# AdaRevD: Adaptive Patch Exiting Reversible Decoder Pushes the Limit of Image Deblurring

## Supplementary Material

Table 1. Summary of four public datasets.

| Dataset | Types | Train | Test |
|---|---|---|---|
| GoPro [6] | synthetic | 2,103 | 1,111 |
| HIDE [8] | synthetic | - | 2,025 |
| RealBlur-R [7] | real-world | 3,758 | 980 |
| RealBlur-J [7] | real-world | 3,758 | 980 |

Table 2. Confusion Matrix of the Classifier on GoPro testset.

| gt \ pred | $\leq$ 20 dB | $\leq$ 25 dB | $\leq$ 30 dB | $\leq$ 35 dB | $\leq$ 40 dB | $>$ 40 dB | total |
|---|---|---|---|---|---|---|---|
| $\leq$ 20 dB | 394 | 61 | 0 | 0 | 0 | 0 | 455 |
| $\leq$ 25 dB | 38 | 2998 | 164 | 0 | 0 | 0 | 3200 |
| $\leq$ 30 dB | 0 | 141 | 3148 | 206 | 0 | 0 | 3495 |
| $\leq$ 35 dB | 0 | 0 | 101 | 1248 | 154 | 0 | 1503 |
| $\leq$ 40 dB | 0 | 0 | 0 | 22 | 192 | 3 | 217 |
| $>$ 40 dB | 0 | 0 | 0 | 0 | 13 | 5 | 18 |
| total | 432 | 3200 | 3413 | 1476 | 359 | 8 | 8888 |

## 1. Details of Experiment

**Dataset** As shown in Sec. 4.1 in the main paper, we evaluate our method on the four datasets shown in Table 1 and report two groups of results:

- $\mathcal{A}$. train on GoPro, test on GoPro [6] / HIDE [8] / RealBlur-R / RealBlur-J [7];
- $\mathcal{B}$. train and test on RealBlur-J / RealBlur-R respectively;

**Dataset Distribution** Figs. 1 - 3 shows the distribution of the blur patches from different dataset. As indicated in Fig. 1 and Fig. 2, the degraded patches from GoPro, HIDE and RealBlur-J dataset are almost all fall within the range of [15 dB, 45 dB]. Thus, we group the patches to 6 degradation degrees: ($\leq$ 20 dB, $\tilde{c} = 1$), ($\leq$ 25 dB, $\tilde{c} = 2$), ($\leq$ 30 dB, $\tilde{c} = 3$), ($\leq$ 35 dB, $\tilde{c} = 4$), ($\leq$ 40 dB, $\tilde{c} = 5$) and ($>$ 40 dB, $\tilde{c} = 6$) for the classifier training. Different from other dataset, the degraded patches from RealBlur-R are almost all fall in the range of [15 dB, 55 dB] (shown in Fig. 3). Thus, we group the patches from RealBlur-R to 8 degradation degrees: ($\leq$ 20 dB, $\tilde{c} = 1$), ($\leq$ 25 dB, $\tilde{c} = 2$), ($\leq$ 30 dB, $\tilde{c} = 3$), ($\leq$ 35 dB, $\tilde{c} = 4$), ($\leq$ 40 dB, $\tilde{c} = 5$), ($\leq$ 45 dB, $\tilde{c} = 6$), ($\leq$ 50 dB, $\tilde{c} = 7$) and ($>$ 50 dB, $\tilde{c} = 8$) for the classifier training.

**Clustering Criteria** We split the image patches into different classes according to PSNR, which is a direct and efficient measure of degradation degree. We conduct experiments: ① In paper, we apply a step ($\gamma$) of 5 dB to cluster the blur patches into 6 degradation degrees from 15 dB to 45 dB. Here we change $\gamma$ within the range of $[3, 4, 5, 6, 10]$, and obtain almost the same PSNRs (34.50 dB). Classification accuracies and utilization rates of the sub-decoders (D-rates) are (83.7%, 87.2%), (87.0%, 86.0%), (89.8%, 84.3%), (91.4%, 87.7%), and (94.6%, 84.8%) respectively. First, although the larger step acquires better classification accuracy, AdaRevD-B (4 sub-decoders) has a big tolerance for accuracy corresponding to a small $\gamma$ (*e.g.*, 9 classes when $\gamma = 3$). Second, it is observed that almost all the misclassified patches are classified to the adjacent degradation degree, shown in Table 2 ($\gamma = 5$). Only a few patches would exit at earlier sub-decoder (slightly reduce PSNR of the whole image), while a few exit at later sub-decoder (slightly increase PSNR), which have certain complementary effects on the final PSNR. ② Following ClassSR [4], we separate PSNRs into 6 classes with the **same numbers** of blur patches, the PSNR is also the same (34.50 dB), even with lower classification accuracy (85.1%). Thus, AdaRevD does not demand a very high classification accuracy, and it is acceptable that a small number of patches are classified to adjacent degradation degree.

**Evaluation Metric** The computational complexity of MACs (G) and the number of parameters (M) are reported in Table 3. Table 3 illustrates that our method can further explore the well-trained NAFNet's [1] insufficient decoding capability (33.69 dB) to a higher level (34.10 dB), which is similar to UFPNet [3] (34.06 dB 243 G), but with fewer MACs (168 G).

**Early-exit Signal** In AdaRevD, early-exit signal $E_c$ is determined by $\mathbf{O}_c^j$ and $\tau$. The $\mathbf{O}_c^j$ of RevD-B on GoPro, RealBlur-J and RealBlur-R datasets are summarized in Tables 4, 5 and 6. Furthermore, the $\mathbf{O}_c^j$ of RevD-L on these three datasets are shown in Tables 7, 8 and 9. The first $\mathbf{O}_c^{j-1}$ where its next $\mathbf{O}_c^j$ is smaller than $\tau = 0.05$ (the patch exit in the $((j-1)$th sub-decoder) is highlighted in the tables.

As illustrated in these tables, blur patches with varying degradation degrees exhibit distinct improvements in PSNR within the same sub-decoder. The higher the PSNR, the less restoration the patch undergoes in the identical sub-decoder. As more sub-decoders are progressively stacked,

Table 3. The comparison involves the computational complexity of MACs (G) and the number of parameters (M), when the input size is $256 \times 256$. PSNR (dB) is calculated on GoPro test set.

| Method | MIMO-UNet++ [2] | DeepRFT+ [5] | Restormer [10] | NAFNet64 [1] | UFPNet [3] | RevD-B(NAFNet) | RevD-B(UFPNet) | RevD-L(UFPNet) |
|---|---|---|---|---|---|---|---|---|
| MACs (G) | 617 | 187 | 141 | 64 | 243 | 168 | 347 | 460 |
| Params (M) | 16.1 | 19.5 | 26.1 | 65.0 | 80.3 | 131.0 | 142.5 | 210.8 |
| PSNR (dB) | 32.68 | 33.52 | 32.92 | 33.69 | 34.06 | 34.10 | 34.51 | 34.64 |


Figure 1. Distribution of GoPro [6] and HIDE [8] Dataset. (a) The ranked PSNR curve of the image patches from GoPro train set; (b) The ranked PSNR curve of the image patches from GoPro test set; (c) The ranked PSNR curve of the image patches from HIDE test set.

Table 4. Improvement of diffent sub-decoders in RevD-B on GoPro dataset. The value in the $c$th row and the $j$th column is $\mathbf{O}_c^j$. The first $\mathbf{O}_c^{j-1}$ that $\mathbf{O}_c^j$ smaller than $\tau = 0.05$ is highlighted in the table.

| | TrainSet | | | | TestSet | | | |
|---|---|---|---|---|---|---|---|---|
| Degree | dec1 | dec2 | dec3 | dec4 | dec1 | dec2 | dec3 | dec4 |
| ≤20 | 11.134 | 0.642 | 0.351 | 0.178 | 10.275 | 0.653 | 0.383 | 0.170 |
| 20-25 | 10.959 | 0.406 | 0.211 | 0.100 | 9.622 | 0.355 | 0.208 | 0.093 |
| 25-30 | 9.184 | 0.214 | 0.105 | 0.047 | 8.097 | 0.191 | 0.100 | 0.045 |
| 30-35 | 6.215 | 0.121 | 0.050 | 0.021 | 5.397 | 0.103 | 0.050 | 0.021 |
| 35-40 | 3.468 | 0.079 | 0.024 | 0.011 | 2.859 | 0.073 | 0.014 | 0.010 |
| >40 | 2.380 | 0.047 | 0.016 | 0.009 | 1.510 | 0.022 | 0.006 | 0.004 |

Table 5. Improvement of diffent sub-decoders in RevD-B on RealBlur-J dataset.

| | TrainSet | | | | TestSet | | | |
|---|---|---|---|---|---|---|---|---|
| Degree | dec1 | dec2 | dec3 | dec4 | dec1 | dec2 | dec3 | dec4 |
| ≤20 | 11.855 | 0.758 | 0.441 | 0.201 | 5.223 | 0.093 | 0.089 | 0.041 |
| 20-25 | 11.236 | 0.563 | 0.336 | 0.139 | 4.842 | 0.138 | 0.092 | 0.041 |
| 25-30 | 10.041 | 0.390 | 0.214 | 0.080 | 4.138 | 0.107 | 0.066 | 0.024 |
| 30-35 | 8.406 | 0.249 | 0.115 | 0.042 | 3.461 | 0.067 | 0.036 | 0.010 |
| 35-40 | 7.101 | 0.168 | 0.064 | 0.030 | 2.537 | 0.073 | 0.034 | 0.007 |
| >40 | 5.525 | 0.126 | 0.044 | 0.028 | 1.380 | -0.030 | 0.000 | 0.000 |

the model's capacity to recover images reaches saturation. Tables 4 and 7 demonstrate that the $E_c$ remains consistent between the training set and test set when $\tau = 0.05$. Moreover, the performance of the various sud-decoders on the train and test set in the tables indicate that selecting the early-exit signal $E_c$ based on the train set ensures effective recovery of patches from the test set. In essence, opting for $E_c$ from the train set is rational, as the sub-decoder saturation observed in the train set aligns with the saturation observed in the test set.

## 2. Viasulizations

The visual results for GoPro [6], HIDE [8], RealBlur-R [7] and RealBlur-J [7] are presented in Figs. 4, 5, 6 and 7, respectively. The visualizations depicted in Fig. 4 and Fig. 5 illustrate AdaRevD's capability to restore sharper images. We also show the visualization results on the RealBlur [7] dataset in Fig. 6 and Fig. 7. It can be observed that our model yields more visually pleasant outputs than other methods on both synthetic and real-world motion deblurring. This is evident when compared to other SOTA methods, such as DeepRFT [5] and UFPNet [3].

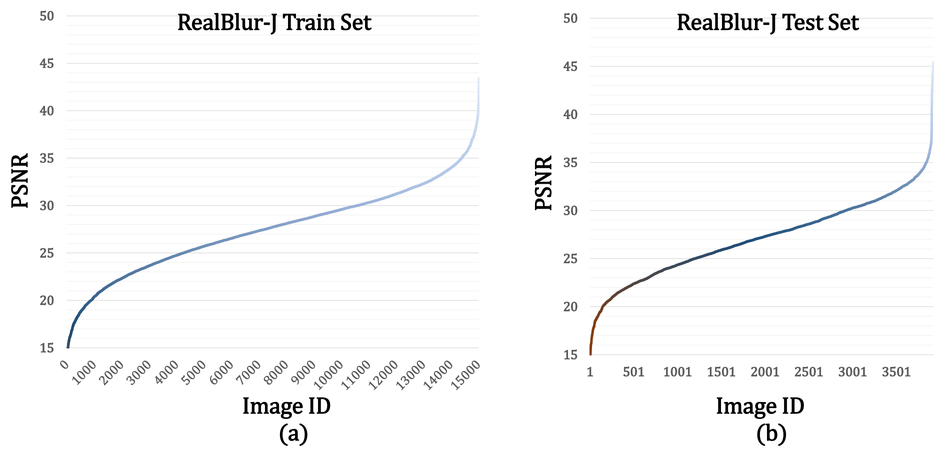



Figure 2. Distribution of RealBlur-J [7] Dataset. (a) The ranked PSNR curve of the image patches from RealBlur-J train set; (b) The ranked PSNR curve of the image patches from RealBlur-J test set.

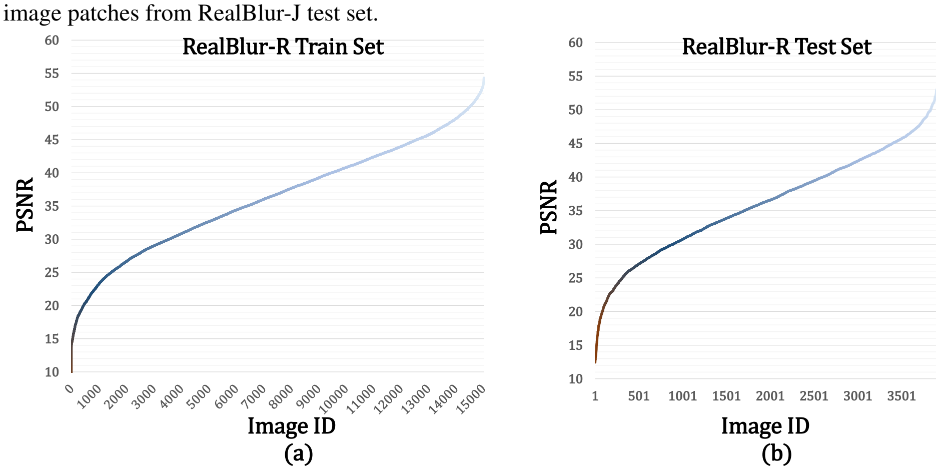



Figure 3. Distribution of RealBlur-R [7] Dataset. (a) The ranked PSNR curve of the image patches from RealBlur-R train set; (b) The ranked PSNR curve of the image patches from RealBlur-R test set.

Table 6. Improvement of diffent sub-decoders in RevD-B on RealBlur-R dataset.

| | **TrainSet** | | | | **TestSet** | | | |
|---|---|---|---|---|---|---|---|---|
| **Degree** | **dec1** | **dec2** | **dec3** | **dec4** | **dec1** | **dec2** | **dec3** | **dec4** |
| ≤20 | 12.401 | 0.863 | 0.308 | 0.098 | 5.541 | 0.152 | 0.063 | 0.029 |
| 20-25 | 12.356 | 0.781 | 0.272 | 0.081 | 5.305 | 0.182 | 0.061 | 0.022 |
| 25-30 | 12.268 | 0.702 | 0.215 | 0.060 | 5.657 | 0.155 | 0.056 | 0.017 |
| 30-35 | 11.593 | 0.588 | 0.178 | 0.0513 | 4.233 | 0.139 | 0.039 | 0.012 |
| 35-40 | 9.681 | 0.390 | 0.115 | 0.042 | 3.431 | 0.133 | 0.037 | 0.016 |
| 40-45 | 7.423 | 0.245 | 0.0778 | 0.032 | 3.043 | 0.111 | 0.028 | 0.012 |
| 45-50 | 5.210 | 0.136 | 0.042 | 0.021 | 2.005 | 0.042 | 0.012 | 0.008 |
| >50 | 3.227 | 0.074 | 0.018 | 0.009 | 1.102 | 0.014 | 0.006 | 0.006 |

Table 7. Improvement of diffent sub-decoders in RevD-L on GoPro dataset.

| | TrainSet | | | | | | | | TestSet | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Degree** | **dec1** | **dec2** | **dec3** | **dec4** | **dec5** | **dec6** | **dec7** | **dec8** | **dec1** | **dec2** | **dec3** | **dec4** | **dec5** | **dec6** | **dec7** | **dec8** |
| ≤20 | 11.068 | 0.601 | 0.257 | 0.222 | 0.198 | 0.109 | 0.047 | 0.014 | 10.201 | 0.610 | 0.252 | 0.269 | 0.198 | 0.107 | 0.048 | 0.015 |
| 20-25 | 10.903 | 0.400 | 0.154 | 0.135 | 0.118 | 0.064 | 0.026 | 0.007 | 9.569 | 0.345 | 0.129 | 0.153 | 0.122 | 0.055 | 0.021 | 0.007 |
| 25-30 | 9.141 | 0.227 | 0.076 | 0.070 | 0.060 | 0.031 | 0.012 | 0.003 | 8.053 | 0.206 | 0.065 | 0.073 | 0.064 | 0.030 | 0.009 | 0.003 |
| 30-35 | 6.176 | 0.145 | 0.037 | 0.035 | 0.029 | 0.015 | 0.007 | 0.002 | 5.362 | 0.125 | 0.030 | 0.041 | 0.031 | 0.015 | 0.005 | 0.002 |
| 35-40 | 3.437 | 0.101 | 0.019 | 0.018 | 0.014 | 0.007 | 0.004 | 0.001 | 2.798 | 0.116 | 0.012 | 0.023 | 0.016 | 0.010 | 0.003 | 0.001 |
| >40 | 2.365 | 0.057 | 0.016 | 0.010 | 0.007 | 0.007 | 0.008 | 0.002 | 1.470 | 0.067 | 0.013 | 0.018 | 0.014 | 0.006 | 0.003 | 0.001 |

Table 8. Improvement of diffent sub-decoders in RevD-L on RealBlur-J dataset.

| | TrainSet | | | | | | | | TestSet | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Degree** | **dec1** | **dec2** | **dec3** | **dec4** | **dec5** | **dec6** | **dec7** | **dec8** | **dec1** | **dec2** | **dec3** | **dec4** | **dec5** | **dec6** | **dec7** | **dec8** |
| ≤20 | 11.718 | 0.788 | 0.429 | 0.227 | 0.221 | 0.131 | 0.101 | 0.006 | 5.131 | 0.074 | 0.065 | 0.052 | 0.056 | 0.032 | 0.022 | 0.002 |
| 20-25 | 11.113 | 0.606 | 0.330 | 0.190 | 0.161 | 0.082 | 0.066 | 0.004 | 4.813 | 0.125 | 0.092 | 0.050 | 0.056 | 0.033 | 0.024 | 0.002 |
| 25-30 | 9.956 | 0.419 | 0.218 | 0.133 | 0.096 | 0.045 | 0.039 | 0.003 | 4.119 | 0.089 | 0.067 | 0.045 | 0.037 | 0.019 | 0.014 | 0.001 |
| 30-35 | 8.348 | 0.263 | 0.129 | 0.082 | 0.050 | 0.022 | 0.025 | 0.002 | 3.486 | 0.066 | 0.038 | 0.028 | 0.021 | 0.009 | 0.010 | 0.001 |
| 35-40 | 7.040 | 0.186 | 0.091 | 0.048 | 0.029 | 0.0159 | 0.019 | 0.002 | 2.444 | 0.052 | 0.050 | 0.032 | 0.013 | 0.004 | 0.006 | 0.000 |
| >40 | 5.444 | 0.158 | 0.081 | 0.039 | 0.022 | 0.014 | 0.018 | 0.001 | 1.462 | -0.021 | -0.018 | 0.026 | -0.021 | 0.007 | 0.003 | -0.003 |

Table 9. Improvement of diffent sub-decoders in RevD-L on RealBlur-R dataset.

| | TrainSet | | | | | | | | TestSet | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Degree** | **dec1** | **dec2** | **dec3** | **dec4** | **dec5** | **dec6** | **dec7** | **dec8** | **dec1** | **dec2** | **dec3** | **dec4** | **dec5** | **dec6** | **dec7** | **dec8** |
| ≤20 | 12.037 | 1.174 | 0.357 | 0.105 | 0.188 | 0.117 | 0.029 | 0.000 | 5.524 | 0.178 | 0.062 | 0.033 | 0.062 | 0.032 | 0.010 | 0.000 |
| 20-25 | 11.988 | 1.102 | 0.314 | 0.098 | 0.174 | 0.098 | 0.024 | 0.000 | 5.252 | 0.227 | 0.062 | 0.037 | 0.064 | 0.023 | 0.007 | 0.000 |
| 25-30 | 11.895 | 1.030 | 0.259 | 0.079 | 0.132 | 0.070 | 0.019 | 0.000 | 5.611 | 0.214 | 0.062 | 0.036 | 0.046 | 0.021 | 0.006 | 0.000 |
| 30-35 | 11.260 | 0.891 | 0.215 | 0.060 | 0.094 | 0.055 | 0.017 | 0.000 | 4.190 | 0.190 | 0.041 | 0.023 | 0.032 | 0.016 | 0.005 | 0.000 |
| 35-40 | 9.419 | 0.640 | 0.136 | 0.035 | 0.055 | 0.038 | 0.013 | 0.000 | 3.372 | 0.189 | 0.037 | 0.019 | 0.026 | 0.017 | 0.006 | 0.000 |
| 40-45 | 7.243 | 0.424 | 0.085 | 0.022 | 0.030 | 0.027 | 0.009 | 0.000 | 3.016 | 0.141 | 0.031 | 0.016 | 0.018 | 0.016 | 0.004 | 0.000 |
| 45-50 | 5.093 | 0.259 | 0.044 | 0.013 | 0.014 | 0.015 | 0.005 | 0.000 | 1.991 | 0.058 | 0.020 | 0.007 | 0.006 | 0.008 | 0.003 | 0.000 |
| >50 | 3.139 | 0.183 | 0.020 | 0.008 | 0.006 | 0.007 | 0.003 | 0.000 | 1.128 | -0.014 | 0.010 | 0.007 | 0.003 | 0.008 | 0.002 | 0.000 |


Figure 4. Examples on the GoPro test dataset. The odd rows show blur image, predicted images of different methods, and ground-truth sharp image. The even rows show the residual of the blur image / predicted sharp images and the ground-truth sharp image.

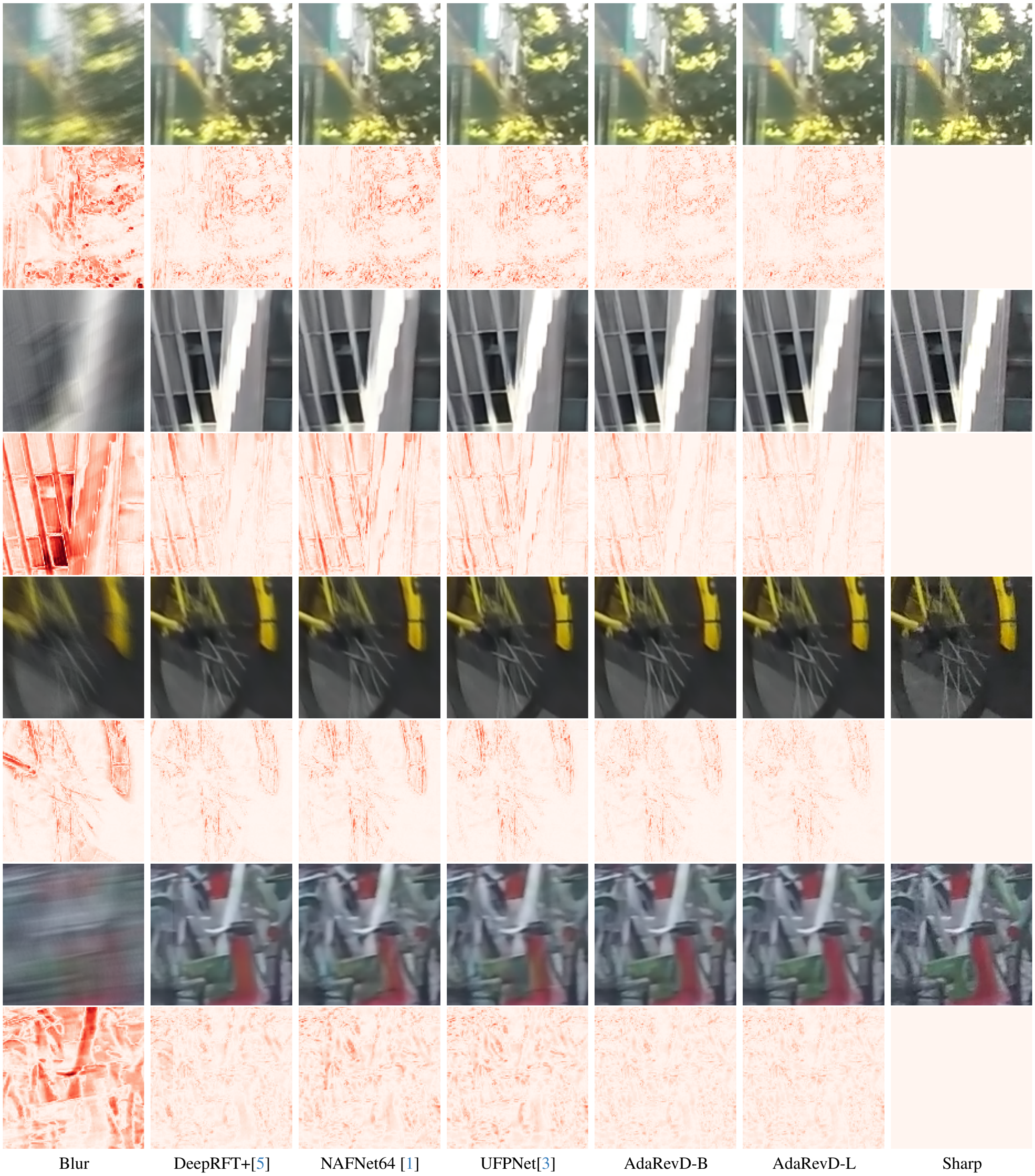
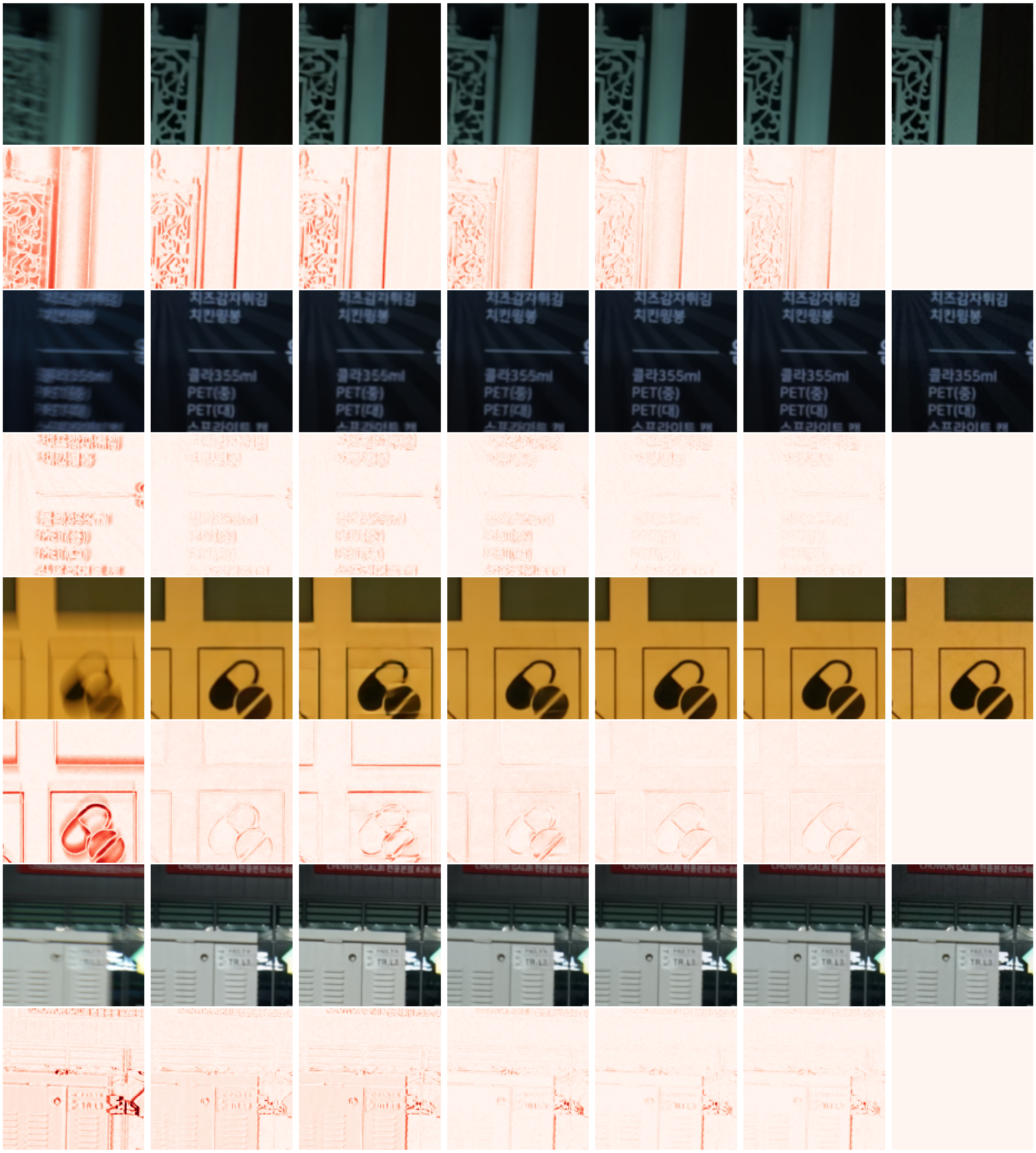



Figure 5. Examples on the HIDE test dataset.




Blur | DeepRFT+ [5] | Stripformer [9] | UFPNet [3] | AdaRevD-B | AdaRevD-L | Sharp

Figure 6. Examples on the RealBlur-J test dataset.

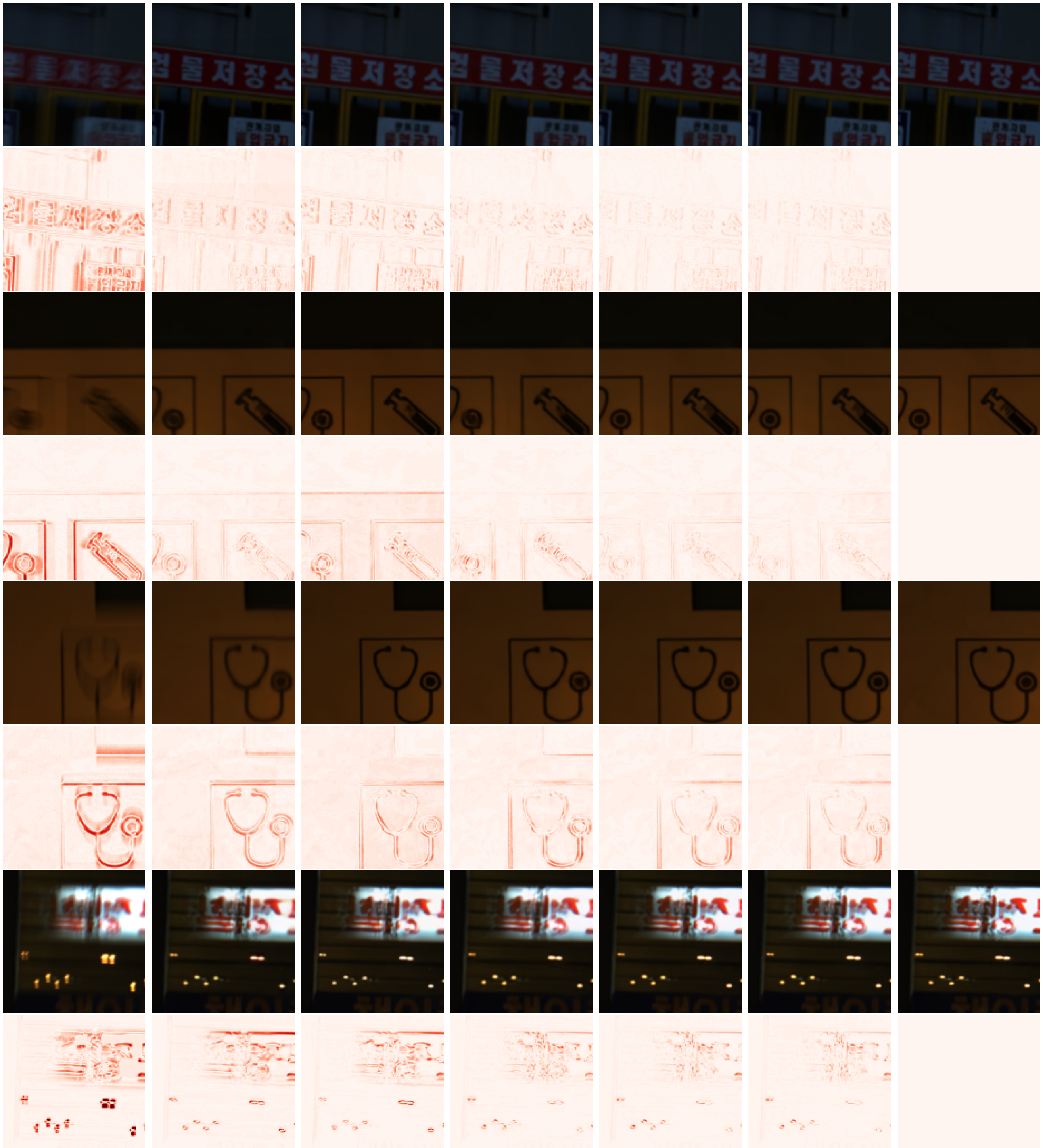



Figure 7. Examples on the RealBlur-R test dataset.

## References


[1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *Proc. ECCV*, 2022. 1, 2, 5, 6

[2] 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 *Proc. ICCV*, 2021. 2

[3] Zhenxuan Fang, Fangfang Wu, Weisheng Dong, Xin Li, Jinjian Wu, and Guangming Shi. Self-supervised non-uniform kernel estimation with flow-based motion prior for blind image deblurring. In *Proc. CVPR*, 2023. 1, 2, 5, 6, 7, 8

[4] Xiangtao Kong, Hengyuan Zhao, Yu Qiao, and Chao Dong. Classsr: A general framework to accelerate super-resolution networks by data characteristic. In *Proc. CVPR*, 2021. 1

[5] Xintian Mao, Yiming Liu, Fengze Liu, Qingli Li, Wei Shen, and Yan Wang. Intriguing findings of frequency selection for image deblurring. In *Proc. AAAI*, 2023. 2, 5, 6, 7, 8

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

[7] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *Proc. ECCV*, 2020. 1, 2, 3

[8] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *Proc. ICCV*, 2019. 1, 2

[9] 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. 7, 8

[10] Syed Waqas Zamir, Aditya Arora, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proc. CVPR*, 2022. 2