Supplementary Material for
Deblurring via Stochastic Refinement

1. Additional Perception-Distortion Plots

The Perception-Distortion plot provided in Section 1 of the main text shows the trade-off between PSNR and Kernel Inception Distance (KID). We observe that other combinations of perceptual (NIQE, LPIPS, FID) and distortion metrics (PSNR, SSIM) follow a similar trend, as shown in Figure 1. We note that formally LPIPS is also a distortion metric, as it is a full-reference based distance computed in a deep feature space. We nonetheless observed that LPIPS corresponds to human perception much better than PSNR or SSIM.

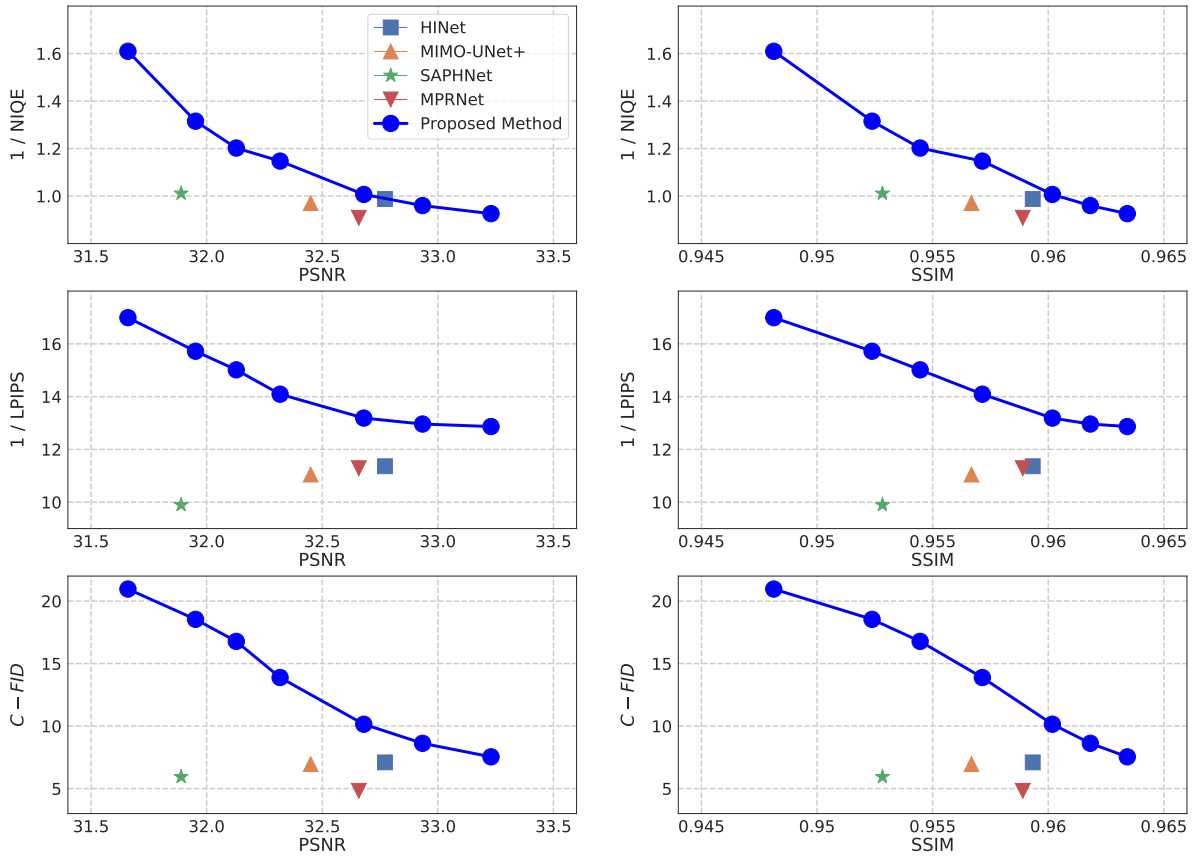

Figure 1. Additional Perception-Distortion plots with respect to different metrics. Left column contains perceptual metrics vs. PSNR, and the right column contains SSIM comparisons. We notice that the same trade-off is present for all (perceptual, distortion) metric pairs.

2. Diversity Analysis

Figure 2 shows the relation between the blurriness (or sharpness) on the input image, and the diversity of the generated deblurred samples. The blurrier the input image is, the more diversity we get in the samples (see figure caption for more details).

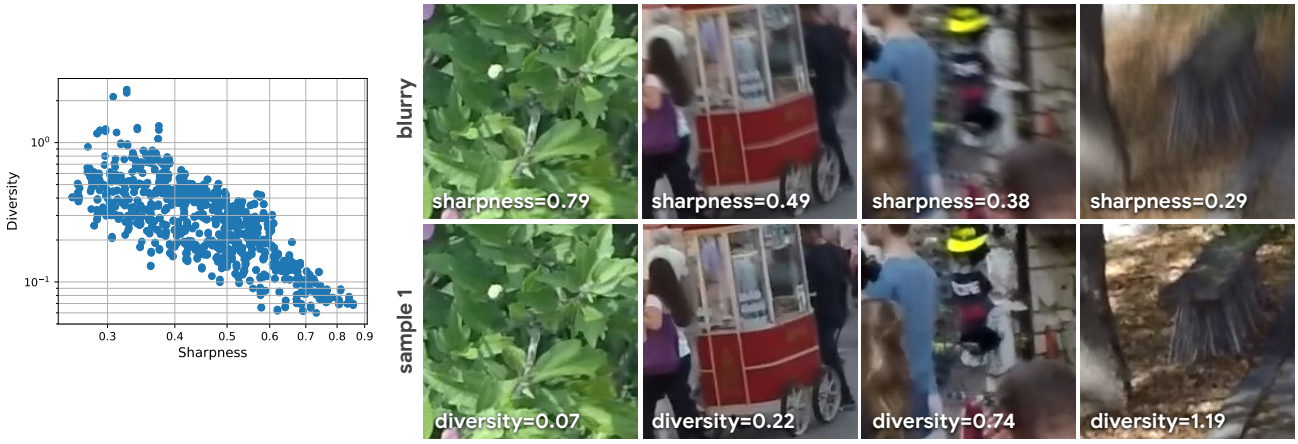

Figure 2. Sample diversity as a function of input image sharpness. The ill-posedness of the restoration task (*i.e.* how strong the blur is) has a direct impact on the diversity of the generated samples. **Left:** Each point in this plot represents an image in the GoPro validation set. Image sharpness is computed as: sharpness $= \|\Delta\text{input}\|/\|\Delta\text{reference}\|$, where Δ is the Laplacian of the given image. Sample diversity is computed as: diversity $= \|\text{Var}\,[\text{sample}]\,\|/\|\Delta\text{reference}\|$, where Var[sample] is the per pixel empirical variance of multiple restored images for a given input. **Right:** Four different blurry image crops with different level of sharpness, and a respective deblurred sample for each one (sample 1).

3. Synthetic DIV2K Deblurring Dataset

To better analyze various aspects of our diffusion deblurring model, we created a custom dataset by applying synthetic camera shake blur (following [5] and noise to the DIV2K dataset [1]. This allows us to make qualitative evaluations in a more controlled environment, since the low-quality ground truth images in existing paired datasets [10, 11] make qualitative assessment difficult and lessens the benefits from using a powerful generative model.

The synthetically generated random kernels are of varying size (31×31 maximal support). Figure 3 shows example kernels. The kernels can be of any size from a perfect Delta (sharp) to about 30 pixels. In addition to the blur, a white Gaussian noise with random standard deviation $\sigma \sim \mathcal{U}[0, 15]$ is added.

4. Omitted Details for DPM Formulation

Equation (2): Marginal at time step t. We proceed by induction. For $t = 1$, we have $\bar{\alpha}_1 = \alpha_1$, so Eq. (2) reduces to the diffusion transition kernel:

$$q(\boldsymbol{x}_1 \mid \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_1; \sqrt{\alpha_1}\boldsymbol{x}_0, (1-\alpha_1)\boldsymbol{I}_d\right).$$

Now suppose we have $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}_d)$ for some $t > 1$, which we reparameterize as

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d).$$

Then by applying a single diffusion step $q(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t)$ to the above, we get

$$\begin{aligned}
\boldsymbol{x}_{t+1} &\overset{(1)}{=} \sqrt{\alpha_{t+1}}\boldsymbol{x}_t + \sqrt{1-\alpha_{t+1}}\boldsymbol{\epsilon}' \\
&\overset{(2)}{=} \sqrt{\alpha_{t+1}}\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{\alpha_{t+1}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} + \sqrt{1-\alpha_{t+1}}\boldsymbol{\epsilon}' \\
&\overset{(3)}{=} \sqrt{\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \sqrt{\alpha_{t+1}-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon} + \sqrt{1-\alpha_{t+1}}\boldsymbol{\epsilon}' \\
&\overset{(4)}{=} \sqrt{\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon}'',
\end{aligned}$$

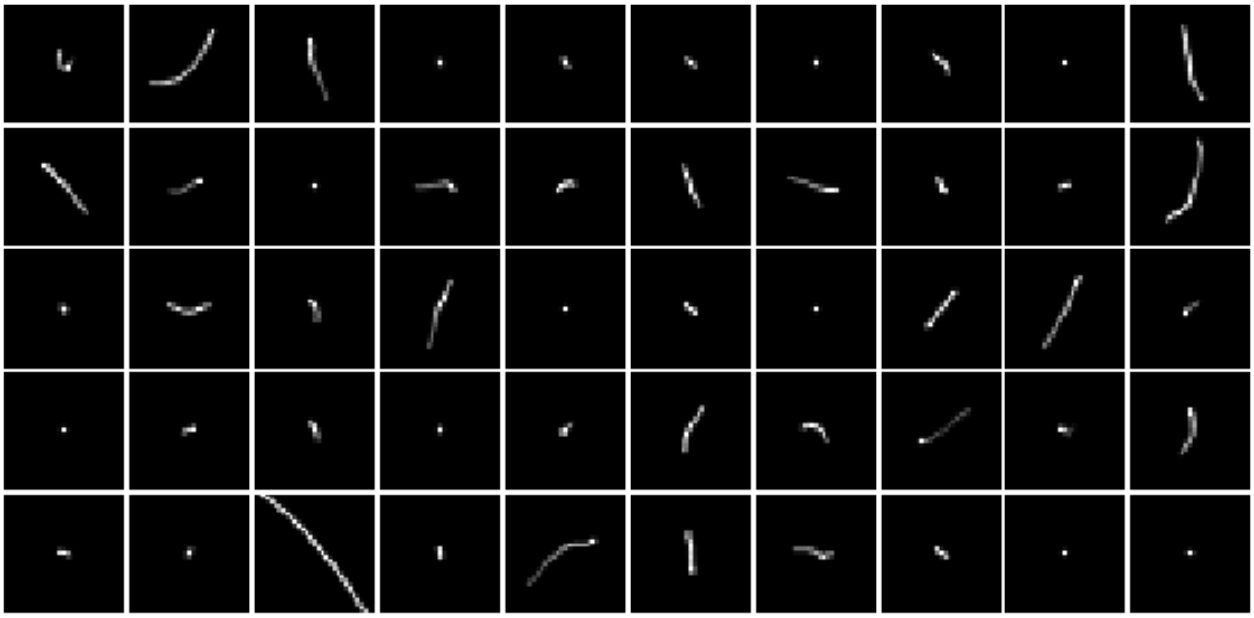

Figure 3. Examples of synthetically generated random kernels (following [5]) used to generate the deblurring dataset.

where the first step uses a reparameterization $\boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, the second step is from the inductive hypothesis, and the last step follows from summing two independent Gaussian random variables. Thus

$$\boldsymbol{x}_{t+1} \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_{t+1}}\boldsymbol{x}_0, (1-\bar{\alpha}_{t+1})\boldsymbol{I}_d\right),$$

which concludes the inductive step.

Reverse diffusion step expressions. Applying Bayes' Rule to Eq. (3) leads to the following expressions for the mean and variance for the reverse diffusion step:

$$\boldsymbol{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t,$$
$$\beta_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}(1-\alpha_t).$$

We refer the reader to Ho *et al*. [6] for a more thorough treatment of the DPM formulation.

Specifying the noise schedule. Following [3, 13], given a fixed budget of T steps, we sample the *continuous* noise level $\sqrt{\bar{\alpha}}$ from a piecewise uniform distribution. Specifically, we define T intervals (l_{i-1}, l_i), where $l_0 \triangleq 1$ and $l_i \triangleq \sqrt{\bar{\alpha}_i}$ for $i > 0$. Then to sample a continuous noise level $\bar{\alpha}$, we first randomly pick an interval (l_{k-1}, l_k), and sample $\bar{\alpha} \sim \mathcal{U}[l_{k-1}, l_k]$.

Now all that remains is to specify the schedule $\alpha_1, \ldots, \alpha_T$. While there are many options (*e.g*. as explored by Chen *et al*. [3]), we used a simple linear schedule on the variance of the forward process by fixing the two endpoints and linearly interpolating the intermediate values.

5. Model Details

Network architecture. We use a U-Net [12] architecture similar to the one used by SR3 [13]. A crucial difference is that our network was made fully-convolutional by removing self-attention, group normalization, and positional encoding. At the input, the noisy sample $\boldsymbol{x}_t$ is concatenated with the conditioning input $\boldsymbol{y}$ channel-wise.

As shown in Fig. 4, our U-Net has four resolution depths with channel multipliers $\{1, 2, 3, 4\}$. Both the denoiser network and initial predictor use this architecture. Their main difference is size, where the starting channel count is 64 for the initial

predictor and 32 for the denoiser. This results in the initial predictor having ∼26M parameters, and the denoiser having ∼7M parameters. Note that the input and output would change slightly when this architecture is used for the initial predictor, which tries to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$ (no $\boldsymbol{x}_t$ and $\bar{\alpha}$ in the input, and the output is not $\boldsymbol{\epsilon}$).

Training details. We train all of our models for 1M steps using 32 TPUv3 cores. For our main model with the initial predictor and the denoiser network, it takes about 27 hours to train the model. We used the AdamW [9] optimizer with a fixed learning rate of 0.0001, weight decay rate of 0.0001, and EMA decay rate of 0.9999. During training, we used fine-grained diffusion process with $T = 2000$ steps. As described above, we used a linear noise schedule with the two endpoints set as: $1 - \alpha_0 = 1 \times 10^{-6}$ and $1 - \alpha_T = 0.01$.

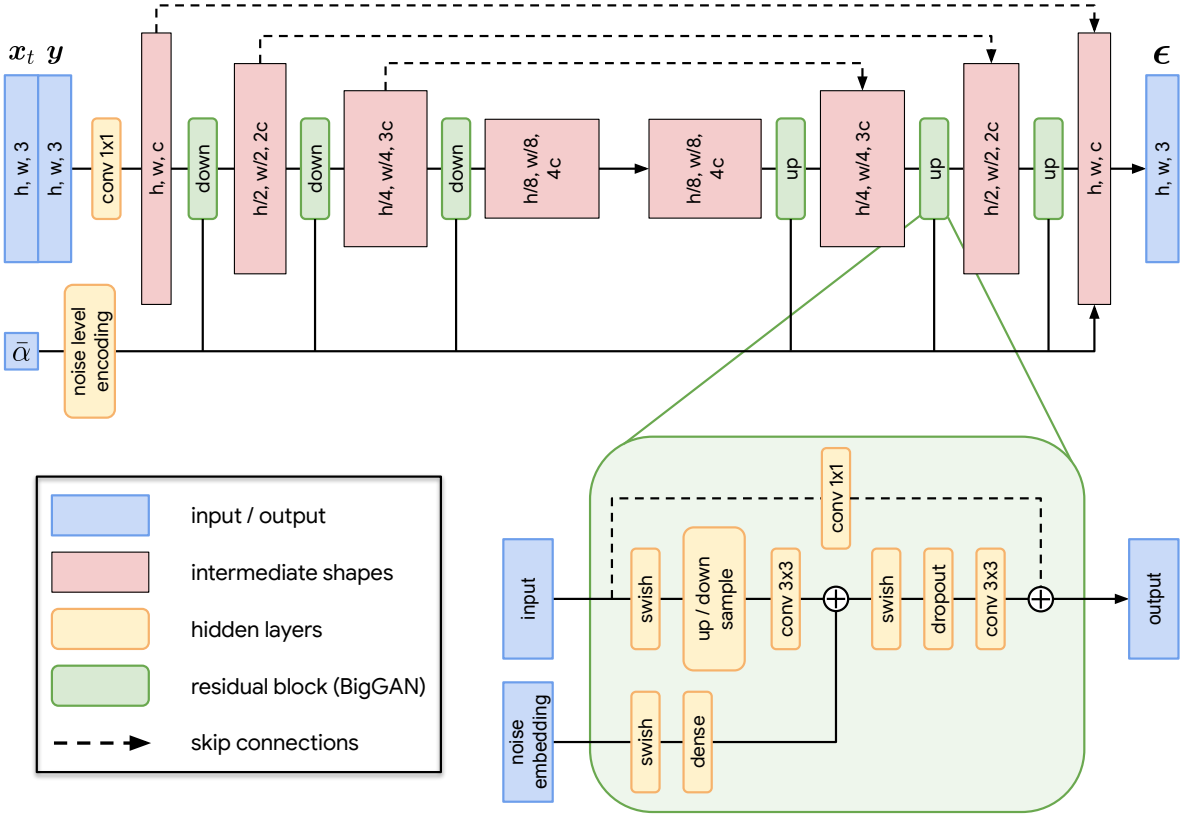

Figure 4. Diagram describing the U-Net architecture used for both the denoiser network and the initial predictor in our experiments. Note that the input and output depicted here are for the denoiser network.

6. Evaluation Details

For all our experiments (on all datasets: GoPro, HIDE, DIV2K), we performed a grid search over the following hyperparameter combinations during inference:

1. Inference steps (T): 10, 20, 30, 50, 100, 200, 300, 500.
2. Noise schedule ($\boldsymbol{\alpha}_{1:T}$): We fixed the initial forward process variance ($1 - \alpha_0$) to 1×10^{-6}. For the final variance ($1 - \alpha_T$), we sweep over $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$. The intermediate values are linearly interpolated.

How baseline samples are obtained. As mentioned in Sec. 5 of the main text, we computed various perceptual metrics ourselves as the existing literature often only reports PSNR and SSIM. To ensure fairness in our comparisons, we tried to use author-produced restoration results whenever possible. Otherwise, we used the official implementations and pre-trained models released by the authors of each paper and produced restorations ourselves.

Specifically, for HINet [2], MPRNet [16], and SAPHNet [15], we used restorations produced by the authors for both GoPro and HIDE results. For MIMO-UNet+ [4] and DeblurGANv2 [7], we used the authors' implementation and model checkpoints from their respective Github repositories. For SimpleNet [8], we could not obtain either the restorations nor the code, so we only reported the metrics from the paper (PSNR, SSIM, LPIPS).

7. Large GoPro and HIDE Results

In Figures 5–7, we include larger versions of the GoPro and HIDE restorations shown in the main text. Figures 5 and 6 are from GoPro [10], and Figure 7 is from HIDE dataset [14].

8. Additional Results

GoPro dataset. In Figures 8–12 we present additional results on the GoPro dataset [10] where we compare our diffusion deblurring method to SAPHNet [15], DeblurGAN-v2 [7], MIMO-Unet+ [4], MPRNet [16], and HINet [2]. Consistent with the main text, "Ours-SA" refers to the sample averaging variant of our method.

DIV2K Deblurring dataset. In Figures 13–16 we present additional results on the synthetically generated DIV2K deblurring dataset. For comparison purposes, we train a regression-based model (to minimize L2 loss, thus maximizing PSNR) that has the same architecture as the one we used for the initial predictor. Compared to the over-smoothed restorations from the regression-based baseline trained to minimize distortion, our method produces more realistic textural details.

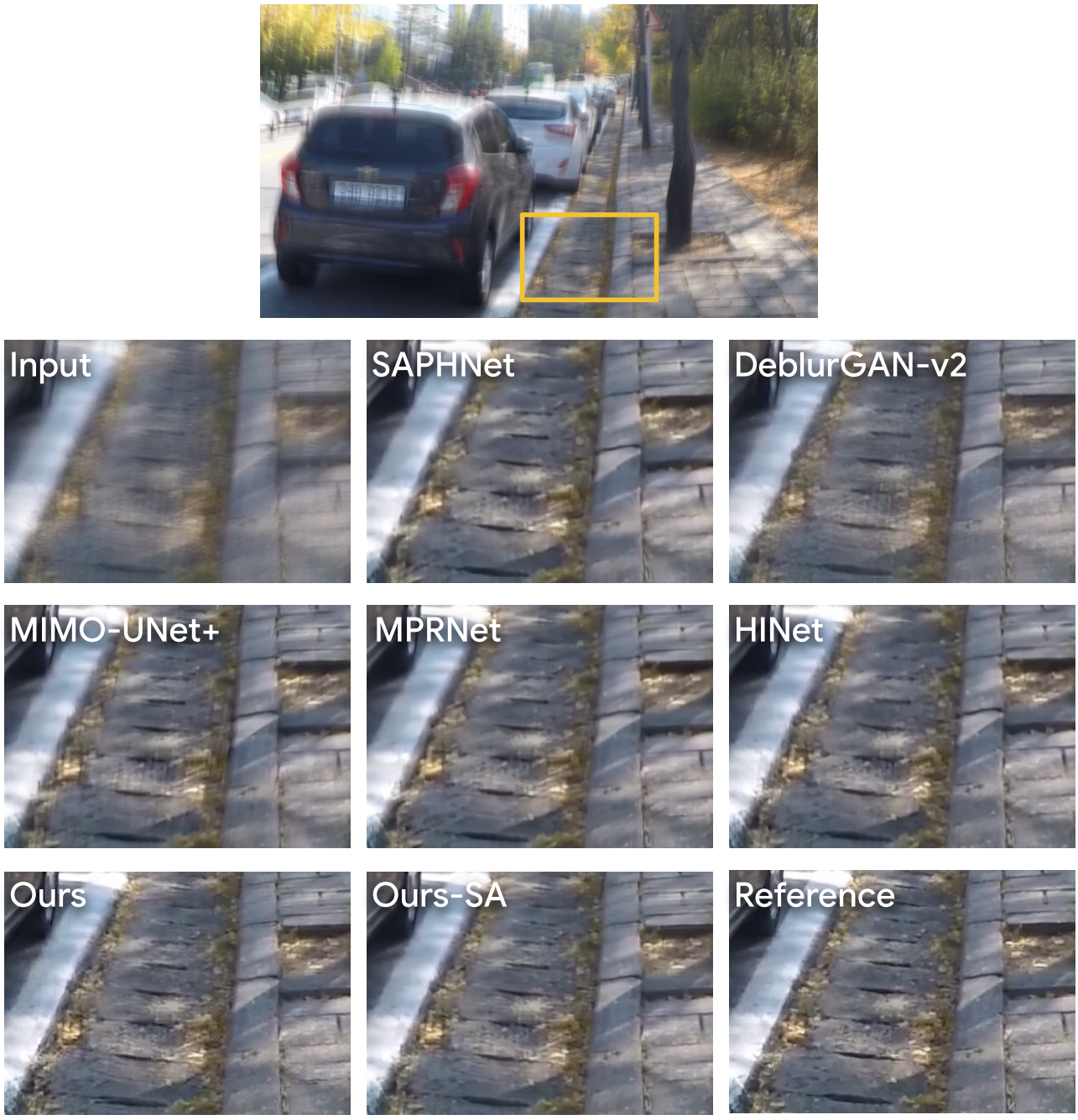

Figure 5. Full comparison of the GoPro [10] deblurring result presented in the main text. The compared methods are: SAPHNet [15], DeblurGAN-v2 [7], MIMO-Unet+ [4], MPRNet [16], and HINet [2]. We include restorations from our method with and without sampling averaging ("Ours" and "Ours-SA").

Figure 6. Full comparison of the GoPro [10] deblurring result presented in the main text. The compared methods are: SAPHNet [15], DeblurGAN-v2 [7], MIMO-Unet+ [4], MPRNet [16], and HINet [2]. We include restorations from our method with and without sampling averaging ("Ours" and "Ours-SA").

Figure 7. Full comparison of the HIDE [14] deblurring result presented in the main text. The compared methods are: SAPHNet [15], DeblurGAN-v2 [7], MIMO-Unet+ [4], MPRNet [16], and HINet [2]. We include restorations from our method with and without sampling averaging ("Ours" and "Ours-SA").

Figure 8. Additional deblurring results on the GoPro [10] dataset. The compared methods are: SAPHNet [15], DeblurGAN-v2 [7], MIMO-Unet+ [4], MPRNet [16], HINet [2], and our method with and without sampling averaging.

Figure 9. Additional deblurring results on the GoPro [10] dataset. The compared methods are: SAPHNet [15], DeblurGAN-v2 [7], MIMO-Unet+ [4], MPRNet [16], HINet [2], and our method with and without sampling averaging.

Figure 10. Additional deblurring results on the GoPro [10] dataset. The compared methods are: SAPHNet [15], DeblurGAN-v2 [7], MIMO-Unet+ [4], MPRNet [16], HINet [2], and our method with and without sampling averaging.

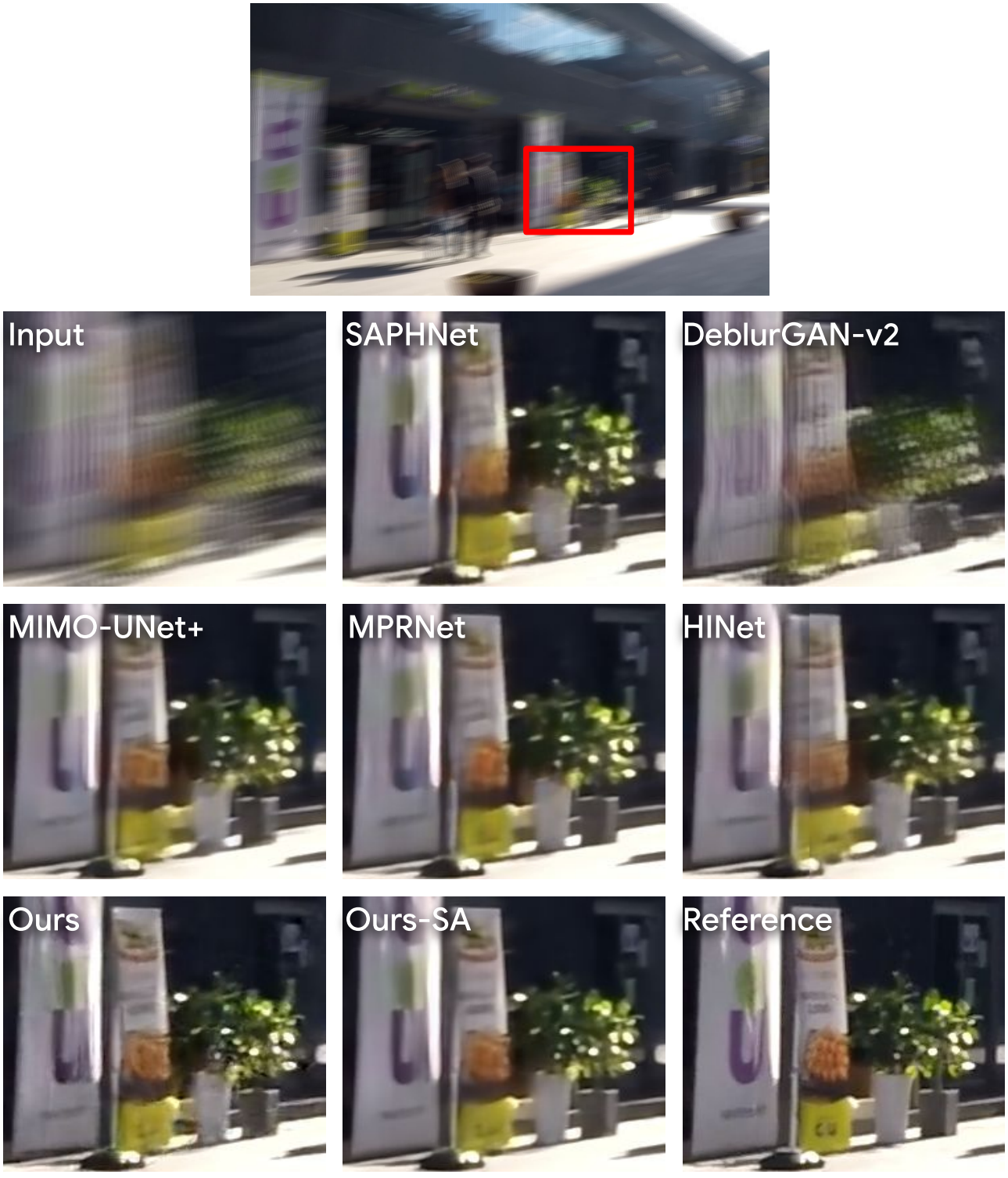

Figure 11. Additional deblurring results on the GoPro [10] dataset. The compared methods are: SAPHNet [15], DeblurGAN-v2 [7], MIMO-Unet+ [4], MPRNet [16], HINet [2], and our method with and without sampling averaging.

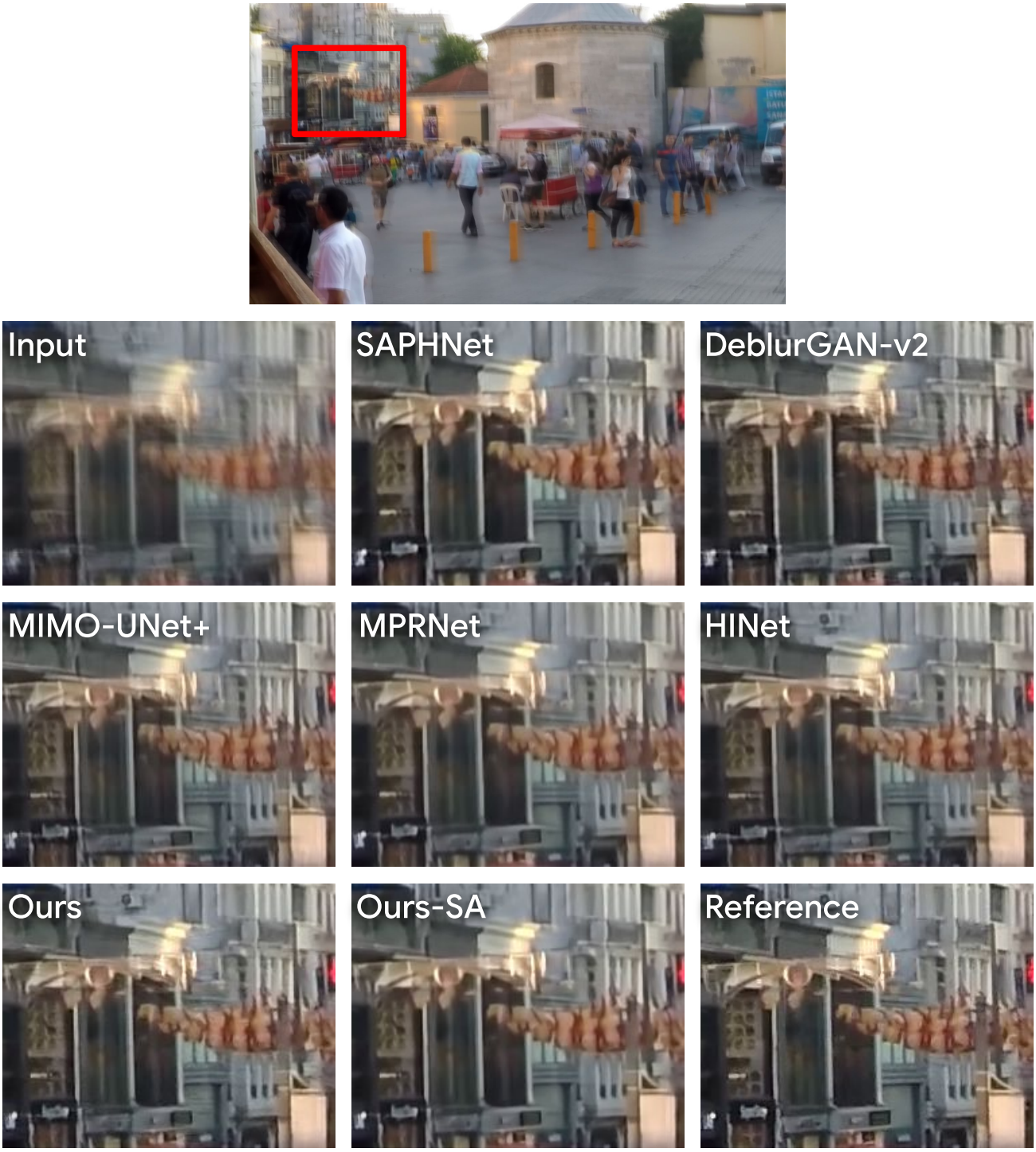

Figure 12. Additional deblurring results on the GoPro [10] dataset. The compared methods are: SAPHNet [15], DeblurGAN-v2 [7], MIMO-Unet+ [4], MPRNet [16], HINet [2], and our method with and without sampling averaging.

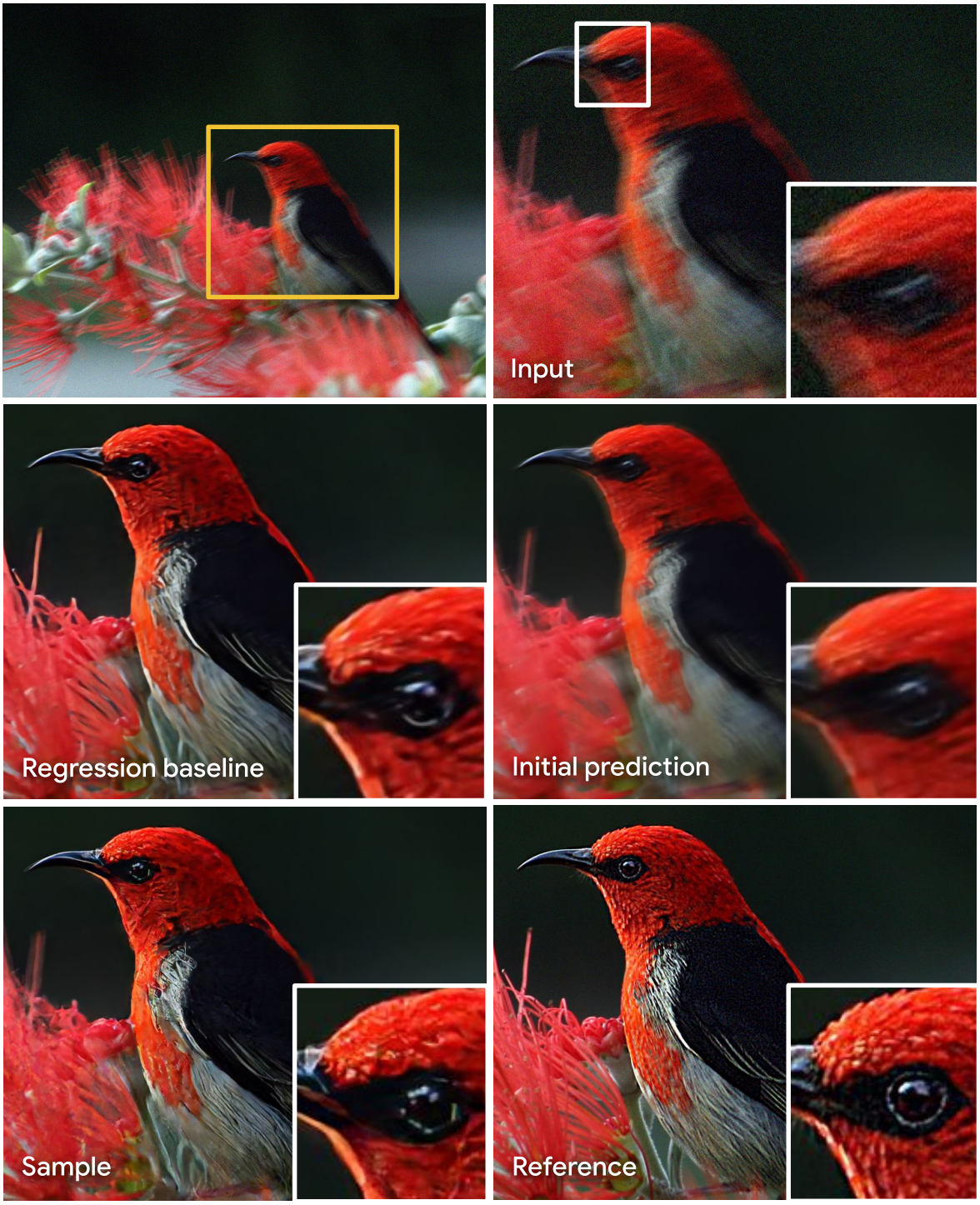

Figure 13. Additional deblurring results on the custom DIV2K dataset. We see that the initial predictor's blurry output is enhanced by the denoiser with realistic details.

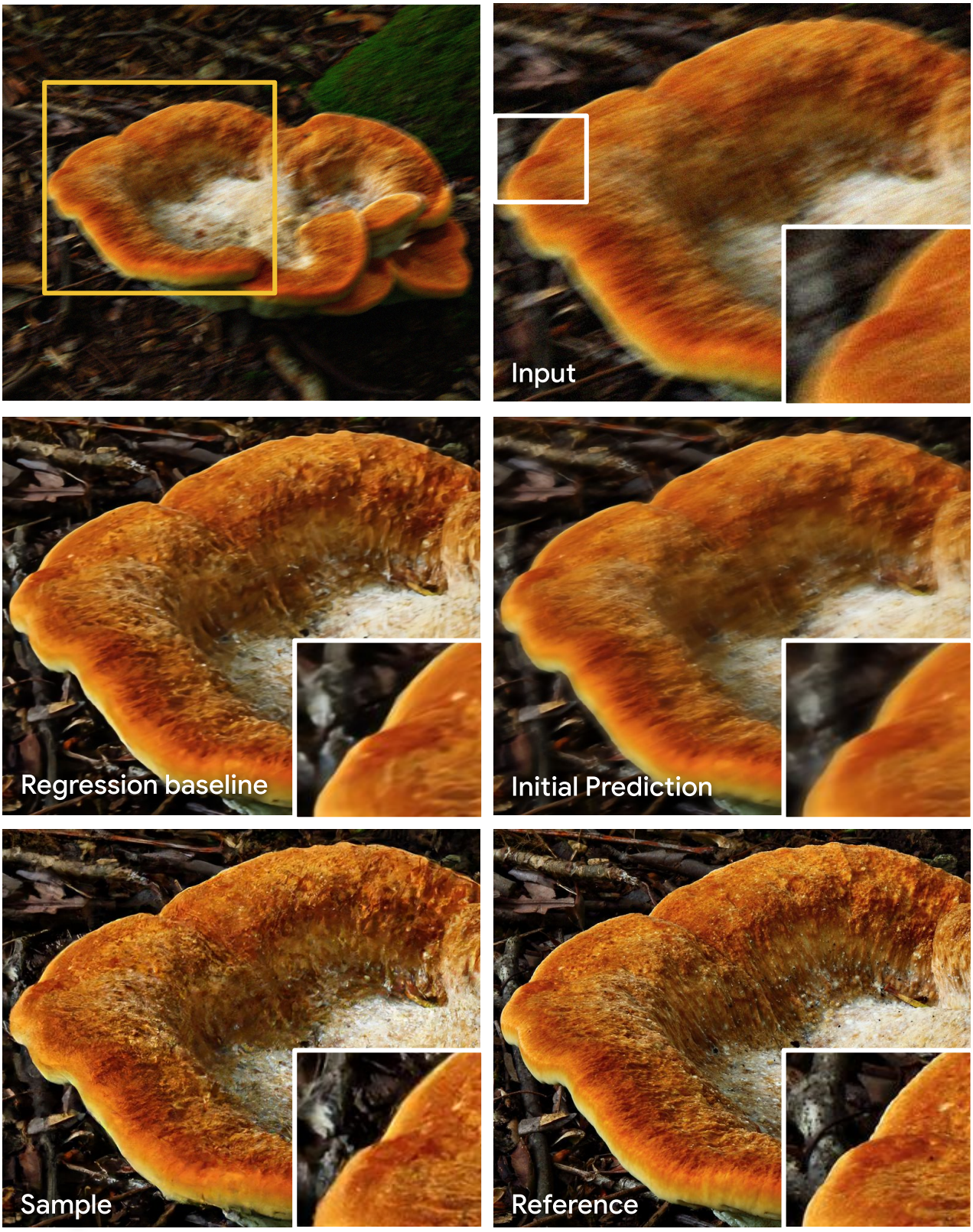

Figure 14. Additional deblurring results on the custom DIV2K dataset. We see that the initial predictor's blurry output is enhanced by the denoiser with realistic details.

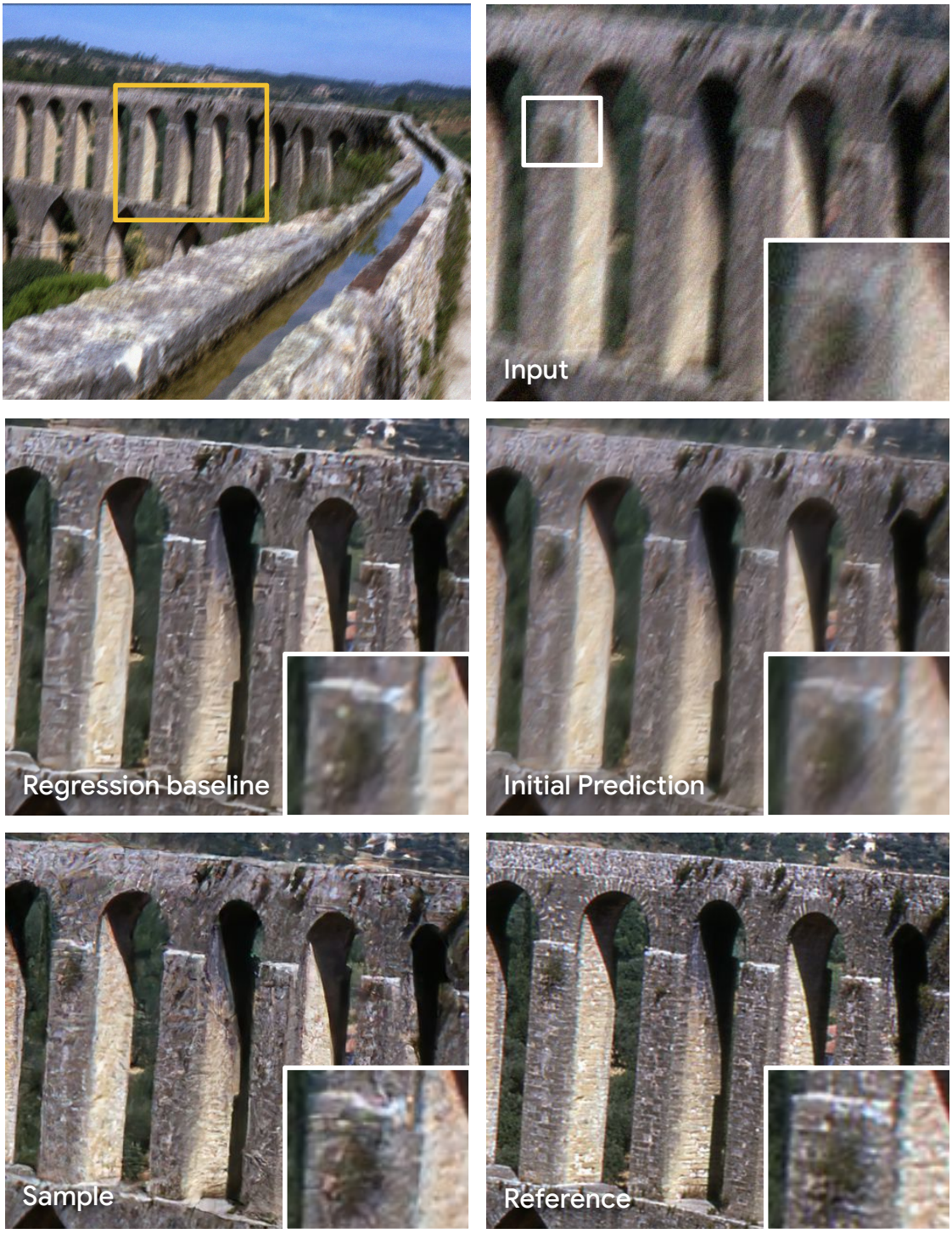

Figure 15. Additional deblurring results on the custom DIV2K dataset. We see that the initial predictor's blurry output is enhanced by the denoiser with realistic details.

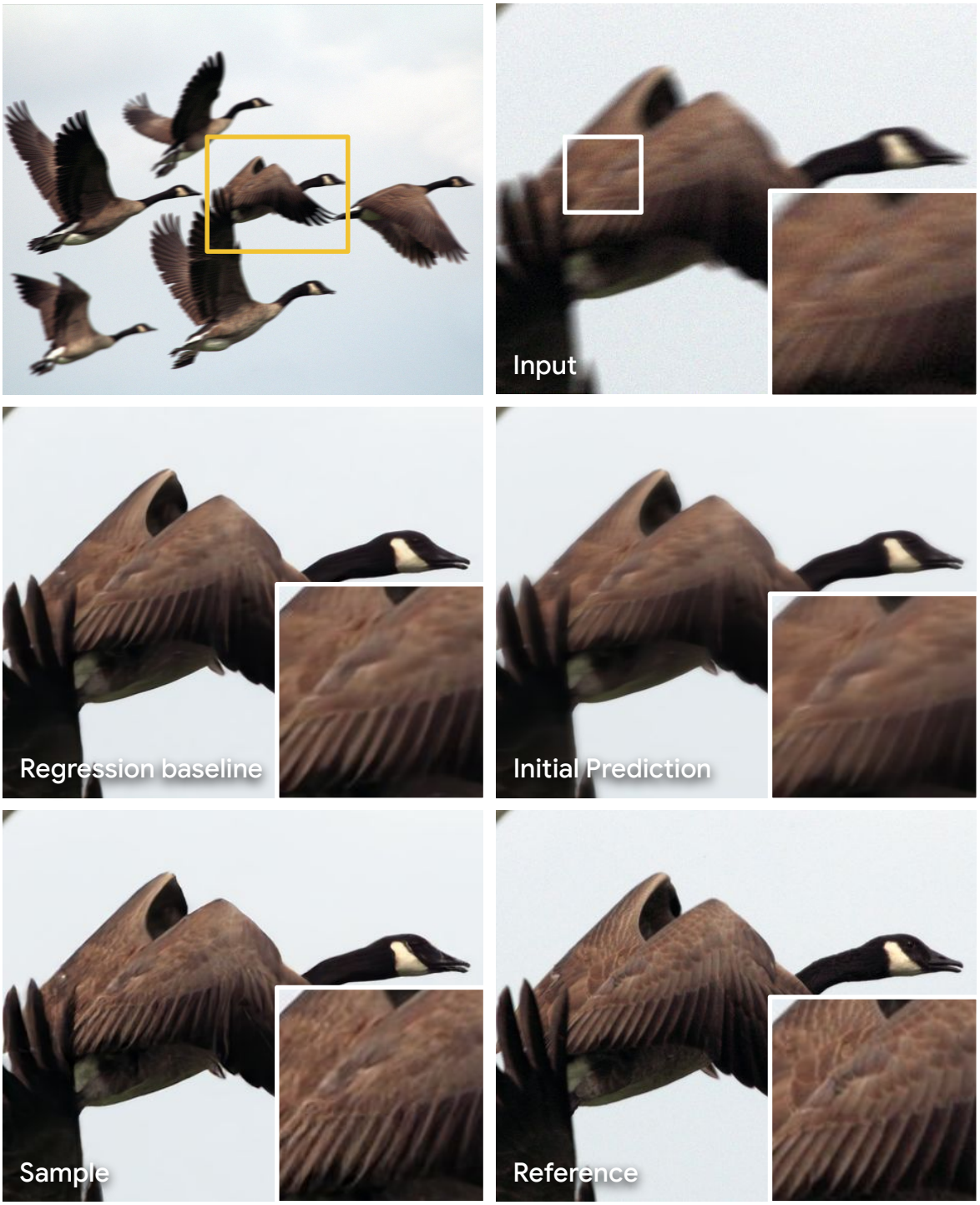

Figure 16. Additional deblurring results on the custom DIV2K dataset. We see that the initial predictor's blurry output is enhanced by the denoiser with realistic details.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 2

[2] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 182–192, June 2021. 5, 6, 7, 8, 9, 10, 11, 12, 13

[3] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2020. 3

[4] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4641–4650, October 2021. 5, 6, 7, 8, 9, 10, 11, 12, 13

[5] Mauricio Delbracio and Guillermo Sapiro. Removing camera shake via weighted fourier burst accumulation. *IEEE Transactions on Image Processing*, 24(11):3293–3307, 2015. 2, 3

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 3

[7] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 5, 6, 7, 8, 9, 10, 11, 12, 13

[8] Jichun Li, Weimin Tan, and Bo Yan. Perceptual variousness motion deblurring with light global context refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4116–4125, October 2021. 5

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 4

[10] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 5, 6, 7, 9, 10, 11, 12, 13

[11] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *European Conference on Computer Vision*, pages 184–201. Springer, 2020. 2

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[13] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*, 2021. 3

[14] Ziyi Shen, Wenguan Wang, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *IEEE International Conference on Computer Vision*, 2019. 5, 8

[15] Maitreya Suin, Kuldeep Purohit, and A. N. Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5, 6, 7, 8, 9, 10, 11, 12, 13

[16] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14821–14831, June 2021. 5, 6, 7, 8, 9, 10, 11, 12, 13