SelfHVD: Self-Supervised Handheld Video Deblurring

Supplementary Material

The content of the appendix involves:

- Presence of sharp frames in Appendix A.
- More details of GoProShake and HVD in Appendix B.
- Effectiveness on object motion blur in Appendix C.
- More ablation studies in Appendix D.
- More evaluation details and results in Appendix E.

A. Presence of Sharp Frames

Modern smartphones from almost all major manufacturers, such as Huawei, Xiaomi and Apple, are equipped with image stabilization technologies like OIS as standard features. In typical handheld scenarios, such as walking or jogging while recording, the shake frequency commonly falls within the effective compensation range of these stabilization systems. As a result, sharp frames consistently appear in handheld video recordings, as illustrated in Fig. A and further supported by the sharp frame ratios reported in Table A, which are typically around 30% across different models from various manufacturers. This observation forms the foundation for our self-supervised approach.

Table A. Recent smartphone models (2022–2024) with OIS support and sharp frame ratio in captured videos

Brand	Model	Release Year	OIS	Sharp Frame Ratio
Huawei	Mate 50	2022	✓	33.33%
	Mate 60	2023	✓	32.50%
	Mate 70	2024	✓	38.85%
Xiaomi	Xiaomi 13	2022	✓	34.34%
	Xiaomi 14	2023	✓	31.91%
	Xiaomi 15	2024	✓	33.68%
Apple	iPhone 14	2022	✓	30.00%
	iPhone 15	2023	✓	32.55%
	iPhone 16	2024	✓	33.11%

B. More Details of GoProShake and HVD

B.1. Synthetic Dataset GoProShake

GoPro [6] chooses to record the sharp information to be integrated over time for blur image generation, which can be formulated as:

$$\mathbf{B} = g\left(\frac{1}{T}\int_{t=0}^{T}\mathbf{S}(t)dt\right) \simeq g\left(\frac{1}{K}\sum_{i=0}^{K-1}\mathbf{S}[i]\right) \quad (1)$$

where T represent the exposure time and $\mathbf{S}(t)$ denote the sensor signal of a sharp image at time t. Similarly, K denotes the number of sampled frames and $\mathbf{S}[i]$ represents the signal of the i-th sharp frame captured during the exposure. The function g is the camera response function (CRF) that maps the latent sharp signal $\mathbf{S}(t)$ to an observed image and is approximated with a gamma curve:

$$g(x) = x^{1/\gamma} \quad (2)$$

where γ is commonly set to 2.2.

Different from GoPro [6], the synthesis process of GoProShake considers the OIS technology in the mobile phone. According to **Characteristic of Handheld Video** in Sec **??**, the blur degree is often proportional to the motion speed of the shooting device. Therefore, unlike GoPro [6], whose number of sampled frames K in Eq. (1) remains odd constant within the same video, in GoProShake, it is proportional to the motion speed between frames. Specifically, we first use MonST3R [14] to roughly estimate the 3D motion trajectory of the mobile phone, and then calculate the movement distance between frames based on the motion trajectory:

$$d_i = \int (v_r + v_s)dt \quad (3)$$

where v_r and v_s represent the rotational and translational velocity vector of the mobile phone, respectively. From the pose obtained by MonST3R [14], we can calculate the rotational and translational distances vector between frames, then we can get the rotational velocity vector v_r and translational velocity vector v_s from the distances vector.

Our synthesis process also uses a sliding window approach, with the window size and step size set to the same value as GoPro [6], which is the number of sampled frames K. Therefore, the sequence number m_j of the middle frame of the j-th sliding window is:

$$m_j = j * K + K/2 \quad (4)$$

where $j = 0, 1, 2...$ and we can calculate the average movement distance in the j-th sliding window as:

$$\bar{d}_j = \frac{1}{K}\sum_{i=m_j-K/2}^{m_j+K/2} d_i \quad (5)$$

Then the number of sampled frames for each sliding window is:

$$k_j = \min\left(1, K * \frac{\bar{d}_j}{D}\right) \quad (6)$$

where D is a normalization constant. The number of sampled frames k_j is proportional to the movement distance $\bar{d}_j$.

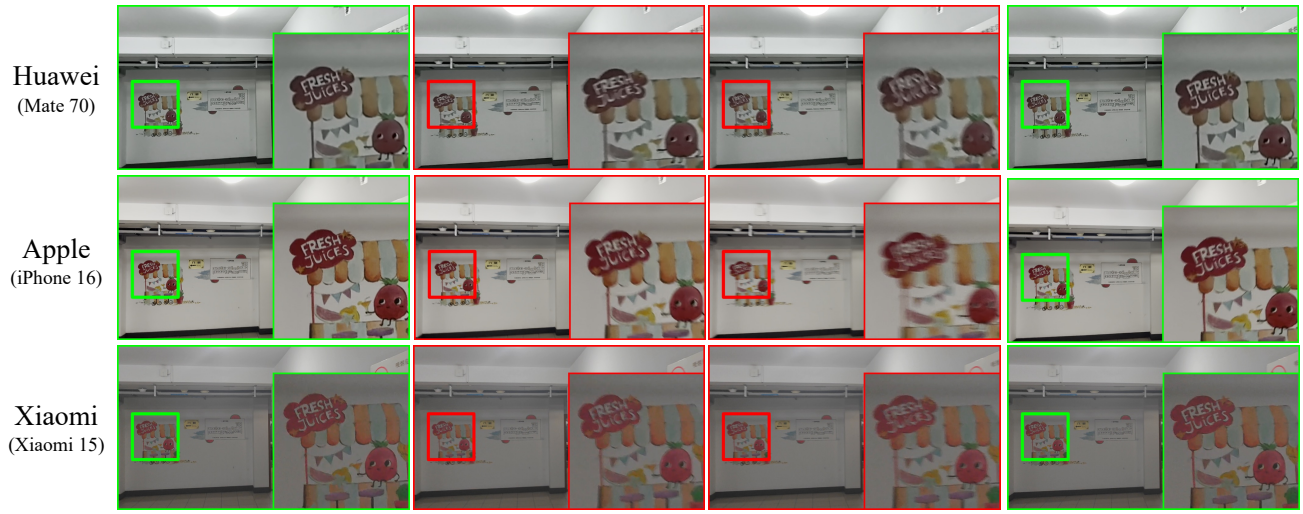

Figure A. Illustration of sharp and blurry frames coexisting in videos captured by different smartphones (Huawei, Apple, and Xiaomi) during handheld walking. Each row corresponds to a different device. Green boxes highlight sharp frames, while red boxes indicate blurry frames.

The smaller the movement distance, the fewer frames are sampled. The final synthetic frame B_j in the j-th sliding window is:

$$\mathbf{B}_j = g\left(\frac{1}{k_j}\sum_{i=m_j-k_j/2}^{m_j+k_j/2}\mathbf{S}[i]\right) \tag{7}$$

where interpolation processing is applied before averaging and g is the estimated CRF in [7]. It is noted that the j-th frame is sharp when $k_j = 1$. Our GoProShake dataset contains 22 training videos and 11 test videos, consistent with GoPro [6]. The visualization of the video from GoProShake as shown in Fig. B

B.2. Real-World Dataset HVD

The videos of HVD are captured by walking normally in various scenes, such as *night scenes of commercial streets, campuses, underground parking lots* and *subway stations*, using HUAWEI P40, Xiaomi 15 and iPhone 16. All videos are recorded at a frame rate of 30fps with an exposure time of 16ms. HVD contains a total of 180 videos, with 120 used for training and 60 (20 Huawei P40, 20 Xiaomi 15, and 20 iPhone 16) for testing. The visualization of the video from HVD as shown in Fig. 1(a) and Fig. C.

C. Deblurring Effects on Object Motion Blur

Our method is capable of handling not only camera motion blur but also object motion blur. As described in the main paper, due to object motion is typically non-uniform, relatively sharp content is retained when the object is still or moves slowly. Video deblurring models can aggregate information across multiple frames, allowing these sharp contents to provide crucial clues for dealing with blur when the object moves fast. Compared to deblurring results from the blurry video without sharp clues, those from the original blurry video perform better on object motion, as illustrated in the middle row of Fig. **??** in the main paper and Fig. E. The high-quality training pairs constructed by SEVD also offer more reliable supervision for object motion deblurring. As a result, our method achieves better than DaDeblur [4] on the object motion blur. Some visualizations on HVD can be seen in Fig. F.

D. More Ablation Studies

D.1. Effect of the Masks

Table C shows the ablation results of the $\mathbf{M}_{uncer}$ and the $\mathbf{M}_{occ}$. Both masks individually bring performance gains, indicating their effectiveness in handling uncertain or occluded regions. And combining both yields the best results, highlighting their complementary roles in enhancing reconstruction quality. Fig. I and Fig. J visualize the proposed masks on the synthetic dataset GoProShake and the real-world dataset HVD, respectively. As shown in the figures, the masks effectively identify and suppresses misaligned regions that result from inaccurate optical flow or large content discrepancies between frames. This prevents erroneous supervision and ensures that only reliable regions contribute to the learning process.

D.2. Effect of SEVD and SCSCM

Table B presents the ablation results of SEVD and SCSCM on two backbones (RVRT [5] and BasicVSR++ [1]) across both the synthetic dataset GoProShake and the real-world dataset HVD. Individually introducing SEVD or SCSCM improves performance across most metrics, validat-

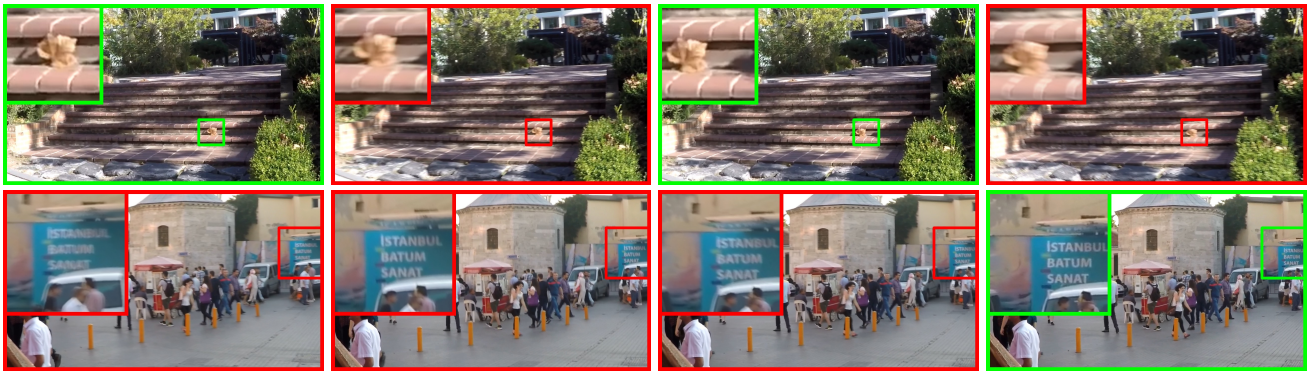

Figure B. Visualization of GoProShake dataset. The top and bottom are training and test videos, respectively. GoProShake takes into account the OIS technology on handheld video capture, synthesizing blurry videos (red boxes) that contain sharp frames (green boxes).

Figure C. Visualization of HVD dataset. Sharp frames (green boxes) are present and reliable in most cases of handheld shooting scenarios.

Table B. SEVD and SCSCM ablation across backbones (RVRT [5] and BasicVSR++ [1]) and datasets (GoProShake and HVD).

SEVD	SCSCM	**RVRT on GoProShake** PSNR / SSIM	**RVRT on HVD** MUSIQ / MANIQA	**BasicVSR++ on GoProShake** PSNR / SSIM	**BasicVSR++ on HVD** MUSIQ / MANIQA
✗	✗	34.34 / 0.9155	26.9798 / 0.2098	35.61 / 0.9263	26.9677 / 0.2060
✗	✓	36.11 / 0.9288	27.6052 / 0.2189	37.09 / 0.9342	27.7905 / 0.2103
✓	✗	35.89 / 0.9210	27.2834 / 0.2149	36.67 / 0.9290	27.2445 / 0.2061
✓	✓	**36.31 / 0.9300**	**28.4142 / 0.2627**	**37.44 / 0.9359**	**28.0040 / 0.2175**

Table C. Effect of $\mathbf{M}_{uncer}$ and $\mathbf{M}_{occ}$.

M_{uncer}	M_{occ}	PSNR↑	SSIM↑
✗	✗	36.13	0.9157
✗	✓	37.25	0.9334
✓	✗	37.00	0.9343
✓	✓	**37.44**	**0.9359**

ing their respective contributions. Notably, the combination of SEVD and SCSCM consistently achieves the best performance in all settings, highlighting their complementary effectiveness across different backbones and datasets.

D.3. Effect of the Optical Flow Model

We investigate the impact of different optical flow models on our deblurring performance. As shown in Table D, replacing SEA-RAFT [12] with RAFT [11] or FlowFormer++[10] results in PSNR drops of 0.66dB and 0.20dB, respectively, and slight SSIM declines. This demonstrates that SEA-RAFT provides more accurate optical flow estimation, enabling better frame alignment and enhanced restoration quality. Furthermore, Fig.D illustrates the robustness of SEA-RAFT under varying degrees of blur, where it consistently yields reliable flow predictions even in severely degraded regions.

D.4. Effect of Sharp Frame Selection Interval

Table E investigates how different sharp frame selection intervals affect selection accuracy and deblurring performance on GoProShake. The accuracy is computed by comparing our selected sharp frames with manually labeled

Table D. Effect of the optical flow method.

Optical Flow Method	PSNR↑	SSIM↑
RAFT	36.78	0.9353
FlowFormer++	37.24	0.9327
SEA-RAFT	**37.44**	**0.9359**

Table E. Selection accuracy and deblurring performance under different sharp frame selection intervals on GoProShake. An interval of k means one sharp frame is selected for every k frames.

Selection Interval	Selection Accuracy	PSNR↑/SSIM↑
5	71.03%	37.33 / 0.9305
10	88.51%	37.36 / 0.9342
20	96.77%	**37.44 / 0.9359**
30	**98.28**%	37.03 / 0.9326

ones. A smaller interval 5 yields denser supervision but lower accuracy 71.03%. Increasing the interval enhances selection accuracy, reaching 98.28% at interval 30, but this comes at the cost of sparsity in supervision. Among all settings, an interval of 20 achieves the best deblurring performance, striking a good balance between selection reliability and coverage. Therefore, we adopt this interval in all main experiments. We also apply the same sharp frame selection strategy to the real-world HVD dataset, and observe reasonable accuracy 91.88%, confirming that real videos also contain sharp frames that can serve as reliable supervision.

D.5. Effect of Supervised Pre-training.

To assess the effect of supervised pre-training, we first evaluate the performance of the fully supervised BasicVSR++ trained on different datasets, as shown in the upper part of Table F. These results serve as baselines. We then apply our self-supervised method, SelfHVD$_{BasicVSR++}$, to fine-tune these pre-trained models on our real-world dataset HVD. The lower part of the table presents the results after self-supervised adaptation. Regardless of the pre-training dataset, SelfHVD$_{BasicVSR++}$ consistently improves quality over the original supervised models. And better supervised pre-training generally results in better performance after self-supervised fine-tuning. These results demonstrate that our self-supervised method effectively enhances the performance of different pre-trained models on real-world handheld blurry videos.

E. More Evaluation Details and Results

E.1. More Evaluation Details

Under full-supervision, the GoProShake training set (w/ GT) is used for training, while the GoProShake and HVD-Huawei test sets, as well as the HVD-Xiaomi and HVD-iPhone, are used for evaluation. Under self-supervision, for synthetic data, the GoProShake training set (w/o GT) is used for training and the GoProShake test set is used for evaluation; for real-world data, the HVD-Huawei training set is used for training and the HVD-Huawei test set, HVD-Xiaomi, and HVD-iPhone are used for evaluation.

Table F. Quantitative comparison on real-world HVD dataset. 'Pre-training' denotes the dataset used for pre-training models. The best results in each category are **bolded**, and the second-best results are underlined.

	Methods	Pre-training	MUSIQ↑	MANIQA↑
Fully-Supervised	BasicVSR++	BSD-2-16	**26.5821**	**0.2303**
		GoPro	25.9927	0.2014
		GoProShake	25.2488	0.2006
Self-Supervised	SelfHVD$_{BasicVSR++}$	-	28.0040	0.2175
		BSD-2-16	28.1463	0.2135
		GoPro	27.7622	0.2189
		GoProShake	**28.2212**	**0.2231**

Table G. Model complexity and average inference time comparison of different video deblurring backbones.

Networks	#Params(M)	#FLOPs(G)	Time(ms)
IFIRNN [8]	1.64	29.55	16.53
ESTRNN [15]	2.47	146.96	79.31
RVRT [5]	10.78	1379.84	472.50
BasicVSR++ [1]	9.76	72.53	27.38

Table H. Temporal consistency comparison of self-supervised methods on the synthetic dataset GoProShake. The best results in each category are **bolded**, and the second-best results are underlined.

Methods	tOF↓	tLP↓	FVD↓	VBench↑
Ren *et al.* [13]	4.9773	3.8688	112.60	0.8978
DaDeblur [4]	2.0680	2.2800	31.40	0.9018
SelfHVD$_{IFIRNN}$	**1.5423**	1.5953	**6.18**	0.9064
SelfHVD$_{ESTRNN}$	1.7895	1.8452	7.24	0.9044
SelfHVD$_{RVRT}$	1.7451	1.4712	6.85	0.9065
SelfHVD$_{BasicVSR++}$	1.5911	**1.1539**	6.71	**0.9069**

E.2. More Visual Results

To further validate the visual effectiveness of our method, we present additional qualitative comparisons in Figs.G and H. As shown in Fig.G, SelfHVD$_{\text{BasicVSR++}}$ consistently generates sharper results on our synthetic dataset GoProShake, outperforming previous self-supervised approaches, and illustrates the robustness of our method on the real-world dataset HVD. Lastly, Fig.H demonstrates that under the same test-time training setting as DaDeblur, SelfHVD achieves better visual quality on BSD [15], RBVD [2], and RealBlur [9]. These results further support the quantitative improvements reported in the main paper and confirm the generalization capability of SelfHVD across both synthetic

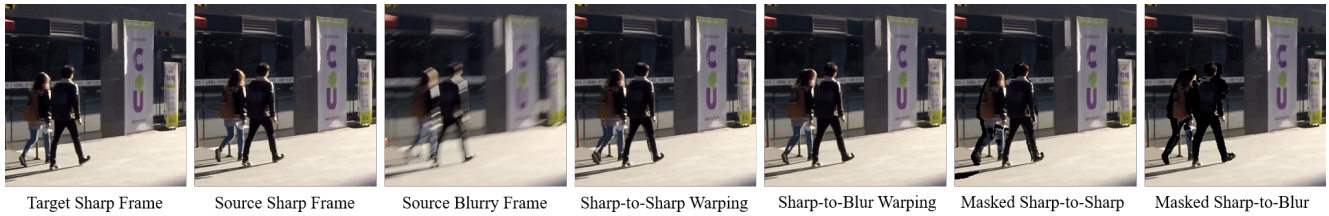

Figure D. Qualitative alignment results under different blur levels. The source sharp and blurry frames are generated by fusing different numbers of high-frame-rate images, with the sharp frame typically being a single mid-frame and the blurry frame formed by averaging multiple consecutive frames.

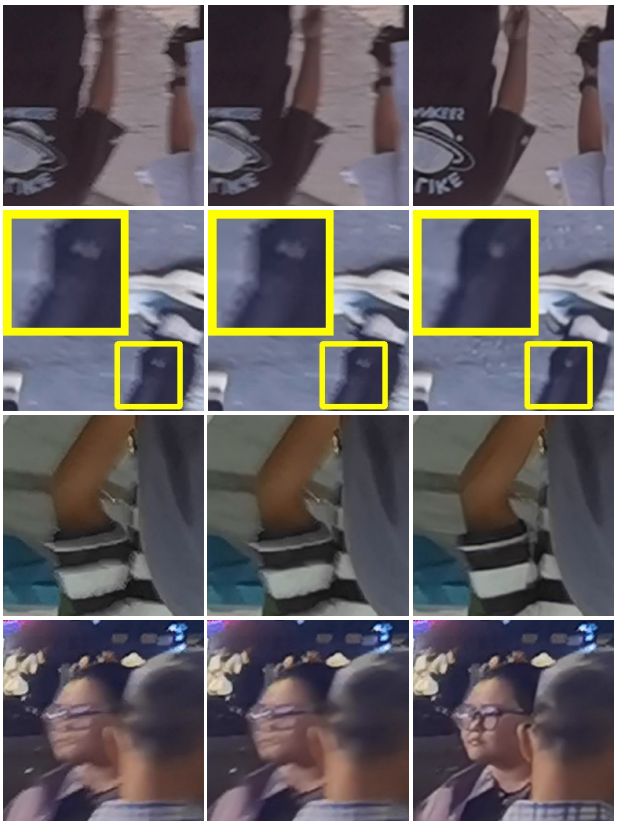

Figure E. From left to right: sharp-clues-less blurry video, deblurring result of sharp-clues-less blurry video, deblurring result of original input video. SEVD improves object motion blur handling by constructing higher-quality paired data.

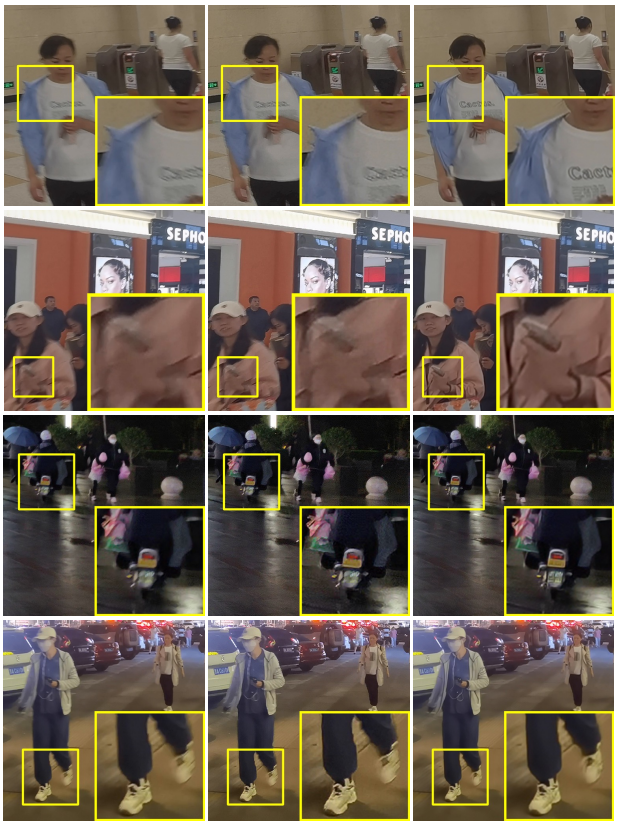

Figure F. Visual comparisons on handling object motion blur. Even for challenging cases involving object motion blur, our method achieves better performance compared to DaDeblur.

and real-world datasets.

E.3. Running Efficiency

Since our framework can be applied to various video deblurring networks, we select representative backbones with different architectures and model sizes, including IFIRNN [8], ESTRNN [15], RVRT [5], and BasicVSR++ [1], where ESTRNN [15] is also adopted by Ren *et al*. [13] and DaDeblur [4]. The model complexity (numbers of parameters (#Param), FLOPs (#FLOPs)), and the average inference time (Time) are shown in Tab. G.

E.4. Temporal Consistency

We adopt tOF and tLP [3] as temporal consistency metrics. As shown in Tab. H, SelfHVD built on different backbones (IFIRNN [8], ESTRNN [15], RVRT [5], and BasicVSR++ [1]) consistently achieves lower tOF and tLP values on GoProShake than previous self-supervised methods Ren *et al*. [13] and DaDeblur [4]. These results demonstrate the better temporal consistency of SelfHVD.

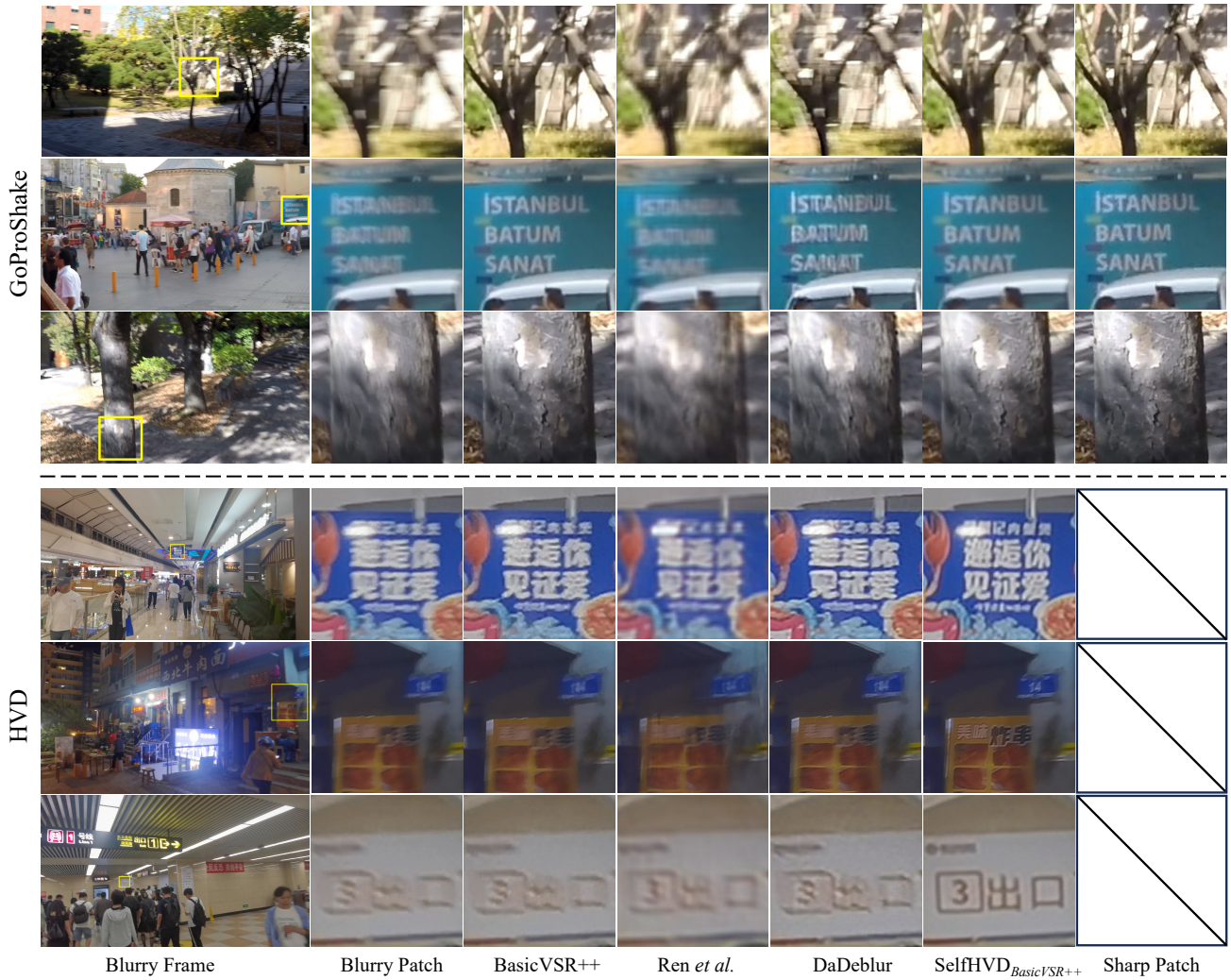

Figure G. More qualitative comparison on our synthetic dataset GoProshake and real-world dataset HVD.

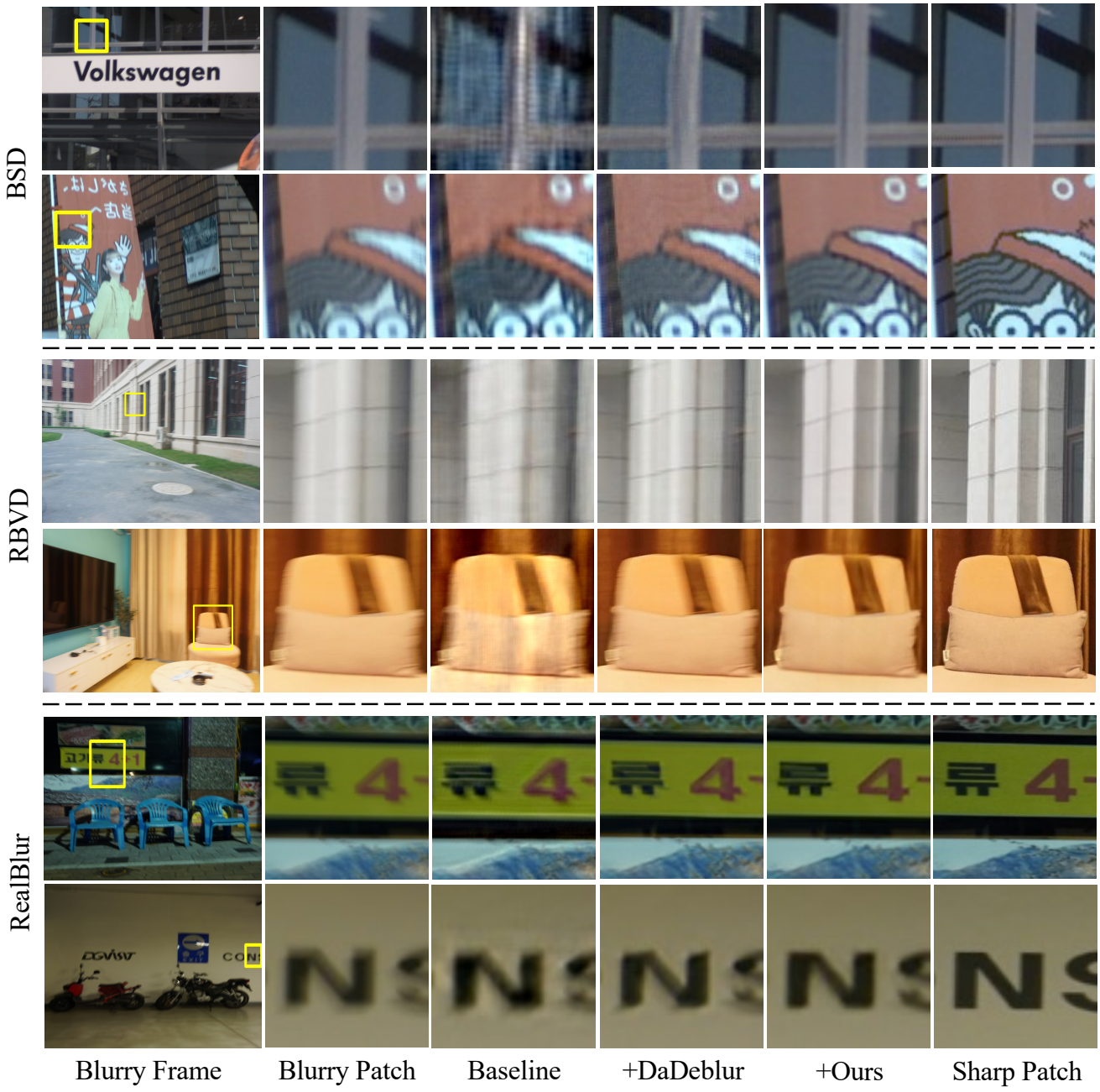

Figure H. More qualitative comparison on BSD, RVRB and RealBlur.

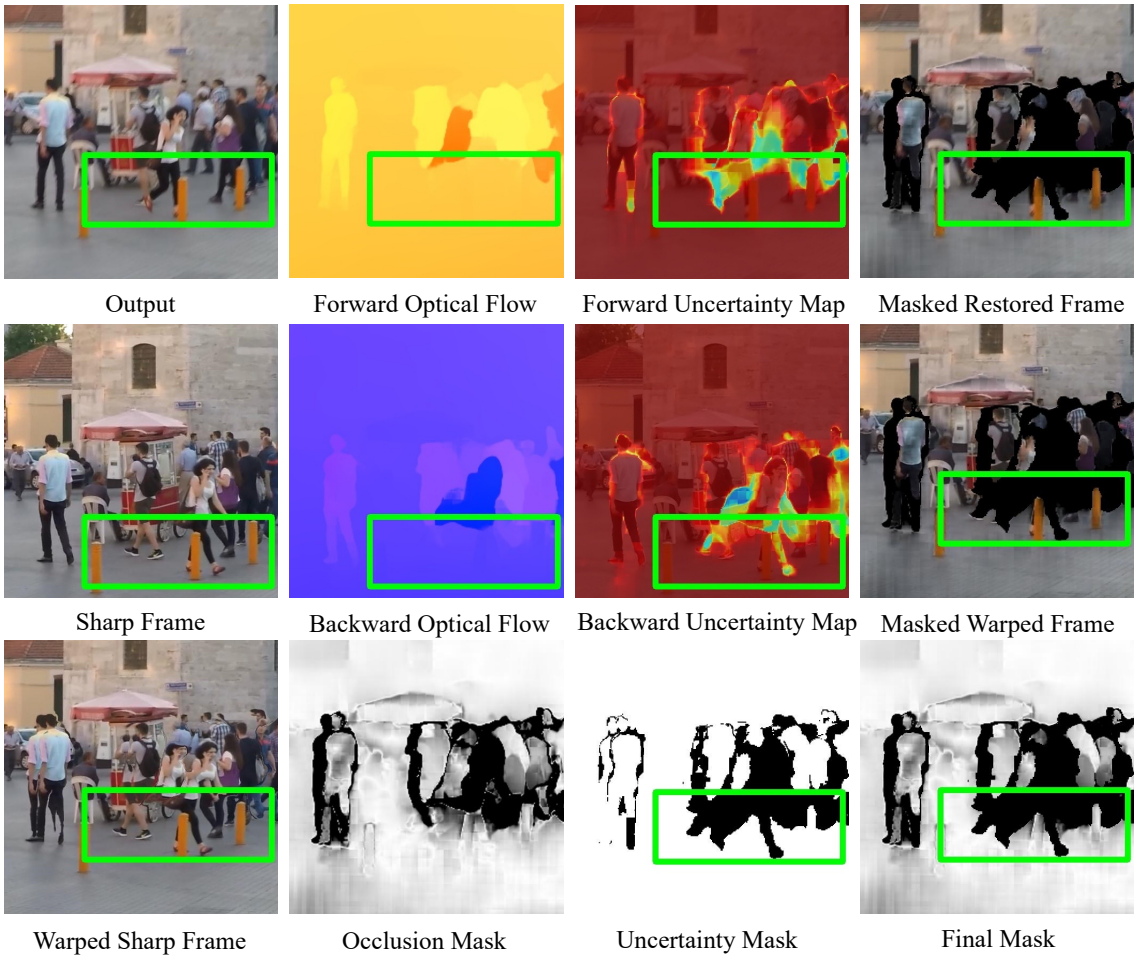

Figure I. Visualization of the masks on the synthetic dataset GoProShake. The green box indicates the region where the optical flow is inaccurate. The uncertainty map will perceive the inaccurate region, and the uncertainty mask is calculated from it to mask the region.

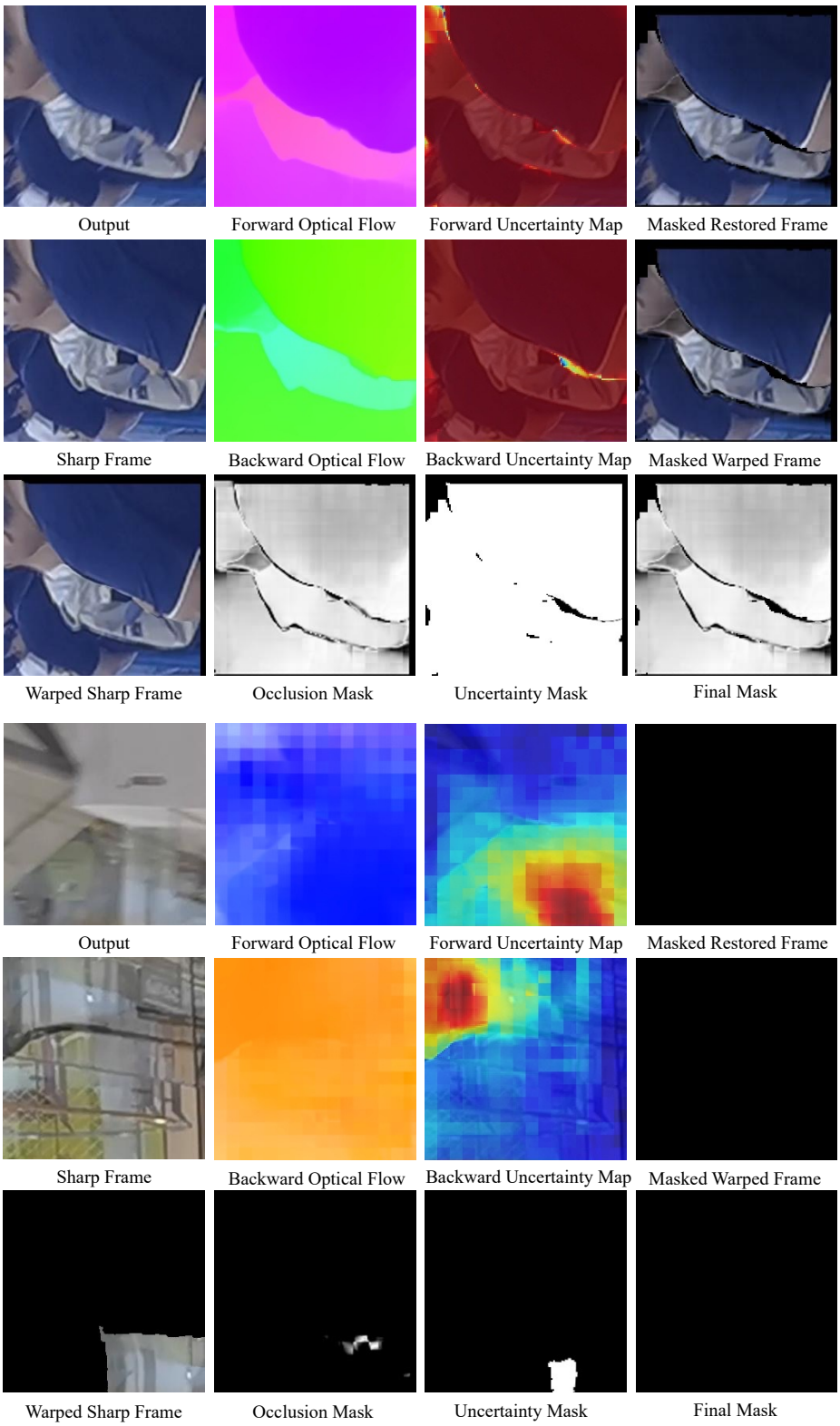

Figure J. More mask visualization on the real-world dataset HVD, showing the behavior of our masks under varying degrees of content discrepancy between the predicted output and the sharp frame.

References

[1] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, 2020. 2, 3, 4, 5

[2] Zhu Chao, Dong Hang, Pan Jinshan, Liang Boyang, Huang Yuhao, Fu Lean, and Wang Fei. Deep recurrent neural network with multi-scale bi-directional propagation for video deblurring. In *AAAI*, 2022. 4

[3] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *TOG*, 2020. 5

[4] 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. 2, 4, 5

[5] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc Van Gool. Recurrent video restoration transformer with guided deformable attention. *arXiv*, 2022. 2, 3, 4, 5

[6] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 1, 2

[7] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRW*, 2019. 2

[8] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *CVPR*, 2019. 4, 5

[9] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, 2020. 4

[10] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *CVPR*, 2023. 3

[11] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 3

[12] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *ECCV*, 2024. 3

[13] Qifeng Chen Xuanchi Ren, Zian Qian. Video deblurring by fitting to test data. In *arxiv*, 2020. 4, 5

[14] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv*, 2024. 1

[15] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *ECCV*, 2020. 4, 5