

F2former: When Fractional Fourier Meets Deep Wiener Deconvolution and Selective Frequency Transformer for Image Deblurring

Supplementary Material

A Implementation details

A.1 Loss function

For training F2former, we adopt L1 loss in both spatial and FFT domain similar to [2, 5]. We also include L1 loss in FRFT domain to ensure intactness of the spatial-frequency information. Hence, the overall objective function to optimize our model is defined as,

$$\mathcal{L}_{total} = \mathcal{L}_s + \lambda_{t_1} \mathcal{L}_{t_1} + \lambda_{t_\alpha} \mathcal{L}_{t_\alpha}, \quad (1)$$

where, \mathcal{L}_s , \mathcal{L}_{t_1} , and \mathcal{L}_{t_α} correspond to loss in spatial domain, FFT domain and FRFT domain with parameter α , respectively. From Figure 3 (a) of main paper, they can be defined as,

$$\begin{aligned} \mathcal{L}_s &= \sum_{p=1}^3 \frac{1}{N_p} \|\hat{\mathbf{Y}}_p - \mathbf{Y}_p\|_1, \\ \mathcal{L}_{t_1} &= \sum_{p=1}^3 \frac{1}{N_p} \|\mathcal{F}(\hat{\mathbf{Y}}_p) - \mathcal{F}(\mathbf{Y}_p)\|_1, \\ \mathcal{L}_{t_\alpha} &= \sum_{p=1}^3 \frac{1}{N_p} \|\mathcal{F}_\alpha(\hat{\mathbf{Y}}_p) - \mathcal{F}_\alpha(\mathbf{Y}_p)\|_1, \end{aligned} \quad (2)$$

where, \mathbf{Y}_p is the scaled GT at p -th scale, N_p is number of elements for normalization at p -th scale, $\mathcal{F}(\cdot)$ is FFT operation and $\mathcal{F}_\alpha(\cdot)$ is FRFT operation with parameter α . λ_{t_1} and λ_{t_α} are weights corresponding to \mathcal{L}_{t_1} and \mathcal{L}_{t_α} , respectively. We empirically decide $\mathcal{L}_{t_1} = \mathcal{L}_{t_\alpha} = 0.1$ and $\alpha = 0.5$.

A.2 Parameter settings for training F2former

We adopt the training strategy of NAFNet [1] and FFTformer [4]. Specifically, adopting their data augmentation strategy and Adam optimizer with default values. Initial learning rate is set at 1×10^{-3} and is updated with cosine annealing upto 600,000 iterations. The minimum learning rate is set at 1×10^{-7} . F2former is trained with input images of shape $256 \times 256 \times 3$ with a batch size of 8. For computing self-attention we set the patch size as 8×8 . Other parameters like α for FRFT computation in FHTB block and λ as the cut-off frequency for cosine-bell function in FM-FFN are set as learnable parameter and are updated during training based on minimizing the loss function. The corresponding details are discussed later in ablation section. All the experiments are carried out in a single 80GB Nvidia A100 GPU and according to that the training configuration in NAFNet has been modified.

A.3 Kernel estimation block (KEB)

For a given blurry input \mathbf{X} , KEB estimates its corresponding blurry kernel \mathbf{K} . Its detail overflow is shown in Figure 1. Pri-

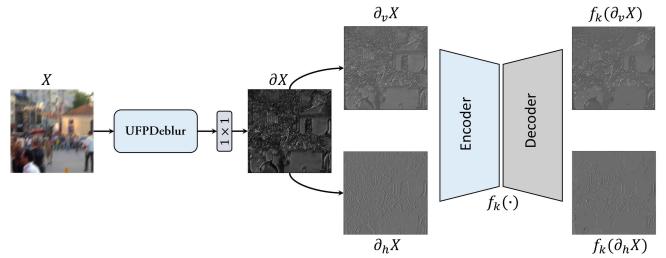


Figure 1: Workflow of our KEB module.

marily, we use kernel estimation network from [3] where for each pixel the blur kernel is estimated of shape 19×19 . However, this kernel can not be used directly for Wiener deconvolution as it will increase the overall computation cost. Therefore, upon reshaping this estimated kernel, it turns out the gradient map ($\partial\mathbf{X}$) while also highlighting the blur pattern. To estimate an accurate blur kernel from $\partial\mathbf{X}$, we leverage the blur kernel estimation method in [6]. Basically, we estimate gradient at the vertical ($\partial_v\mathbf{X}$) and horizontal ($\partial_h\mathbf{X}$) direction from $\partial\mathbf{X}$ which is later passed to a 6-layer auto-encoder as shown in Figure 1. The encoder part removes unnecessary details of image gradients while the decoder extracts the major structural details with enhanced edges formulated as $\partial_v\mathbf{X}_e = f_K(\partial_v\mathbf{X})$ and $\partial_h\mathbf{X}_e = f_K(\partial_h\mathbf{X})$, where $f_K(\cdot)$ is auto-encoder operation as shown in above Figure. The design of auto-encoder as well as its training is carried out according to the respective details in [6].

After obtaining the salient edges as $\partial\mathbf{X}_e = f_K(\partial\mathbf{X})$ for blurry input \mathbf{X} , the blur kernel (\mathbf{K}) can be estimated from the following optimization problems,

$$\begin{aligned} \mathbf{K} &= \arg \min_{\mathbf{K}} \|\mathbf{K} \cdot \partial\mathbf{X}_e - \partial\mathbf{X}\|_2^2 + \eta \|\mathbf{K}\|_2^2, \\ \mathbf{X}_l &= \arg \min_{\mathbf{X}_l} \|\mathbf{K} \cdot \mathbf{X}_l - \mathbf{X}\|_2^2 + \gamma \|\mathbf{X}_l\|_2^2, \end{aligned} \quad (3)$$

where, η and γ are parameters for regularization terms, and \mathbf{X}_l is intermediate latent image. Similar to [6], we use \mathbf{X}_l to update the edge information in $\partial\mathbf{X}_e$. From [6], the closed form solution of the first optimization problem in equation 3 can be written in continuous domain as,

$$K = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(\partial_h X_e)} \mathcal{F}(\partial_h X) + \overline{\mathcal{F}(\partial_v X_e)} \mathcal{F}(\partial_v X)}{\mathcal{F}(\partial_h X_e)^2 + \mathcal{F}(\partial_v X_e)^2 + \eta} \right), \quad (4)$$

where $\mathcal{F}^{-1}(\cdot)$ is inverse FFT and $\overline{\mathcal{F}(\cdot)}$ is complex conjugate operation FFT domain. Then the kernel estimation can be formulated as per following algorithm

Here, the values of parameters η , γ , and τ are selected as 1, 0.002, and 15, respectively, according to [6]. Figure 2 shows the estimated kernel using the above algorithm as well

Algorithm 1 An algorithm for estimating blur kernel \mathbf{K}

Require: Blurry input image \mathbf{X}

$$\partial\mathbf{X} \leftarrow \text{Conv}_{1 \times 1}(\text{UFPdeblur}(\mathbf{X}))$$

$$\partial_v\mathbf{X}, \partial_h\mathbf{X} \leftarrow \text{Sobel}(\partial\mathbf{X}) \quad \triangleright \text{Sobel filter operation for horizontal and vertical gradient extraction}$$

$$\partial_v\mathbf{X}_e, \partial_h\mathbf{X}_e \leftarrow f_K(\partial_v\mathbf{X}), f_K(\partial_h\mathbf{X}) \quad \triangleright \text{Autoencoder operation}$$

for $i = 1 : \tau$ **do**

 Solve for \mathbf{K} using equation 4

 Solve for \mathbf{X}_l using 2nd optimization in equation 3

$$\partial\mathbf{X}_l \leftarrow \text{Conv}_{1 \times 1}(\text{UFPdeblur}(\mathbf{X}_l))$$

$$\partial\mathbf{X}_e \leftarrow \partial\mathbf{X}_l$$

end for
return Estimated blur kernel \mathbf{K}

as F2WD operation using this. Clearly, using the estimated kernel, F2WD is able to perform deblurring in latent space accurately. The accuracy of F2WD can be further improved if more accurate kernel estimation algorithm can be developed in feature space itself. We will investigate this further in future studies.

To estimate blur kernel for GoPro, RealBlur and HIDE, we use the pre-trained model of UFPNet. For defocus kernel estimation, we have retrained UFPNet for DDPD dataset from scratch according to the corresponding author's guideline.

B Proof of Proposition 1

The signal observation model in image domain can be written as,

$$\mathbf{Y} = \mathbf{K}\mathbf{X} + \mathbf{n}, \quad (5)$$

where, \mathbf{Y} is observed data, \mathbf{X} is true signal, \mathbf{n} is associated noise elements and \mathbf{K} is the degradation process. Here, we consider the noise \mathbf{n} to be independent of input processes and is zero mean.

We assume that the Wiener deconvolution operator is \mathbf{G} in FRFT domain. Then, we can define the estimate of the true signal as,

$$\hat{\mathbf{X}} = \mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y}, \quad (6)$$

where, \mathbf{C} corresponds to chirp multiplication with diagonal elements $C_{kk} = e^{-ik^2 \cot \theta/2}$. This is included due to the convolution theorem in FRFT domain as shown in equation 8 (main paper). The design of filter or Wiener deconvolution operator is based on minimizing the mean square error (MSE), which can be defined as following,

$$J_{\alpha} = \frac{1}{N} \mathbb{E} \{ (\mathbf{X} - \hat{\mathbf{X}})^H (\mathbf{X} - \hat{\mathbf{X}}) \} \quad (7)$$

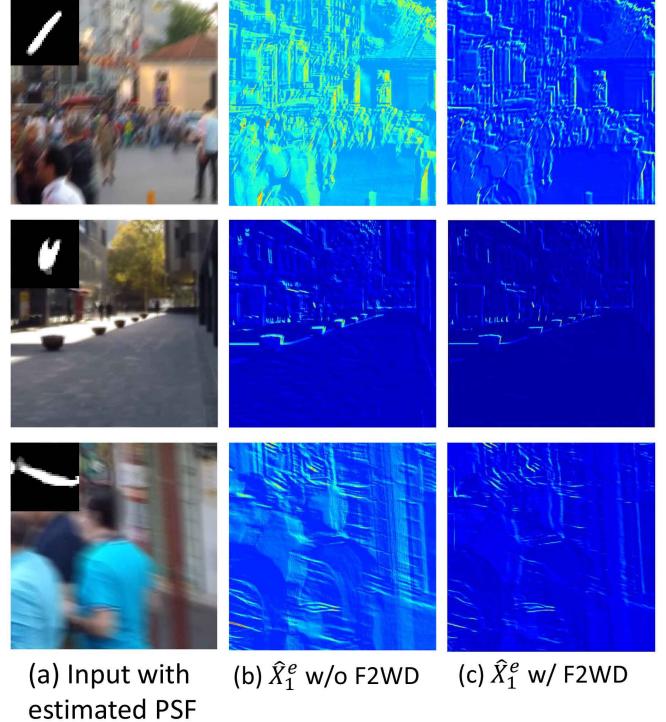


Figure 2: Estimated Kernel for a given blurry image and visualization of F2WD operation using it. It is also compared with features without any Wiener deconvolution operation for proper understanding.

Now, substituting equation 6 in above equation will result in

$$\begin{aligned} J_{\alpha} &= \frac{1}{N} \mathbb{E} \{ (\mathbf{X} - \mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y})^H (\mathbf{X} - \mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y}) \} \\ &= \frac{1}{N} [\mathbb{E} \{ \mathbf{X}^H \mathbf{X} \} - \mathbb{E} \{ \mathbf{X}^H \mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y} \} \\ &\quad - \mathbb{E} \{ (\mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y})^H \mathbf{X} \} \\ &\quad + \mathbb{E} \{ (\mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y})^H \mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y} \}] \end{aligned} \quad (8)$$

Now, we analyse each term in equation 8. For the first term $\mathbb{E} \{ \mathbf{X}^H \mathbf{X} \}$, it suggest the autocorrelation matrix of true signal denoted as $\mathbf{S}^{xx} = \mathbb{E} \{ \mathbf{X}^H \mathbf{X} \}$.

The second term can be re-written in terms of cross-correlation matrix between \mathbf{X} and \mathbf{Y} , which can be defined as, $\mathbf{S}^{xy} = \mathbb{E} \{ \mathbf{X} \mathbf{Y}^H \}$. Hence, $\mathbb{E} \{ \mathbf{X}^H \mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y} \} = \mathbb{E} \{ \mathbf{S}^{xy} \mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \}$.

Similarly the third term can be written as,

$$\mathbb{E} \{ (\mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y})^H \mathbf{X} \} = \mathbb{E} \{ (\mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha})^H \mathbf{S}^{yx} \}. \quad (9)$$

For the last term, it can be simplified as following,

$$\begin{aligned} &\mathbb{E} \{ (\mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y})^H \mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y} \} \\ &= \mathbb{E} \{ \mathbf{Y}^H \mathbf{F}^{-\alpha} \mathbf{G}^H \mathbf{C}^H \mathbf{F}^{\alpha} \mathbf{F}^{-\alpha} \mathbf{C} \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y} \} \\ &= \mathbb{E} \{ \mathbf{Y}^H \mathbf{F}^{-\alpha} \mathbf{G}^H \mathbf{G} \mathbf{F}^{\alpha} \mathbf{Y} \} \end{aligned} \quad (10)$$

In equation 10, the last line is from the properties of FRFT that $\mathbf{F}^{\alpha} \mathbf{F}^{-\alpha} = \mathbf{F}^{-\alpha+\alpha} = \mathbf{F}^0$, which is a unitary matrix. Similarly, $\mathbf{C}^H \mathbf{C} = \mathbf{I}$, due to the orthogonal properties of chirp function.

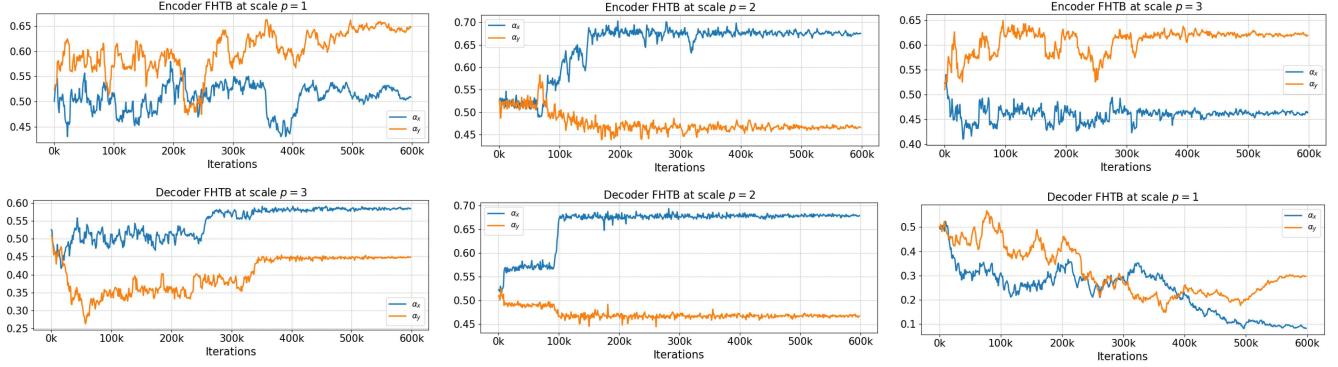


Figure 3: Visualization of variation of α for FRFT operations in FHTBs at different scales of F2former base model

Now, as the \mathbf{G} consists of only diagonal elements, the equation 8 can be rewritten in terms of trace of the matrix. So, using the above expression, it can be reformulated as,

$$J_\alpha = \frac{1}{N} [\text{tr}(\mathbf{S}^{xx}) - \text{tr}(\mathbf{S}^{xy}\mathbf{F}^{-\alpha}\mathbf{CGF}^\alpha) - \text{tr}((\mathbf{F}^{-\alpha}\mathbf{CGF}^\alpha)^H\mathbf{S}^{yx}) + \text{tr}(\mathbf{Y}^H\mathbf{F}^{-\alpha}\mathbf{G}^H\mathbf{GF}^\alpha\mathbf{Y})], \quad (11)$$

where, $\text{tr}(\cdot)$ is trace of matrix. Now, for each diagonal element of \mathbf{G}_j , we minimize J_α as following,

$$\frac{\partial J_\alpha}{\partial \mathbf{G}_j} = -(\mathbf{C}_{jj}\mathbf{F}_{jj}^\alpha\mathbf{S}_j^{xy}\mathbf{F}_{jj}^{-\alpha})^H + \mathbf{F}_{jj}^\alpha\mathbf{S}_j^{yy}\mathbf{F}_{jj}^{-\alpha}\mathbf{G}_j. \quad (12)$$

To minimize J_α , we have to make above equation 0, and then solving for \mathbf{G}_j we get,

$$\mathbf{G}_j = \frac{(\mathbf{C}_{jj}\mathbf{F}_{jj}^\alpha\mathbf{S}_j^{xy}\mathbf{F}_{jj}^{-\alpha})^H}{\mathbf{F}_{jj}^\alpha\mathbf{S}_j^{yy}\mathbf{F}_{jj}^{-\alpha}}, \quad (13)$$

where, \mathbf{S}^{yy} is auto-correlation of signal \mathbf{Y} , which can be defined as,

$$\begin{aligned} \mathbf{S}^{yy} &= \mathbb{E}\{\mathbf{YY}^H\} = \mathbb{E}\{(\mathbf{KX} + \mathbf{n})(\mathbf{KX} + \mathbf{n})^H\} \\ &= \mathbb{E}\{\mathbf{KXX}^H\mathbf{K}^H\} + \mathbb{E}\{\mathbf{nn}^H\} + \mathbb{E}\{\mathbf{nX}^H\mathbf{K}^H\} + \mathbb{E}\{\mathbf{XKn}^H\}. \end{aligned} \quad (14)$$

In above equation, $\mathbb{E}\{\mathbf{nX}^H\mathbf{K}^H\} = \mathbb{E}\{\mathbf{XKn}^H\} = 0$, as \mathbf{X} and \mathbf{n} are independent. Now, as $\mathbf{S}^{nn} = \mathbb{E}\{\mathbf{nn}^H\}$ and $\mathbf{S}^{xx} = \mathbb{E}\{\mathbf{XX}^H\}$, we replace them in above equation to get

$$\mathbf{S}^{yy} = \mathbf{KS}^{xx}\mathbf{K}^H + \mathbf{S}^{nn}. \quad (15)$$

Similarly, the cross correlation term \mathbf{S}^{xy} can be written as,

$$\begin{aligned} \mathbf{S}^{xy} &= \mathbb{E}\{\mathbf{XY}^H\} = \mathbb{E}\{\mathbf{X}(\mathbf{KX} + \mathbf{n})^H\} \\ &= \mathbb{E}\{\mathbf{X}\mathbf{X}^H\mathbf{K}^H\} + \mathbb{E}\{\mathbf{Xn}^H\} = \mathbf{S}^{xx}\mathbf{K}^H. \end{aligned} \quad (16)$$

Now transferring equation 14 and 16 in equation 13, we will get the derived formulation in equation 7 (main paper). Hence, this completes the proof.

C Ablation study (continued)

C.1 Effect of F3RB

We train our base model without F3RB which was added to extract features in terms of different contextual information with respect to varied α parameter for FRFT operation. We observe a 0.13 dB drop in PSNR compared to our baseline performance.

C.2 Visualization of learnable α for FRFT in F2WD and FHTB

In all FRFT operations in F2former, we have taken the α parameter as learnable parameter, specifically, we set $\alpha = \{\alpha_x, \alpha_y\}$ to apply FRFT in horizontal and vertical directions of an image separately. The main reason behind this is that based on different blur content at different scales there would be separate requirement of α parameters to represent optimal spatial-frequency information. It also depends on specific task of the block like the role of FHTB blocks in encoder side is different compared to that of decoder side (encoder part minimizes the blur content at different scales, whereas the decoder part efficiently reconstructs the sharp image details). This can be visualized from Figure 3 where each plot represent the average α variation with respect to training iterations and corresponding to a particular FHTB block at particular scale. Usually, for encoder operations α varies between (0.4, 0.7), whereas for decoder operation it varies between (0.1, 0.7). Our hypothesis is that as decoder (specially at last stage) mainly focuses on reconstructing the sharp image details, it majorly focuses on the spatial content of the image making it low α values as shown in last plot of Figure 3. It is also noticeable from the above figure that α also varies across horizontal and vertical directions. This is mostly due to focus on different important gradient information at separate scales and separate levels (encoder or decoder).

Figure 4 shows the α variation to perform featured based fractional Wiener deconvolution at different scales. As at down-sampled level, the spatially varying blur dominates more, we see more focus of F2WD to spatial content (low α values) as the model gets deeper. Due to this optimal choice of α ,

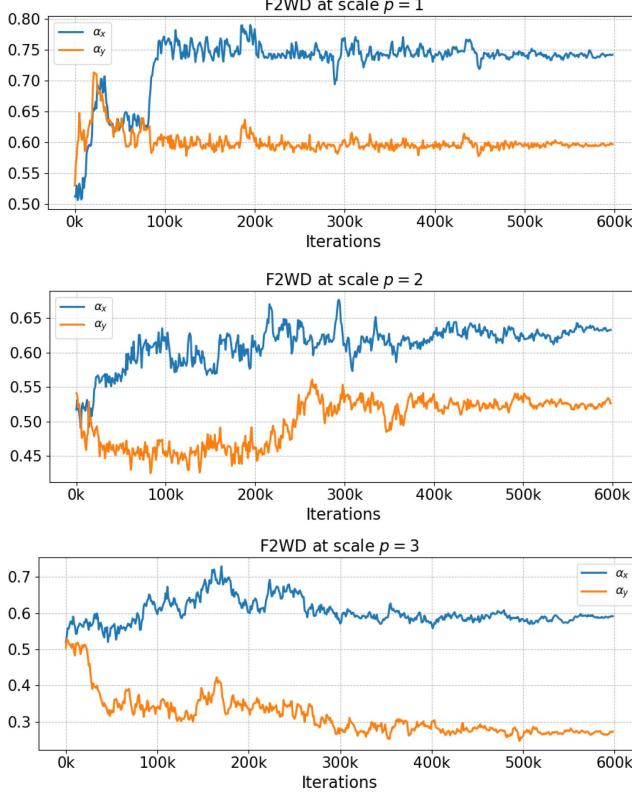


Figure 4: Visualization of variation of α for FRFT operations in F2WDs at different scales of F2former base model

F2WD is able to perform feature based deblurring operation more effectively as shown in Figure 2. For further verification, we retrain our model with fixed $\alpha = 0.5$ for both horizontal and vertical direction as suggested by [7] for both F2WD and FSGT. We observe a 0.18 dB reduced PSNR compared to our baseline performance.

C.3 Effect of Cosine-bell function in FM-FFN

The cosine bell formulation in equation 13 (main paper) is generic. The exact formulation is defined as below,

$$\mathbf{W}_{jk}^l = \begin{cases} 1 & \text{if } |u| \leq u_c - \frac{u_s}{2} \\ 0.5 \left[1 + \cos \left(\frac{\pi(|u| - u_c + \frac{u_s}{2})}{u_s} \right) \right] & \text{if } \lambda_{ls} < |u| < \lambda_{rs} \\ 0 & \text{if } |u| \geq u_c + \frac{u_s}{2}, \end{cases} \quad (17)$$

where, $\lambda_{ls} = u_c - \frac{u_s}{2}$, $\lambda_{rs} = u_c + \frac{u_s}{2}$, and $u = \sqrt{j^2 + k^2}$. So, threshold λ in equation 13 is basically consists of two parameters, u_c which is cut-off frequency and u_s which is pass band frequency to reduce ringing effect in spatial domain. Figure 5 shows the advantage of using cosine bell function compared to other formulations like butterworth and hanning window operation. Clearly cosine bell reduces the ringing artifacts and reconstructs high and low frequency images more effectively for same cut-off frequency.

We keep both u_c and u_s as the learning parameter dur-

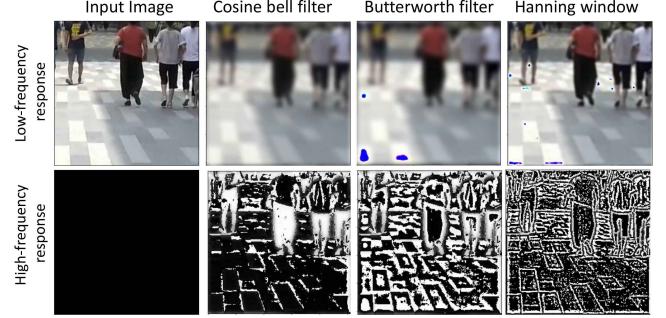


Figure 5: Visualization of extracted low-pass and high-pass image signal using different filters.

Table 1: Effect of different filters on the performance of F2former

Filters	PSNR	SSIM
Cosine Bell	33.24	0.961
Butterworth	33.02	0.953
Hanning window	33.11	0.957

ing training our network. Figure 6 shows the variation of u_c and u_s across the training iterations. For optimal high and low frequency extraction, the cut-off frequency for all FHTB blocks varies between (8,12) in saturation whereas to reduce the ringing effect the pass band frequency mainly varies in (24,34) across all the FHTBs. For further validation we use the three filters shown in Figure 5 for training upto 200k iterations on GoPro dataset. The respective test results are shown in Table 1. Clearly, the utilization of cosine bell function as a filter benefits our proposed F2former compared to other filters.

C.4 Visualization of effect of FM-FFN

Figure 7 shows how FM-FFN extracts high and low frequency features, and based on important frequency information it generates sharp details as shown in the Figure. Clearly, compared to before FFN features, the features after FFN has more sharper edges and more structural details.

References

- [1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022.
- [2] Yuning Cui, Yi Tao, Zhenshan Bing, Wenqi Ren, Xinwei Gao, Xiaochun Cao, Kai Huang, and Alois Knoll. Selective frequency network for image restoration. In *The Eleventh International Conference on Learning Representations*, 2023.
- [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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18105–18114, 2023.

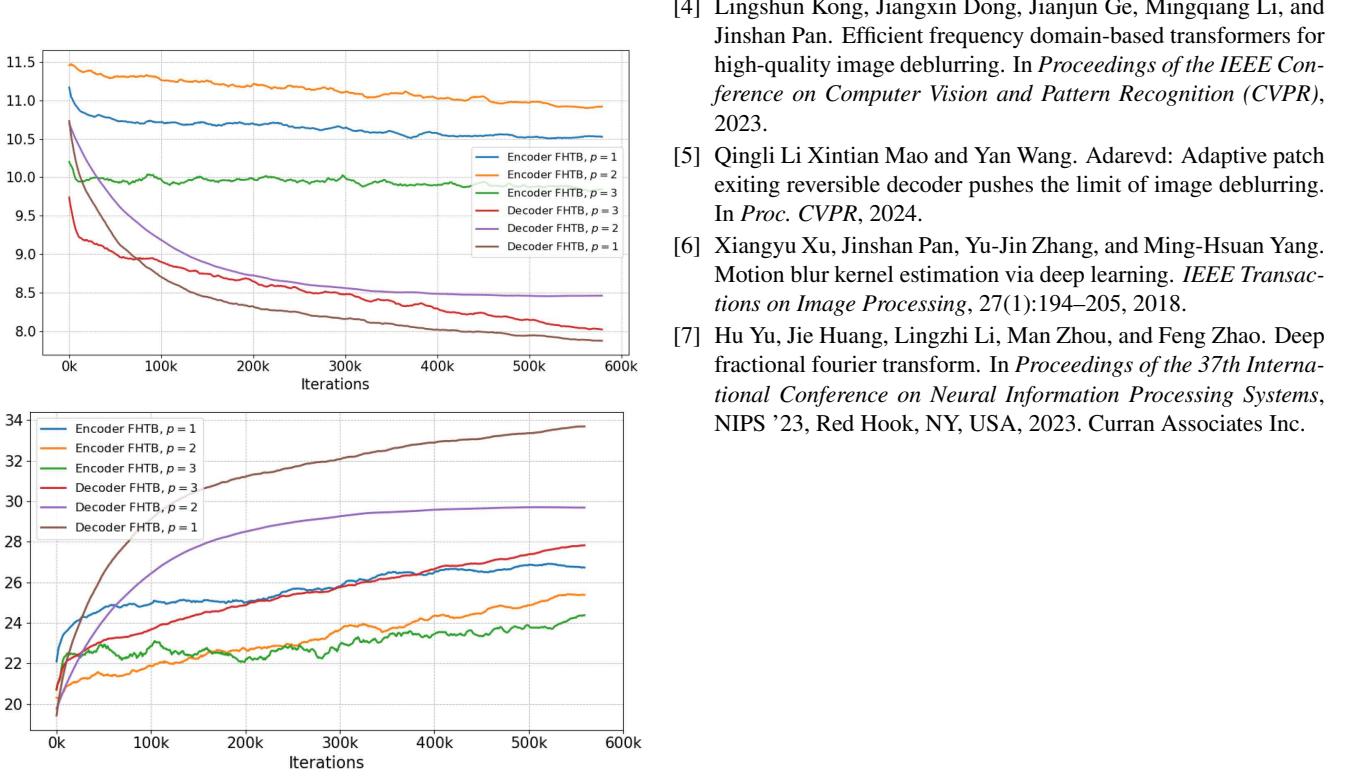


Figure 6: Variation of u_c (top) and u_s (bottom) across the training iterations.

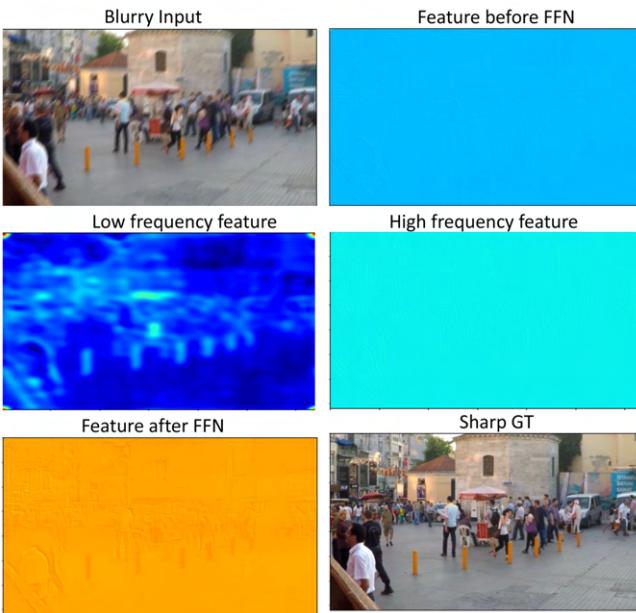


Figure 7: Visualization of extracted low and high frequency features in FM-FFN of last F2TB block (better visualized in 200-300%)

- [4] Lingshun Kong, Jiangxin Dong, Jianjun Ge, Mingqiang Li, and Jinshan Pan. Efficient frequency domain-based transformers for high-quality image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [5] Qingli Li Xintian Mao and Yan Wang. Adarevd: Adaptive patch exiting reversible decoder pushes the limit of image deblurring. In *Proc. CVPR*, 2024.
- [6] Xiangyu Xu, Jinshan Pan, Yu-Jin Zhang, and Ming-Hsuan Yang. Motion blur kernel estimation via deep learning. *IEEE Transactions on Image Processing*, 27(1):194–205, 2018.
- [7] Hu Yu, Jie Huang, Lingzhi Li, Man Zhou, and Feng Zhao. Deep fractional fourier transform. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA, 2023. Curran Associates Inc.

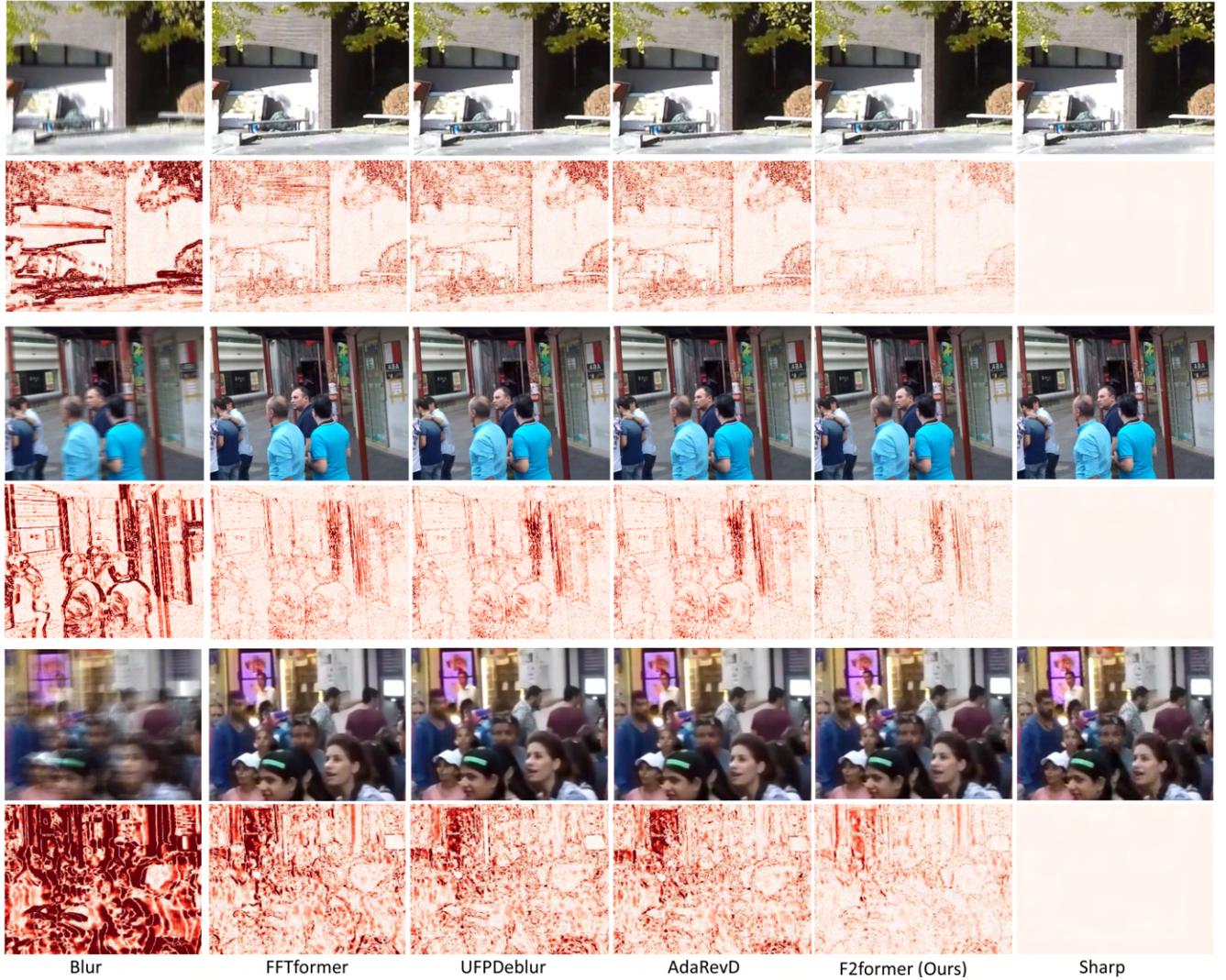


Figure 8: Examples on the GoPro test dataset. The odd rows show blur image, predicted images of different methods, and GT sharp image. The even rows show the corresponding residual of the blur image / predicted sharp images and GT sharp image. Better visualized at 200%.

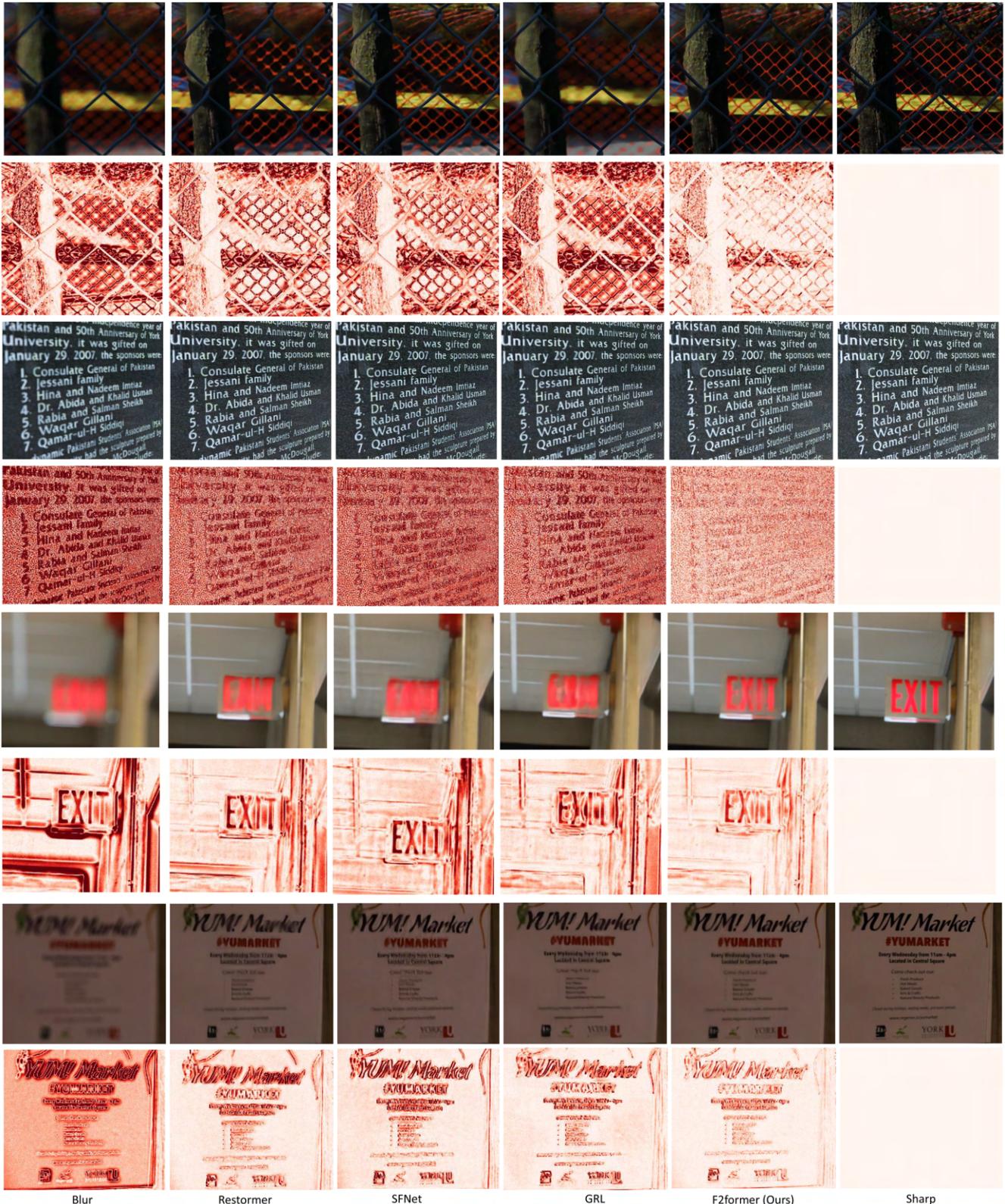


Figure 9: Examples on the DDPD-single pixel test dataset. The odd rows show blur image, predicted images of different methods, and GT sharp image. The even rows show the corresponding residual of the blur image / predicted sharp images and GT sharp image. Better visualized at 200%.



Figure 10: Examples on the DDPD-dual pixel test dataset. The odd rows show blur image, predicted images of different methods, and GT sharp image. The even rows show the corresponding residual of the blur image / predicted sharp images and GT sharp image. Better visualized at 200%.

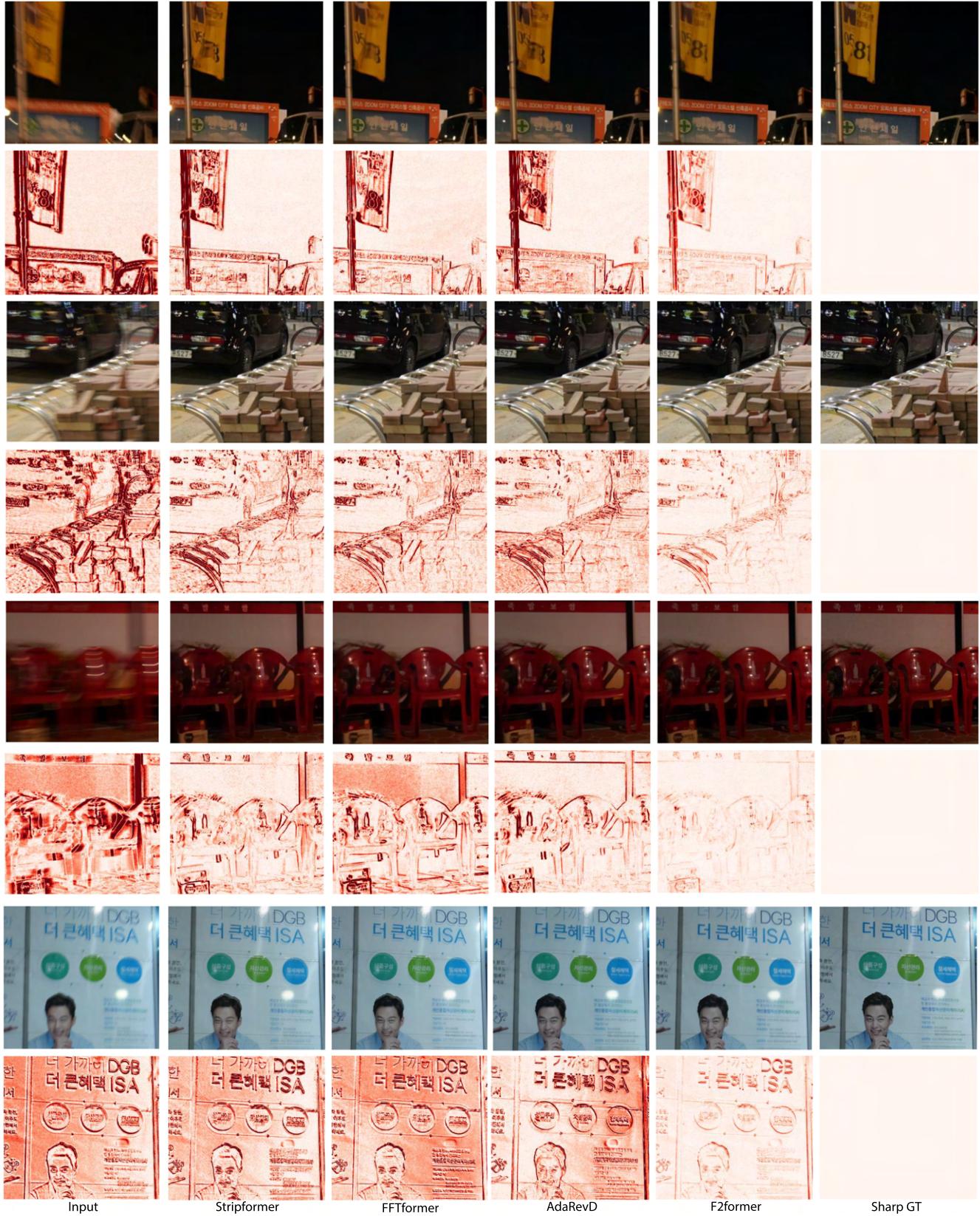


Figure 11: Examples on the RealBlur-J test dataset. The odd rows show blur image, predicted images of different methods, and GT sharp image. The even rows show the corresponding residual of the blur image / predicted sharp images and GT sharp image. Better visualized at 200%.

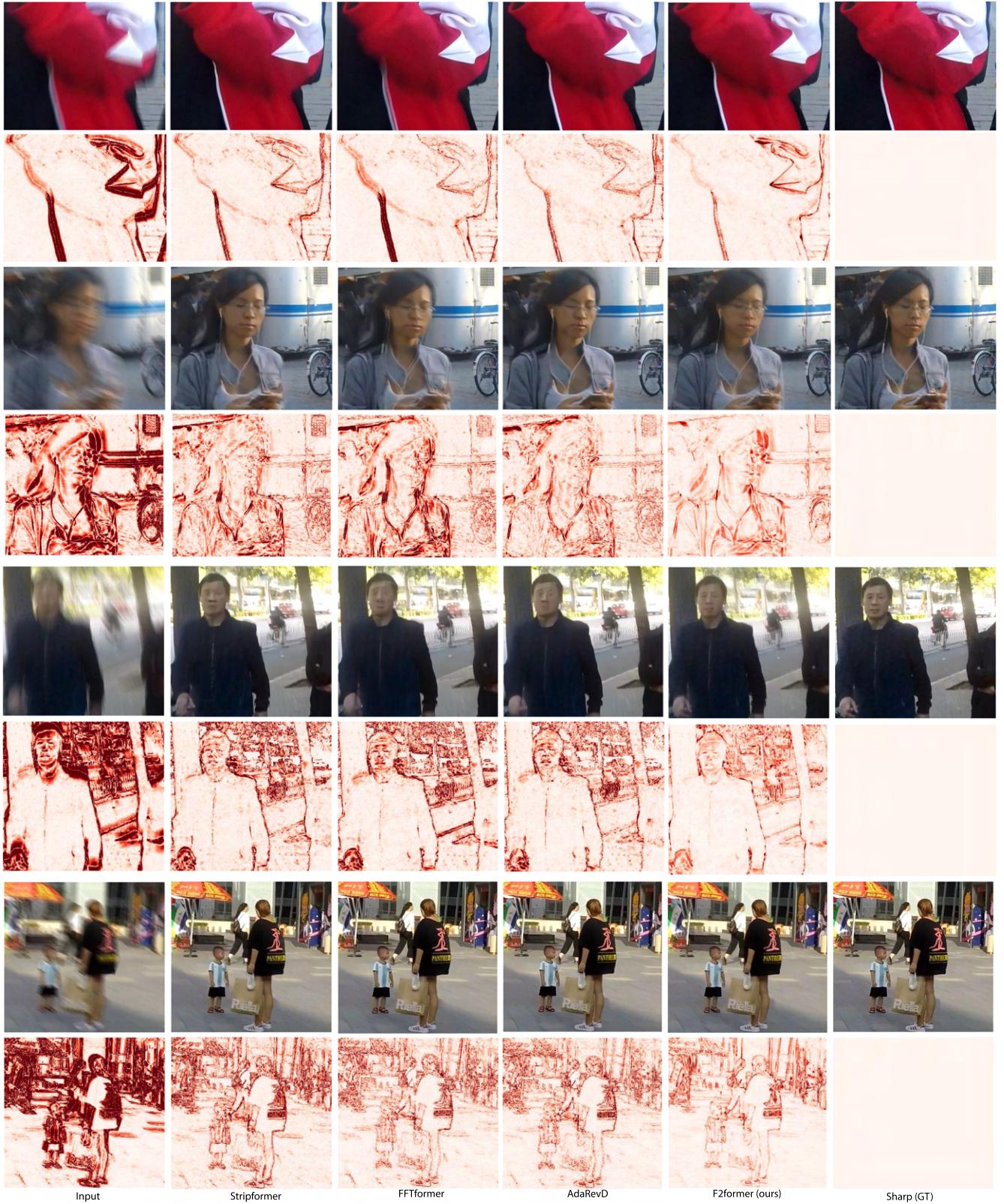


Figure 12: Examples on the HIDE test dataset. The odd rows show blur image, predicted images of different methods, and GT sharp image. The even rows show the corresponding residual of the blur image / predicted sharp images and GT sharp image. Better visualized at 200%.