

foreach *adjacent pair* $(\mathbf{I}_t, \mathbf{I}_{t+1})$ **do**
  Denoise both images (median $\rightarrow$ colored NL-means)
  Compute flow $\mathbf{V}_t = \mathcal{F}(\mathbf{I}_t, \mathbf{I}_{t+1})$ with padding; store $[u_t, v_t]$
foreach $l \in L$ **do**
  Take central sub-sequence $\mathcal{V}_l$ of length l
  $\mathbf{A} \leftarrow \mathbf{0}$, $\mathbf{T} \leftarrow \mathbf{0}$ ▷ accumulated vector and weight
  ▷ Stage 1: per-frame clustering + IQR gating
  foreach $\mathbf{V} = [u, v] \in \mathcal{V}_l$ **do**
    $\mathbf{f} \leftarrow \sqrt{u^2 + v^2}$; $\mathbf{C} \leftarrow K$-means on $[u, v]$
    for $i \leftarrow 1$ **to** K **do**
      Compute percentiles q_{10}, q_{33}, q_{80} of $\mathbf{f}$ in cluster i
      if $q_{33} \geq 0.5$ **then**
        $IQR \leftarrow q_{80} - q_{10}$; $[L, U] \leftarrow [q_{10} - 0.8IQR,\ q_{80} + 0.8IQR]$
        $\mathbf{M}_i \leftarrow \mathbb{1}\{\mathbf{f} \in [L, U]\}$ with overlap to cluster > 0.8
        $\mathbf{T} \leftarrow \mathbf{T} + \mathbf{M}_i$
    $\mathbf{A} \leftarrow \mathbf{A} + \mathbf{V}$
  $\hat{\mathbf{f}} \leftarrow \sqrt{A_u^2 + A_v^2} \,/\, \max(\mathbf{T}, 1)$ ▷ safe pixelwise normalization
  ▷ Stage 2: temporal fusion clustering + IQR re-gating
  $\mathbf{C}' \leftarrow K$-means on accumulated $[A_u, A_v]$; $\mathbf{M}^{(l)} \leftarrow \mathbf{0}$
  for $i \leftarrow 1$ **to** K **do**
    Compute q_{25}, q_{33}, q_{75} of $\hat{\mathbf{f}}$ in cluster i
    if $q_{33} \geq 0.5$ **then**
      $IQR \leftarrow q_{75} - q_{25}$; $[L, U] \leftarrow [q_{25} - 1.5IQR,\ q_{75} + 1.5IQR]$
      $\mathbf{N}_i \leftarrow \mathbb{1}\{\hat{\mathbf{f}} \in [L, U]\}$ with overlap to cluster > 0.8
      $\mathbf{M}^{(l)} \leftarrow \mathbf{M}^{(l)} + \mathbf{N}_i$
  $\mathbf{M}^{(l)} \leftarrow \text{binary-closing}\big(\mathbf{M}^{(l)}, \text{disk}(r)\big)$

Blur Region statistics To characterize our dataset, we measure the distribution of blur region proportion per image. Because masks created during dataset construction may contain inaccuracies, we also report the distribution derived from the model's predicted gate map. Concretely,

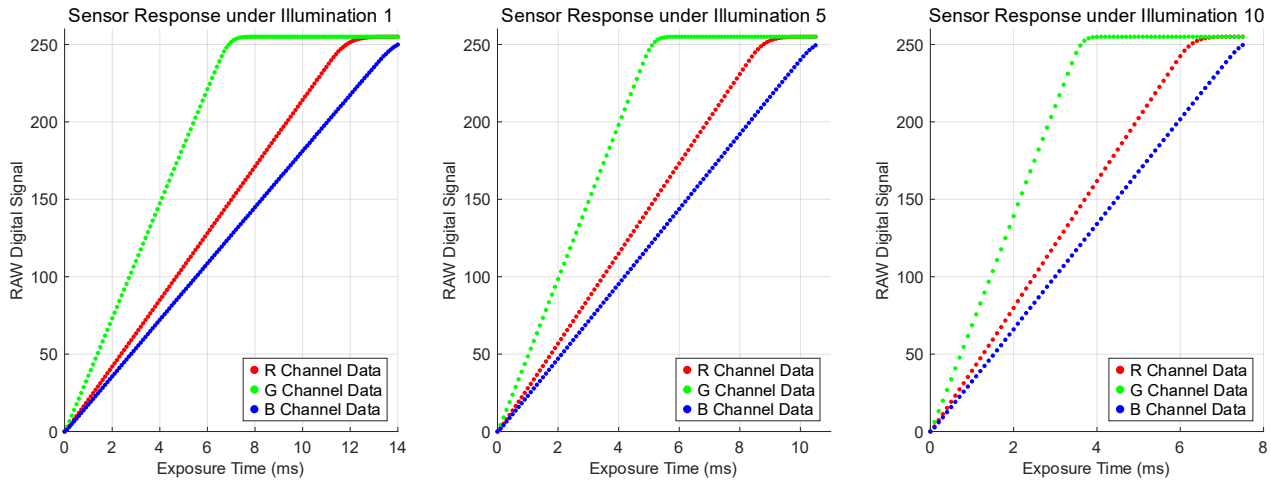

Figure 3. The linear response range test.

the binary mask marks blur pixels, and the blur proportion is computed as the ratio of blur pixels to all pixels. The gate map encodes blur confidence with continuous values in [0,1]. We take its spatial mean over the image as the blur proportion. The resulting histograms are shown in Fig. 4, which summarize the dataset's distribution. The averages are closely matched: 35.61% when computed from masks and 34.76% from gate maps, suggesting reasonable agreement despite annotation noise. In light of this distribution, we refrain from random cropping during training and instead adopt a Blur-Aware Patch Cropping Strategy (Sec. B.3) that prioritizes patches with higher blur content to strengthen the network's attention toward blurred regions.

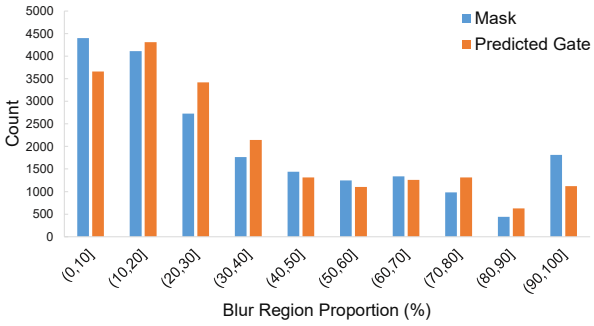

Figure 4. Distribution of Blur Region Proportions. X-axis: blur-region proportion (%) per image; Y-axis: image count.

B. OMDNet Architecture and Training

B.1. Diff-TAM

The Differential Transformer enhances attention to relevant context while suppressing noise, thereby promoting sparse attention patterns [4]. It has been shown to outperform traditional Transformers in language modeling and scaling, and demonstrates strong performance in practical tasks such as long-context modeling, hallucination mitigation, and in-context learning, improving both accuracy and robustness to input order permutations. Given its properties, we find it particularly suitable for enhancing motion representations when combined with the transposed attention mechanism from Restormer [5]. Accordingly, we propose the Differential Transposed Attention Module (Diff-TAM) as the motion feature extractor in our network.

Given a layer normalized tensor $X \in \mathbb{R}^{\hat{H}\times\hat{W}\times\hat{C}}$, we first project it into query, key, and value representations $Q_1, Q_2, K_1, K_2 \in \mathbb{R}^{\hat{H}\times\hat{W}\times\hat{C}}$, $V \in \mathbb{R}^{\hat{H}\times\hat{W}\times 2\hat{C}}$ by applying 1×1 convolutions followed by 3×3 depth-wise convolutions to encode both channel and spatial context. We then treat the channel dimension, which corresponds to different feature types, as tokens for the attention operation. The query, key, and value tensors are reshaped into $\hat{Q}_1, \hat{Q}_2, \hat{K}_1, \hat{K}_2 \in \mathbb{R}^{\hat{C}\times\hat{H}\hat{W}}$, $\hat{V} \in \mathbb{R}^{\hat{H}\hat{W}\times 2\hat{C}}$. The differential transposed attention map $\mathrm{A} \in \mathbb{R}^{\hat{C}\times\hat{C}}$ is computed as follows:

$$\begin{aligned}[Q_1; Q_2] &= W_d^Q W_p^Q X \\ [K_1; K_2] &= W_d^K W_p^K X \\ [V] &= W_d^V W_p^V X\end{aligned} \tag{2}$$

$$\mathrm{A} = softmax(\frac{\hat{Q}_1\hat{K}_1^T}{\alpha_1}) - \lambda softmax(\frac{\hat{Q}_2\hat{K}_2^T}{\alpha_2}) \tag{3}$$

where $W_p^{(\cdot)}$ denotes a 1×1 point-wise convolution, $W_d^{(\cdot)}$ a 3×3 depth-wise convolution, and α_1, α_2 and λ are learnable scaling parameters. Following[4], we re-parameterize λ as:

$$\lambda = \exp(\lambda_{q1}\cdot\lambda_{k1}) - \exp(\lambda_{q2}\cdot\lambda_{k2}) + \lambda_{\text{init}} \tag{4}$$

where $\lambda_{q1}, \lambda_{k1}, \lambda_{q2}, \lambda_{k2}$ are learnable vectors and λ_{init} is a hyperparameter.

To maintain symmetry and meet the computational requirements of differential transposed attention, we split $\hat{V}$ into $\hat{V}_1, \hat{V}_2 \in \mathbb{R}^{\hat{H}\hat{W}\times\hat{C}}$, and compute the output feature $\hat{X}$ as:

$$\hat{X} = W_p\text{Concat}(\text{A}\cdot\hat{V}_1, \text{A}\cdot\hat{V}_2) + X \tag{5}$$

In practice, we employ the multi-head mechanism [3], using head-wise RMS normalization to account for the greater diversity of statistical information across heads in Diff-TAM.

B.2. Overall Model Architecture and Hyperparameters

As introduced in Section 4 of our main paper, OMDNet is a U-shaped network in which the standard skip connections are replaced by a Motion–Appearance Extract Block (MAEB) followed by a Flow-guided Gate Predictor (FGP). The model adopts a single-input, multi-output (SIMO) configuration.

The object-motion-blurred input image $B_1 \in \mathbb{R}^{H\times W\times 3}$ is first processed by a Base Block composed of a 3×3 convolution, a ReLU, and 20 residual blocks. Each residual block contains two convolutions (the first followed by ReLU and the second without), with a residual connection. With a stride-2 downsampling layer between levels, the feature at level i is

$$X_i \in \mathbb{R}^{\frac{H}{2^{i-1}}\times\frac{W}{2^{i-1}}\times(2^{i-1}C)}, \quad i\in\{1,2,3\},$$

where $C = 32$ in our experiments.

Each X_i is fed into an MAEB to produce a motion feature mf_i and an appearance feature af_i. In MAEB we set $\lambda_{\text{init}} = 0.8$; the numbers of attention heads at levels $i = 1, 2, 3$ are 2, 4, and 8, respectively. The motion feature mf_i is passed to the FGP to predict a gate $\hat{G}_i$, while the appearance feature af_i is concatenated with the upsampled feature from the lower level decoder-side Base Block and then fed into the corresponding higher-level Base Block.

Decoder upsampling between Base Blocks uses a 2×2 transposed convolution with stride 2, which halves the channel dimension and doubles the spatial resolution. Each decoder-side Base Block outputs a residual image

$$R_i \in \mathbb{R}^{\frac{H}{2^{i-1}}\times\frac{W}{2^{i-1}}\times 3}, \quad i\in\{1,2,3\},$$

and we obtain the restored image $\hat{S}_i$ before gating by

$$\hat{S}_i = B_i + R_i,$$

where B_i is the 2^{i-1}-downsampled version of B_1. Each $\hat{S}_i$ is further fused with B_i via an adaptive gated fusion mechanism. During training, the fusion uses the predicted gate $\hat{G}_i$, a precomputed mask M_i, and the sharp ground-truth S_i to produce the gated restoration $\hat{S}_{iG}$ and its enhanced variant $\hat{S}^*_{iG}$. At test time, only $\hat{G}_i$ is required to obtain the final gated restoration $\hat{S}_{iG}$.

B.3. Blur-Aware Patch Cropping Strategy

The original BAPC[1] mechanism selects random anchors from blur regions but often yields patches with insufficient blur content due to its boundary constraints and random positioning.

Our enhanced approach introduces forced blur region selection with 70% probability during training. When activated, we iteratively search for patch locations that achieve at least 30% blur coverage, removing the boundary margin restriction. This ensures each selected patch contains substantial blur content while maintaining training diversity through the probabilistic activation.

C. More experimental results

C.1. Detailed Quantitative Evaluation of Deblurring Methods

To provide a more fine-grained evaluation of deblurring performance, we test all models on three OMoBlur subsets (easy / medium / hard), corresponding to 7, 11 and 15 frames used for blur generation.

Table 1. Detailed quantitative evaluations on the OMoBlur dataset. Each deblurring method is evaluated on three subsets: the first row is subset-easy (blur averaged from 7 frames), the second row is subset-medium (11 frames), and the third row is subset-hard (15 frames).

Methods	PSNR↑	SSIM↑	PSNR$_w$↑	SSIM$_w$↑	LPIPS↓	DISTS↓
MIMO-UNet	36.17	0.8898	34.58	0.8752	0.2108	0.0791
	35.34	0.8794	33.10	0.8530	0.2325	0.0909
	34.57	0.8735	31.74	0.8321	0.2474	0.0999
Restormer	36.42	0.8951	34.80	0.8816	0.2182	0.0858
	35.55	0.8857	33.21	0.8596	0.2403	0.0973
	34.67	0.8809	31.70	0.8390	0.2556	0.1062
NAFNet	36.41	0.8952	34.76	0.8815	0.2172	0.0855
	35.50	0.8854	33.11	0.8587	0.2400	0.0984
	34.63	0.8806	31.63	0.8381	0.2555	0.1075
EVSSM	36.38	0.8936	34.74	0.8793	0.2226	0.0879
	35.52	0.8837	33.13	0.8557	0.2444	0.0996
	34.71	0.8789	31.73	0.8351	0.2591	0.1087
LBAG	36.40	0.8955	34.75	0.8821	0.2175	0.0846
	35.56	0.8862	33.21	0.8602	0.2381	0.0955
	34.68	0.8811	31.70	0.8386	0.2527	0.1042
LMD-ViT	**36.49**	**0.8967**	35.07	0.8855	0.2157	0.0843
	35.69	0.8875	33.51	0.8644	0.2380	0.0959
	34.87	0.8828	32.03	0.8446	0.2531	0.1044
OMDNet (Ours)	36.43	0.8959	**35.38**	**0.8892**	**0.1979**	**0.0680**
	35.72	**0.8881**	**34.01**	**0.8712**	**0.2197**	**0.0801**
	35.04	**0.8831**	**32.72**	**0.8543**	**0.2341**	**0.0885**

As Tab. 1 shows, OMDNet exhibits the smallest degradation in weighted pixel fidelity as difficulty increases: its

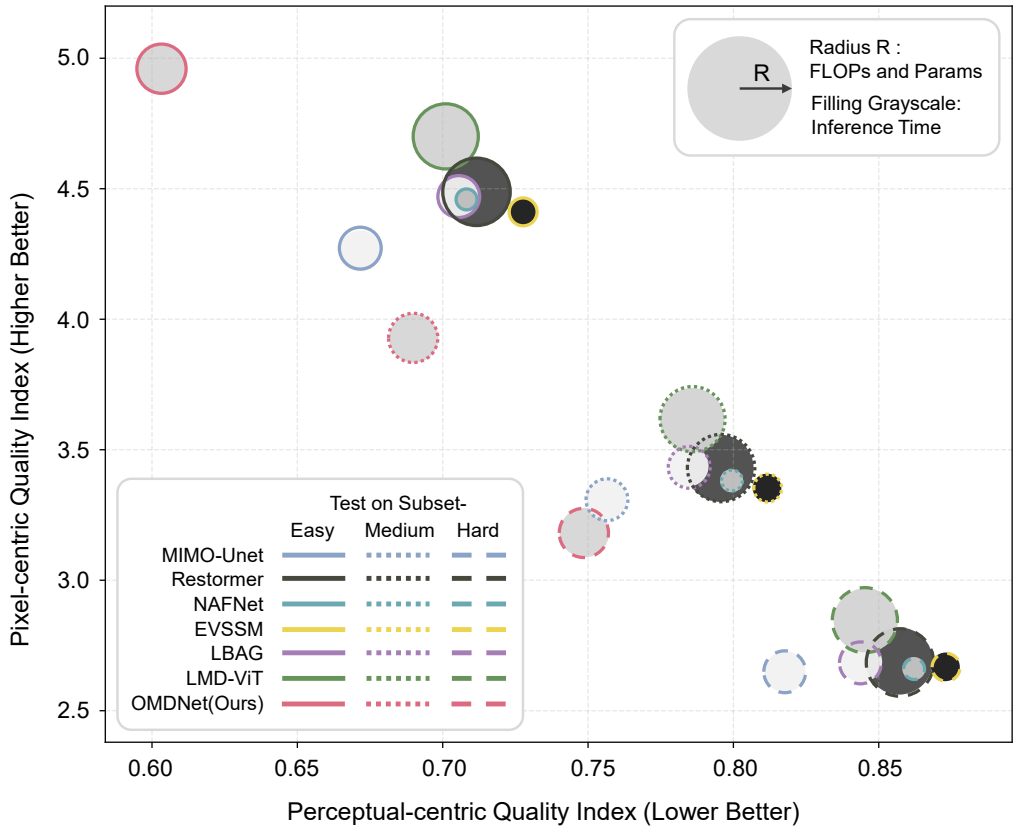

Figure 5. Efficiency vs. Performance Trade-off. Circle size denotes computational cost (FLOPs/Params) and grayscale intensity represents inference time. OMDNet achieves the best balance between restoration quality and efficiency.

PSNR_w drops from 35.38 dB (easy) to 32.72 dB (hard), a decline of 2.66 dB, while other competitors suffer an average PSNR_w decline of over 3 dB (e.g., Restormer $\approx$ 3.10 dB, NAFNet $\approx$ 3.13 dB, LMD-ViT $\approx$ 3.04 dB). This smaller drop indicates stronger robustness to increased motion complexity and longer temporal blur aggregation.

Concurrently, OMDNet attains the best perceptual scores. Its superior LPIPS and DISTS confirm a clear perceptual advantage (lower is better), indicating that OMDNet's reconstructions are more visually plausible and better preserve structure than competing methods across all difficulty levels.

To intuitively illustrate the deblurring capabilities of each model under varying difficulty levels, we present a model efficiency plot (Fig. 5). The horizontal axis shows the perceptual metric index, defined as the geometric mean of the ratios between LPIPS and DISTS values of the restored and original blurry images (lower is better). The vertical axis shows the pixel-level improvement, computed as the geometric mean of the ratios between PSNR_w and $1-\text{SSIM}_w$ of the restored and input images (higher is better). Circle size encodes the relative geometric mean of normalized FLOPs and parameters, reflecting computational demand, while fill grayscale represents relative GPU inference time (darker indicates longer runtime). Circle border colors indicate different network architectures, and line styles denote different test subsets, with a legend in the lower-left corner.

From the figure, it is clear that, within all methods, LBAG (purple circles) shifts toward the upper-left as test difficulty increases, showing stronger suitability for heavily blurred scenarios, while NAFNet shows the opposite trend, favoring lightly blurred inputs. Our method consistently occupies the upper-left region relative to other methods across all subsets, with moderate circle size and grayscale, demonstrating that it achieves both optimal pixel-level fidelity and perceptual quality while maintaining high computational efficiency.

C.2. More Visual Comparison of Local Motion Deblurring

We provide additional visual results on the OMoBlur and ReLoBlur datasets for local motion deblurring, as shown in Figs. 7 to 10. OMDNet demonstrates powerful detail re-

construction in complex motion, and OMoBlur significantly enhances the deblurring capability of various baseline models.

To further evaluate generalization across devices, we capture real-world scenes handheldly with a SONY camera (Fig. 11), and set the aperture, ISO, and exposure time to typical daytime values. In this configuration, camera shake blur is minor whereas object motion blur is prominent, underscoring the importance of handling object motion blur specifically. Our results indicate that OMDNet trained on OMoBlur exhibits strong deblurring performance.

Overall, our dataset excels in three key aspects: the authenticity of the blurred images, the high quality of the sharp images, and the large number of data pairs, which cover a wide variety of object motion types. Our single-image deblurring network, OMDNet, effectively leverages the continuous frame sequences in our dataset, using multi-frame supervision during training to extract motion information from the blurred images, thereby achieving state-of-the-art (SOTA) performance.

C.3. The Effectiveness of Diff-TAM

Section 5.3 in our main paper presents a quantitative ablation. To further demonstrate the effectiveness of Diff-TAM, we evaluate a complex scene (Fig. 6). The football rotates while its rotation axis changes direction and recedes, yielding highly complex motion. According to Eqs. (5)–(6) in main paper, the merged output is formed by backwarping the middle-frame GT to the first and the last frame within the blur interval using the predicted optical flow and add them with the restored middle frame. Therefore, the closer the merge is to its GT, the more accurate the flow prediction and the higher the quality of upstream motion features. When inferring the complex motion near the ball's boundary, pixel-wise error maps may show differences not that large due to weak texture. Nevertheless, it is visually evident that the model without Diff-TAM produces motion inconsistent with the true motion direction, whereas the model with Diff-TAM more authentically recovers the motion. This indicates that Diff-TAM boosts the upstream feature extractor to learn richer motion representations, which in turn improves deblurring.

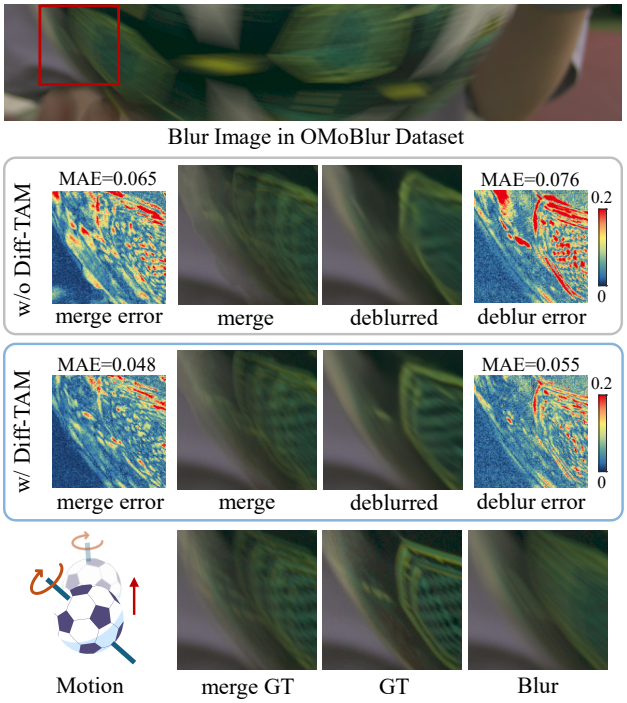

Figure 6. Gains of Diff-TAM for motion estimation and deblurring. We visualize the merge output (a proxy for optical-flow accuracy) and the deblurred result, with and without Diff-TAM, on an extremely challenging scene. Diff-TAM effectively improves complex motion estimation from a single blurry image, thereby strengthening deblurring performance.

References

[1] Haoying Li, Ziran Zhang, Tingting Jiang, Peng Luo, Huajun Feng, and Zhihai Xu. Real-world deep local motion deblurring. In *proceedings of the AAAI conference on artificial intelligence*, pages 1314–1322, 2023. 1, 4

[2] Daniele Menon, Stefano Andriani, and Giancarlo Calvagno. Demosaicing with directional filtering and a posteriori decision. *IEEE Transactions on Image Processing*, 16(1):132–141, 2006. 2

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[4] Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024. 3

[5] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 3

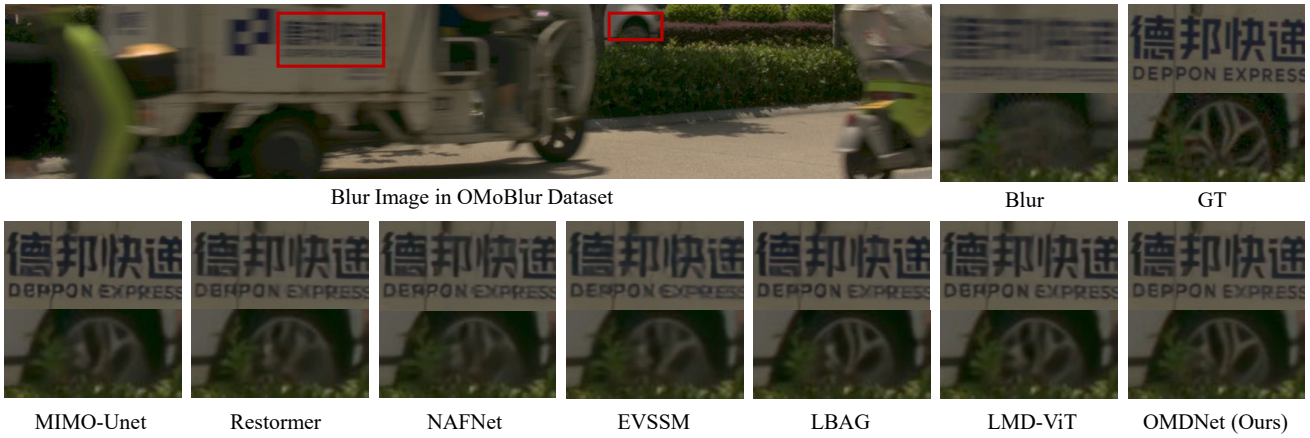

Figure 7. Moving vehicles with a translational component along the optical axis. OMDNet exhibits fewer artifacts and less distortion in the text region, and better detail preservation in the wheels compared to other methods.

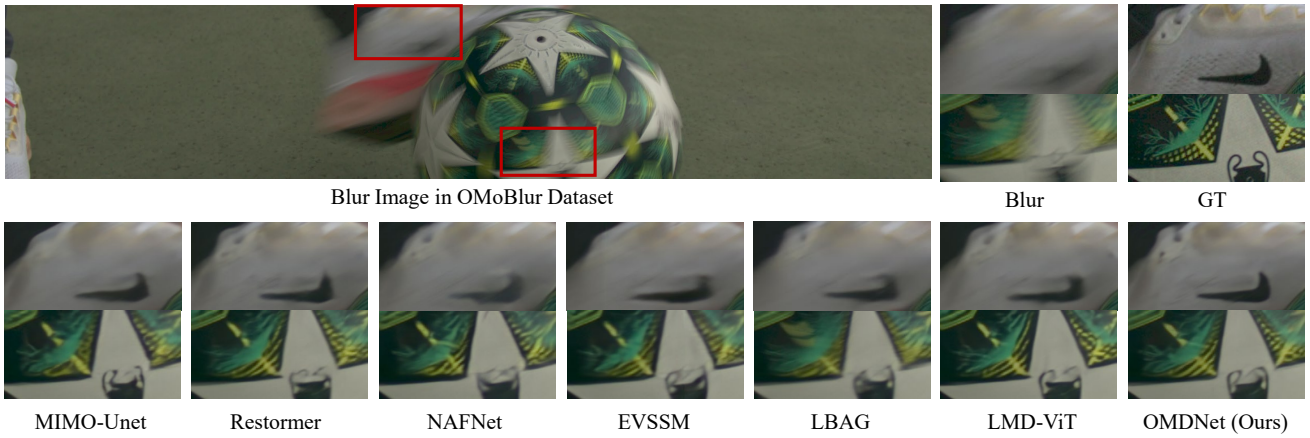

Figure 8. Foot spinning the football. OMDNet demonstrates the best fidelity and detail preservation.

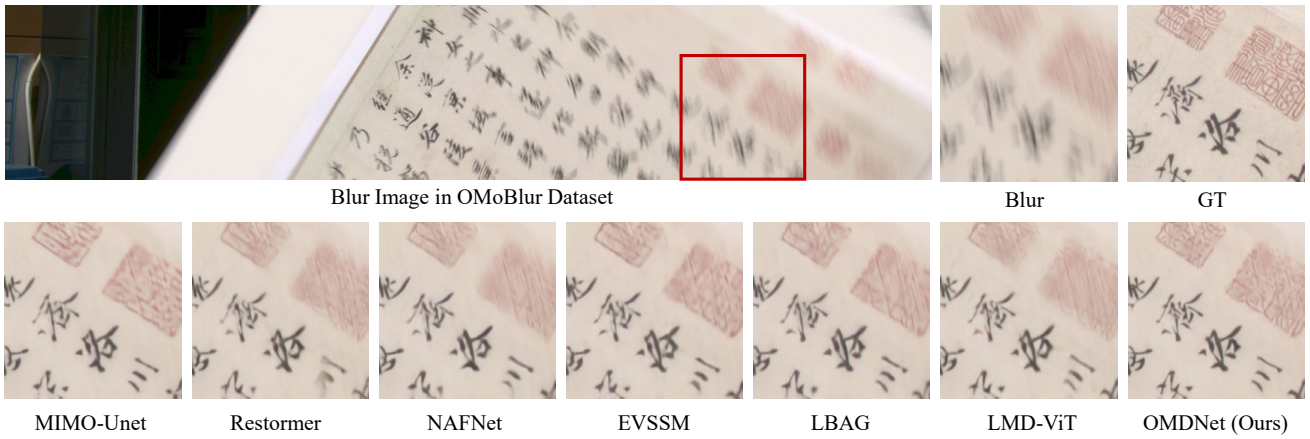

Figure 9. Chinese calligraphy with both translational motion along the optical axis and roll and yaw movements. OMDNet significantly outperforms other methods in reconstructing the intricate details of the text and seal.

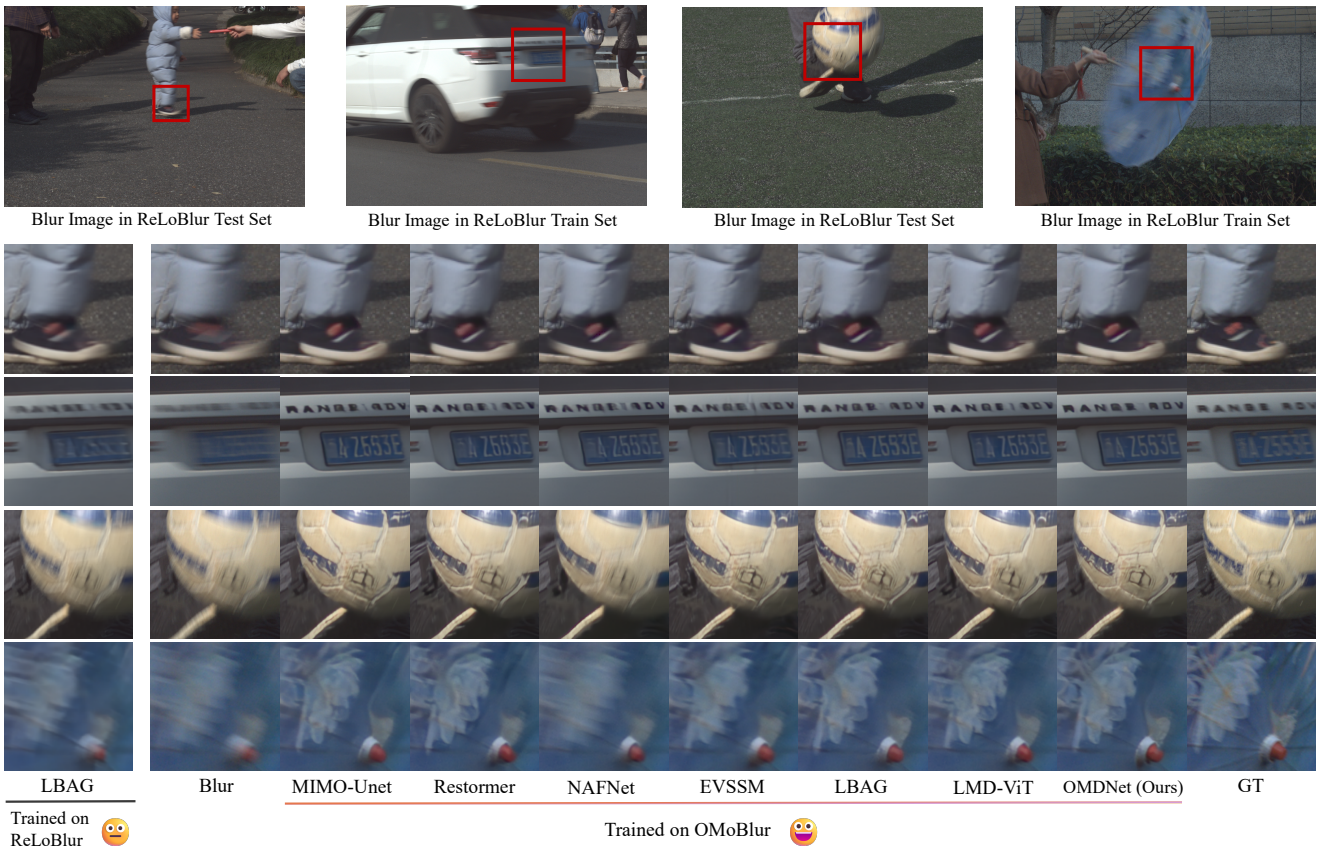

Figure 10. Representative object motions in the ReLoBlur dataset. Networks trained solely on our OMoBlur dataset generalize well on ReLoBlur. Thanks to OMoBlur's large scale and high quality, the visual performance not only surpasses that of LBAG trained on ReLoBlur but also partially exceeds the ground-truth of ReLoBlur.

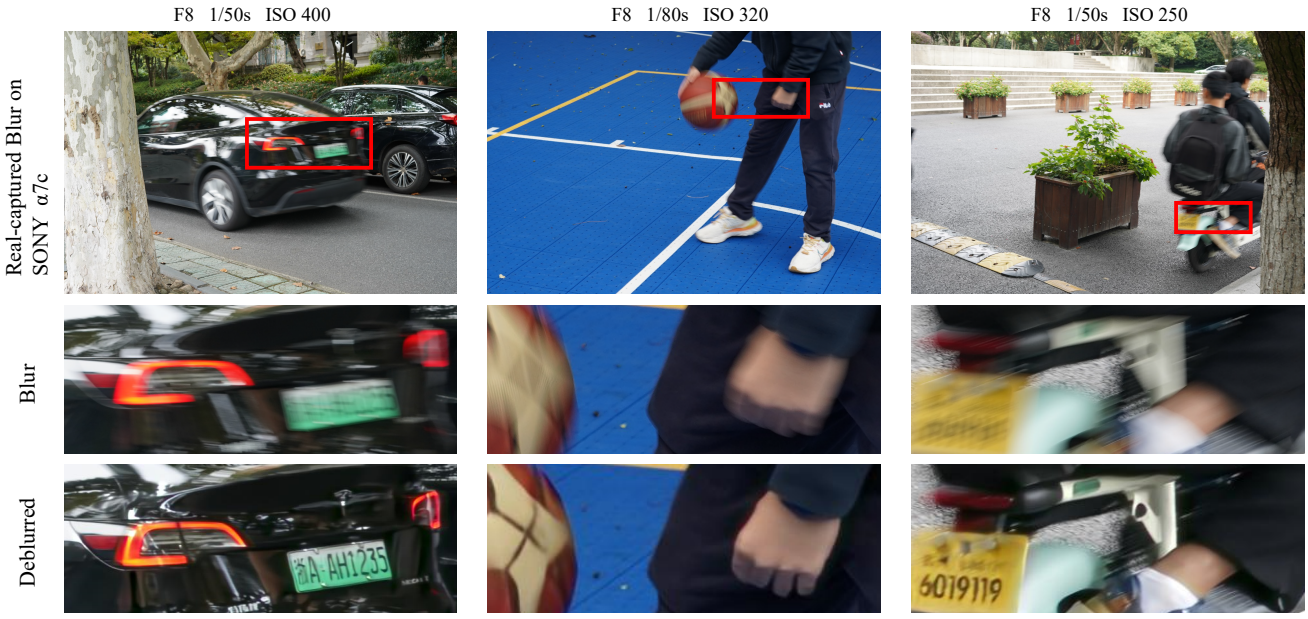

Figure 11. Deblurring results of OMDNet on real-captured blurry images. The images are taken using a SONY α7c camera with a 50 mm prime lens. Capture parameters are listed above each image.