

# DeFlow: Learning Complex Image Degradations from Unpaired Data with Conditional Flows

## Appendix

Valentin Wolf    Andreas Lugmayr    Martin Danelljan    Luc Van Gool    Radu Timofte  
vawolf@ethz.ch    {andreas.lugmayr, martin.danelljan, vangool, radu.timofte}@vision.ee.ethz.ch  
Computer Vision Lab, ETH Zurich, Switzerland

In Sec. A of this appendix, we first derive the closed-form solution of the 1D Gaussian example from Sec. 3.1. We then go on in Sec. B and show that restricting  $p_x$  to a standard normal distribution is absorbed by a single affine layer in the deep flow model. Next, we provide a derivation to the DeFlow method with domain invariant conditioning in Sec. C. We then show in Sec. D that degradations generated by DeFlow are stochastic and can be sampled at varying strengths. Further, we provide a visual comparison of the degradations and more example images of the downstream real-world super-resolution (RWSR) performance in Sec. E. Lastly, we give insight into the set-up of the conducted user study in Sec. F.

### A. Closed-Form Solution for the 1D Gaussian Example

Here we present a detailed derivation for the closed-form solution to the 1-dimensional Gaussian example from Sec. 3.1. To recall, we are given two datasets  $\mathcal{X} = \{x_i\}_{i=1}^N$  and  $\mathcal{Y} = \{y_i\}_{i=1}^M$ . We know that  $x \in \mathcal{X}$  are i.i.d. samples from  $p_x = \mathcal{N}(\mu_x, \sigma_x^2)$ . Further, we know that  $y = x + u \in \mathcal{Y}$  are i.i.d. samples from  $x \sim p_x$  with additive independent Gaussian noise  $u \sim p_u = \mathcal{N}(\mu_u, \sigma_u^2)$ .

The task is to find the parameters  $\theta^* = \{\mu_x, \sigma_x^2, \mu_u, \sigma_u^2\}$  that jointly maximize the marginal likelihoods  $p_x(\mathcal{X})$  and  $p_y(\mathcal{Y})$ .

Proceeding as usual, we apply the i.i.d. property and minimize the negative log-likelihood w.r.t.  $\theta$ ,

$$\begin{aligned} \min l(\theta) = & -\frac{1}{N} \sum_{i=1}^N \frac{(x_i - \mu_x)^2}{\sigma_x^2} + \frac{\ln \sigma_x^2}{2} \\ & -\frac{1}{M} \sum_{i=1}^M \frac{(y_i - \mu_x - \mu_u)^2}{\sigma_x^2 + \sigma_u^2} + \frac{\ln(\sigma_x^2 + \sigma_u^2)}{2} \end{aligned}$$

subject to  $\sigma_x \geq 0, \sigma_u \geq 0$ . (1)

To ensure the estimated variances are non-negative, *i.e.*

$\sigma_x \geq 0$  and  $\sigma_u \geq 0$ , we introduce the Lagrange multipliers  $\lambda_x$  and  $\lambda_u$  and have,

$$\hat{l}(\theta) = l(\theta) - \lambda_x \sigma_x^2 - \lambda_u \sigma_u^2. \quad (2)$$

By the Karush–Kuhn–Tucker theorem,  $\theta^*$  is a optimal solution to  $l(\theta)$  if  $\frac{\partial \hat{l}(\theta^*)}{\partial \theta} = 0$  while  $\lambda_x \geq 0, \lambda_u \geq 0, \lambda_x \sigma_x^2 = 0$  and  $\lambda_u \sigma_u^2 = 0$  hold.

Next, we take partial derivatives of  $\hat{l}(\theta)$  w.r.t. the individual parameters and set them to 0 to obtain the optimal estimates. First, we differentiate w.r.t. the means  $\mu_x$  and  $\mu_u$ , and obtain

$$\frac{\partial \hat{l}(\theta)}{\partial \mu_u} = \frac{1}{M} \frac{1}{\sigma_x^2 + \sigma_u^2} \sum_{y \in \mathcal{Y}} (y - (\mu_x + \mu_u)) \stackrel{!}{=} 0 \quad (3)$$

$$\begin{aligned} \frac{\partial \hat{l}(\theta)}{\partial \mu_x} = & \frac{1}{N} \frac{1}{\sigma_x^2} \sum_{x \in \mathcal{X}} (x - \mu_x) \\ & + \frac{1}{M} \frac{1}{\sigma_x^2 + \sigma_u^2} \sum_{y \in \mathcal{Y}} (y - (\mu_x + \mu_u)) \\ \stackrel{(3)}{=} & \frac{1}{N} \frac{1}{\sigma_x^2} \sum_{x \in \mathcal{X}} (x - \mu_x) \stackrel{!}{=} 0. \end{aligned} \quad (4)$$

It directly follows, that the optimal estimates of  $\mu_x$  and  $\mu_u$  can be written as the empirical means  $\hat{\mu}_x$  and  $\hat{\mu}_y$ ,

$$\mu_x = \hat{\mu}_x = \frac{1}{N} \sum_{x \in \mathcal{X}} x \quad (5)$$

$$\mu_u = \hat{\mu}_y - \hat{\mu}_x, \quad \hat{\mu}_y = \frac{1}{M} \sum_{y \in \mathcal{Y}} y. \quad (6)$$

Now we turn to the estimation of the variances. We first

obtain the following partial derivatives,

$$\frac{\partial \hat{l}(\theta)}{\partial \sigma_u^2} = \frac{1}{2(\sigma_x^2 + \sigma_u^2)} - \frac{1}{2M} \sum_{y \in \mathcal{Y}} \frac{(y - \hat{\mu}_y)^2}{(\sigma_x^2 + \sigma_u^2)^2} - \lambda_u \quad (7)$$

$$\begin{aligned} \frac{\partial \hat{l}(\theta)}{\partial \sigma_x^2} &= \frac{1}{2\sigma_x^2} - \frac{1}{2N} \sum_{x \in \mathcal{X}} \frac{(x - \mu_x)^2}{\sigma_x^4} \\ &+ \frac{1}{2(\sigma_x^2 + \sigma_u^2)} - \frac{1}{2M} \sum_{y \in \mathcal{Y}} \frac{(y - \hat{\mu}_y)^2}{(\sigma_x^2 + \sigma_u^2)^2} - \lambda_x. \end{aligned} \quad (8)$$

Setting  $\frac{\partial \hat{l}(\theta)}{\partial \sigma_u^2}$  to 0 and using the complementary slackness condition that  $\lambda_u \sigma_u = 0$  must hold at the minimum we obtain,

$$\frac{\partial \hat{l}(\theta)}{\partial \sigma_u^2} \doteq 0 \quad (9)$$

$$\iff -(\sigma_x^2 + \sigma_u^2) + 2\lambda_u(\sigma_x^2 + \sigma_u^2)^2 + \hat{\sigma}_y^2 = 0 \quad (10)$$

$$\iff 2\lambda_u \sigma_x^4 - \sigma_x^2 - \sigma_u^2 + \hat{\sigma}_y^2 = 0 \quad (11)$$

$$\iff \sigma_u^2 = 2\lambda_u \sigma_x^4 - \sigma_x^2 + \hat{\sigma}_y^2. \quad (12)$$

where  $\hat{\sigma}_y = \frac{1}{M} \sum_{y \in \mathcal{Y}} (y - \hat{\mu}_y)^2$  is used as short-hand notation for the empirical variance of  $\mathcal{Y}$ .

Similarly, we set  $\frac{\partial \hat{l}(\theta)}{\partial \sigma_x^2}$  to 0. We first define the empirical variance of  $\mathcal{X}$  as  $\hat{\sigma}_x^2 = \frac{1}{N} \sum_{x \in \mathcal{X}} (x - \mu_x)^2$ . By using the complementary slackness condition and the fact that  $\frac{\partial \hat{l}(\theta)}{\partial \sigma_u^2} = 0$ , we achieve

$$\frac{\partial \hat{l}(\theta)}{\partial \sigma_x^2} \doteq 0 \quad (13)$$

$$\iff \frac{1}{2\sigma_x^2} - \frac{1}{2} \frac{\hat{\sigma}_x^2}{\sigma_x^4} - \lambda_x + \lambda_u = 0 \quad (14)$$

$$\iff \sigma_x^2 - \hat{\sigma}_x^2 + 2\lambda_u \sigma_x^4 = 0 \quad (15)$$

$$\iff \sigma_x^2 = \hat{\sigma}_x^2 - 2\lambda_u \sigma_x^4. \quad (16)$$

Finally, the complementary slackness condition leaves us with two cases to consider: (1)  $\lambda_u = 0$  and (2)  $\sigma_u^2 = 0$ . In the former case, it directly follows from (12) and then (16) that

$$\text{Case 1:} \quad \text{valid iff. } \hat{\sigma}_y^2 \geq \hat{\sigma}_x^2 \quad (17)$$

$$\sigma_x^2 = \hat{\sigma}_x^2 \quad (18)$$

$$\sigma_u^2 = \hat{\sigma}_y^2 - \hat{\sigma}_x^2. \quad (19)$$

In the case of  $\sigma_u^2 = 0$ , we first obtain from (12) that

$$2\lambda_u \sigma_x^4 = \sigma_x^2 - \hat{\sigma}_y^2. \quad (20)$$

Inserting this into (16) gives the desired solution for  $\sigma_x^2$  as

$$\text{Case 2:} \quad \text{valid iff. } \hat{\sigma}_y^2 \leq \hat{\sigma}_x^2 \quad (21)$$

$$\sigma_x^2 = \frac{\hat{\sigma}_x^2 + \hat{\sigma}_y^2}{2} \quad (22)$$

$$\sigma_u^2 = 0. \quad (23)$$

The second case thus corresponds to the solution where  $u$  is an unknown constant variable.

## B. Closed-Form Solution for the 1-Dimensional Gaussian Case using DeFlow with a Single Affine Layer

In our proposed DeFlow method, we restrict the base distribution  $p_x$  to be  $\mathcal{N}(0, 1)$ , while keeping  $p_u = \mathcal{N}(\mu_u, \sigma_u^2)$ . We show that a single-affine-layer flow  $f_\theta(x) = ax + b$  is able to obtain the an optimal solution for the 1-dimensional Gaussian setting from the previous section under this restriction. To do so, we simply set

$$a = \frac{1}{\sigma_x}, \quad b = -\frac{\mu_x}{\sigma_x} \quad (24)$$

where  $\mu_x$  and  $\sigma_x$  are the optimal estimates obtained in the previous section. Intuitively, we can interpret the single-layer flow as a learned normalization layer, that ensures a standard normal distribution in the latent space. To recover the optimal parameters  $\tilde{\mu}_u^2$  and  $\tilde{\sigma}_u^2$  of  $p_u$ , we need to adjust the optimal values retrieved in the previous section accordingly to this normalization and obtain

$$\tilde{\mu}_u = \frac{\mu_u}{\sigma_x}, \quad \tilde{\sigma}_u^2 = \frac{\sigma_u^2}{\sigma_x^2}. \quad (25)$$

This shows that the restriction of  $p_x$  to be standard normal simply leads to an absorption of the required normalization in an affine layer of the flow model.

## C. Derivation of the Domain Invariant Conditional DeFlow Method

To generalize the formulation of DeFlow from Sec. 3.2 to include the domain invariant conditioning  $h(x)$ , we extend the flow network to  $z_{x|h(x)} = f_\theta(x; h(x))$  and  $z_{y|h(y)} = f_\theta(y; h(y))$ . By invertibility in the first arguments of  $f_\theta$ , samples can then be retrieved by

$$x = f_\theta^{-1}(z_x; h(x)), \quad y = f_\theta^{-1}(z_x + u; h(y)) \quad (26a)$$

$$z_{x|h(x)} \sim \mathcal{N}(0, I), \quad u \sim p_u = \mathcal{N}(\mu_u, \Sigma_u), \quad z_{x|h(x)} \perp u. \quad (26b)$$

Then, by domain invariance  $h(x) = h(y)$ , it follows that we can sample from the conditional distribution  $p(y|x, h(x), h(y)) = p(y|x)$  using

$$y = f_\theta^{-1}(f_\theta(x; h(x)) + u; h(x)) \sim p(y|x) \quad (27)$$

where  $u \sim \mathcal{N}(\mu_u, \Sigma_u)$ .

By the change of variables formula, we obtain the differentiable expressions for the conditional marginal distribu-

tions,

$$p(x|h(x)) = |\det Df_\theta(x; h(x))| \cdot \mathcal{N}(f_\theta(x; h(x)); 0, I) \quad (28a)$$

$$p(y|h(y)) = |\det Df_\theta(y; h(y))| \cdot \mathcal{N}(f_\theta(y; h(y)); \mu_u, I + \Sigma_u). \quad (28b)$$

As in the unconditional case, the first factor is given by the determinant of the Jacobian  $Df_\theta$  of the flow network, while the second factor stems from the Gaussian base distributions from out latent space formulation.

We can then use (28) to allow the optimization of the new negative log-conditional-likelihood objective

$$L(\theta) = -\frac{1}{n} \sum_{i=1}^n \ln p_x(x_i|h(x_i)) - \frac{1}{m} \sum_{j=1}^m \ln p_y(y_j|h(y_j)). \quad (29)$$

## D. DeFlow Degradation Results

**Stochasticity of Degradations** Current GAN based approaches [1, 2, 5, 3] model the degradation process as a deterministic mapping, ignoring its inherent stochastic nature. In contrast, DeFlow learns the conditional distribution  $p(y|x)$  of a degraded image  $y$  given a clean image  $x$  and thereby allows sampling multiple degraded versions of a single clean image. As shown in Fig. 1, different degraded samples from DeFlow feature different yet realistic noise characteristics without noticeable bias or recurring patterns.

**Varying Degradation Strength** We further show that DeFlow can be extended to enable sampling degradations at different strengths. To do so, we include a temperature parameter  $\tau$  that scales the sampled shift-vector  $u$  in the latent space. This extends (8) to

$$y = f_\theta^{-1}(f_\theta(x; h(x)) + \tau u; h(x)). \quad (30)$$

As shown in Figure 2, setting  $\tau < 1$  yields more nuanced degradations, while  $\tau > 1$  amplifies the noise.

## E. Visual Comparison

While we compared DeFlow to current methods using reference and no-reference based evaluation metrics and a user study, we here provide detailed visual results.

**Degradation Results:** We thus show examples of the synthetic degradations generated from different methods in Figures 4, 6, and 8 for the AIM-, NTIRE-, and DPED-RWSR datasets. As a reference, we further provide examples of real noisy image patches from the respective datasets in Figures 3, 5, and 7. We notice that DeFlow consistently adds more noise compared to the other methods. Yet, on all datasets, the degradations from DeFlow resemble the real noisy data, whereas other learned methods struggle to pickup on the noise characteristics.

**Real-World Super-Resolution Performance:** Further, we provide results of the downstream real-world super-resolution task of the different methods on the AIM-, NTIRE-, and DPED-RWSR datasets in Figures 9, 10, and 11, respectively. It is noticeable, that our proposed approach introduces fewer artifacts than the other methods across all datasets. Further, DeFlow is able to reconstruct fine details and provides sharper images than the White Noise model, which performs surprisingly well on the synthetic datasets. On DPED, the performance of the DeFlow degradations is comparable to the handcrafted approach of Impressionism [2]. While DeFlow retains more noise in smooth patches, Impressionism tends to over-smooth textures.

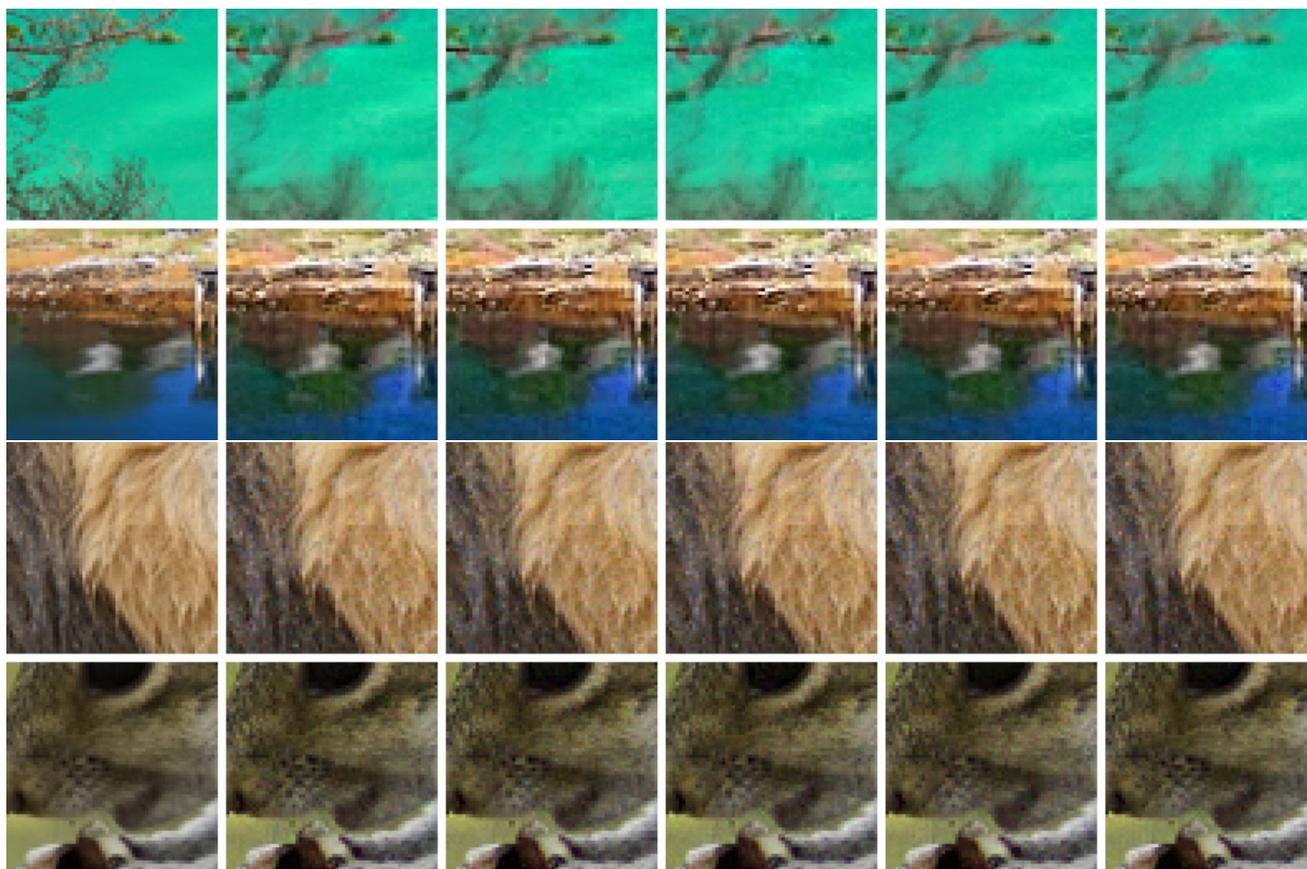
## F. Details of the User Study

In this section, we give insight into how we conducted the user study. On AIM and DPED we chose the top 7 models by their LPIPS score to compare in the user study. On DPED we decided to only compare against Frequency Separation [1] and Impressionism [2] both with their super-resolution pipeline and ours, as we found that other methods performed considerably worse.

For all datasets we used the following set-up for the user study: Participants were shown the same random crop from two different super-resolution models. In addition, we showed them the whole image where the cropped patch was marked in red. Participants were then asked to pick the super-resolved patch that looks more realistic. For that we used three random crops of size  $80 \times 80$  pixels per image of each validation dataset and asked five different study participants per pair.

## References

- [1] M. Fritsche, S. Gu, and R. Timofte. Frequency separation for real-world super-resolution. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3599–3608, 2019. 3, 5, 6, 7, 8, 9, 10
- [2] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 3, 5, 6, 7, 8, 9, 10
- [3] Andreas Lugmayr, Martin Danelljan, Radu Timofte, et al. Aim 2019 challenge on real-world image super-resolution: Methods and results. In *ICCV Workshops*, 2019. 3, 6, 7, 10
- [4] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks. *ECCV*, 2018. 10
- [5] Yunxuan Wei, Shuhang Gu, Yawei Li, and Longcun Jin. Unsupervised real-world image super resolution via domain-distance aware training, 2020. 3, 5, 8, 9



Clean Input

Different Samples with  $\tau = 1.0$

Figure 1. Multiple degraded samples of a clean input image (left column) using DeFlow on the AIM-RWSR (top two rows) and NTIRE-RWSR (bottom two rows).



Clean Input

$\tau = 0.33$

$\tau = 0.66$

$\tau = 1.0$

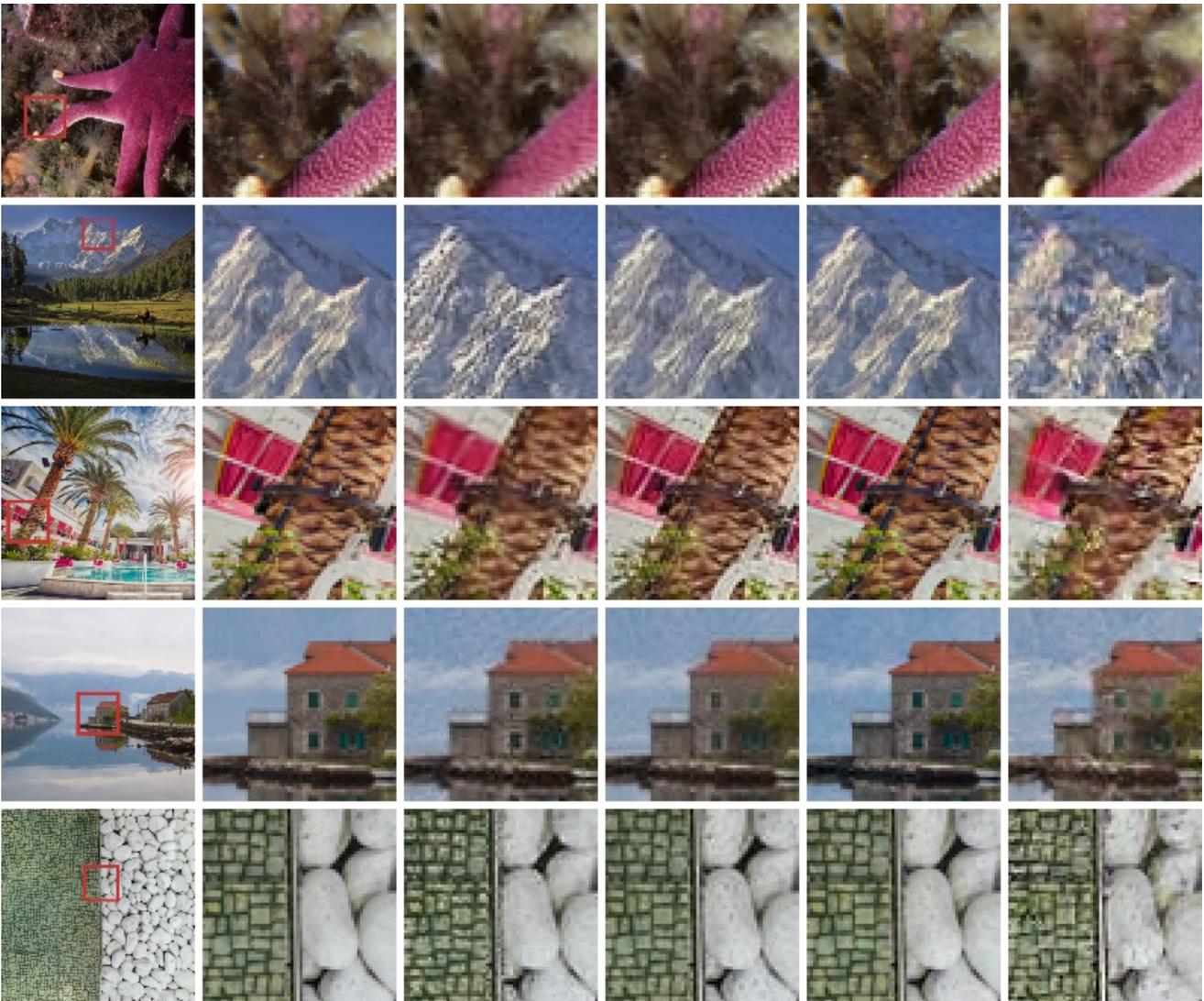
$\tau = 1.33$

$\tau = 1.66$

Figure 2. Sampling degradations from DeFlow with increasing temperature  $\tau$  in (30) on the AIM-RWSR (top row) and NTIRE-RWSR (bottom row).



Figure 3. AIM-RWSR: examples of noisy image patches.



Clean Input

DASR [5]

Frequency Separation [1]

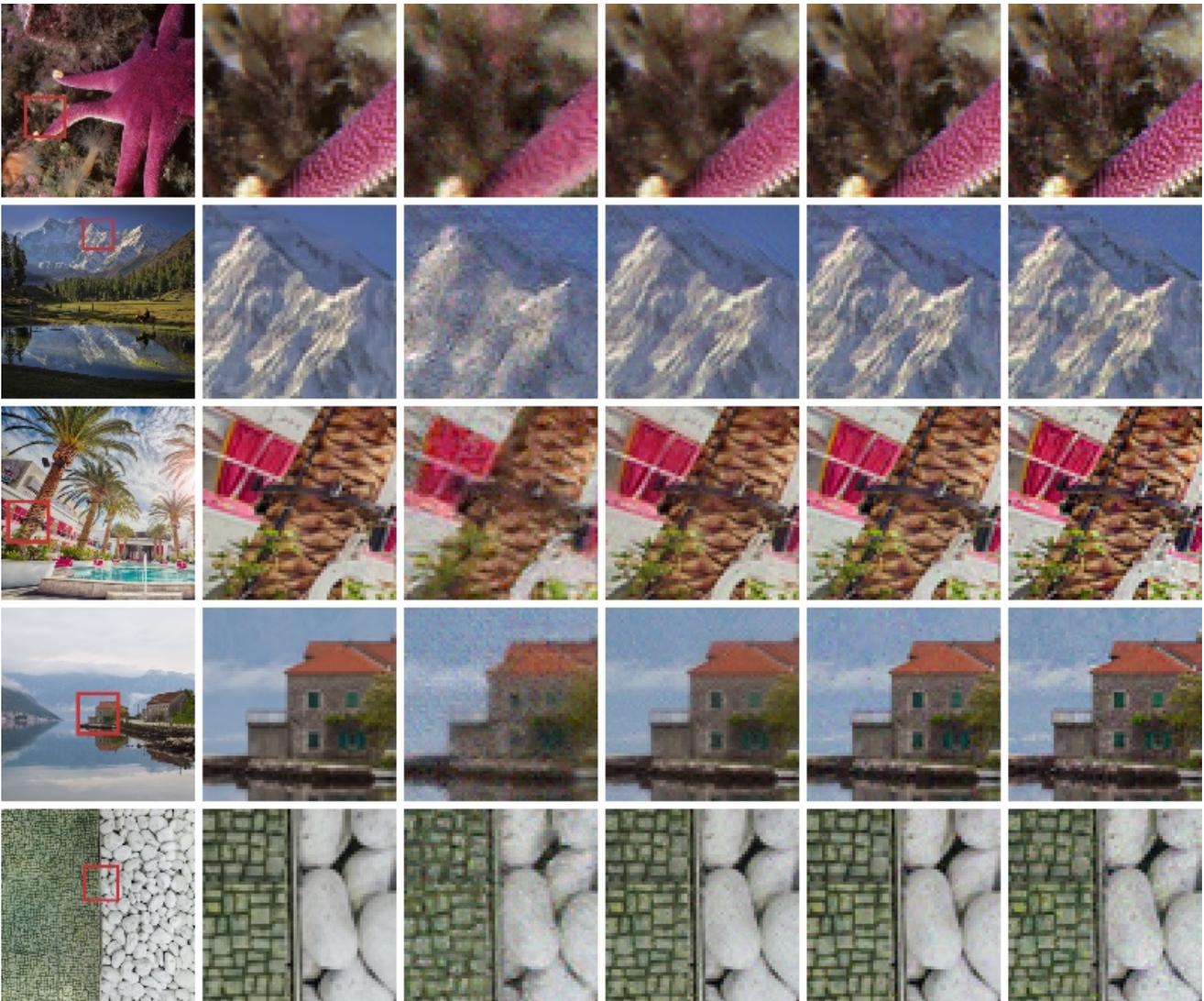
Impressionism [2]

DeFlow (ours)

Figure 4. AIM-RWSR: examples of clean inputs and corresponding synthetically degraded versions from different domain adaption methods.



Figure 5. NTIRE-RWSR: examples of noisy image patches.



Clean Input

CycleGAN [3]

Frequency Separation [1]

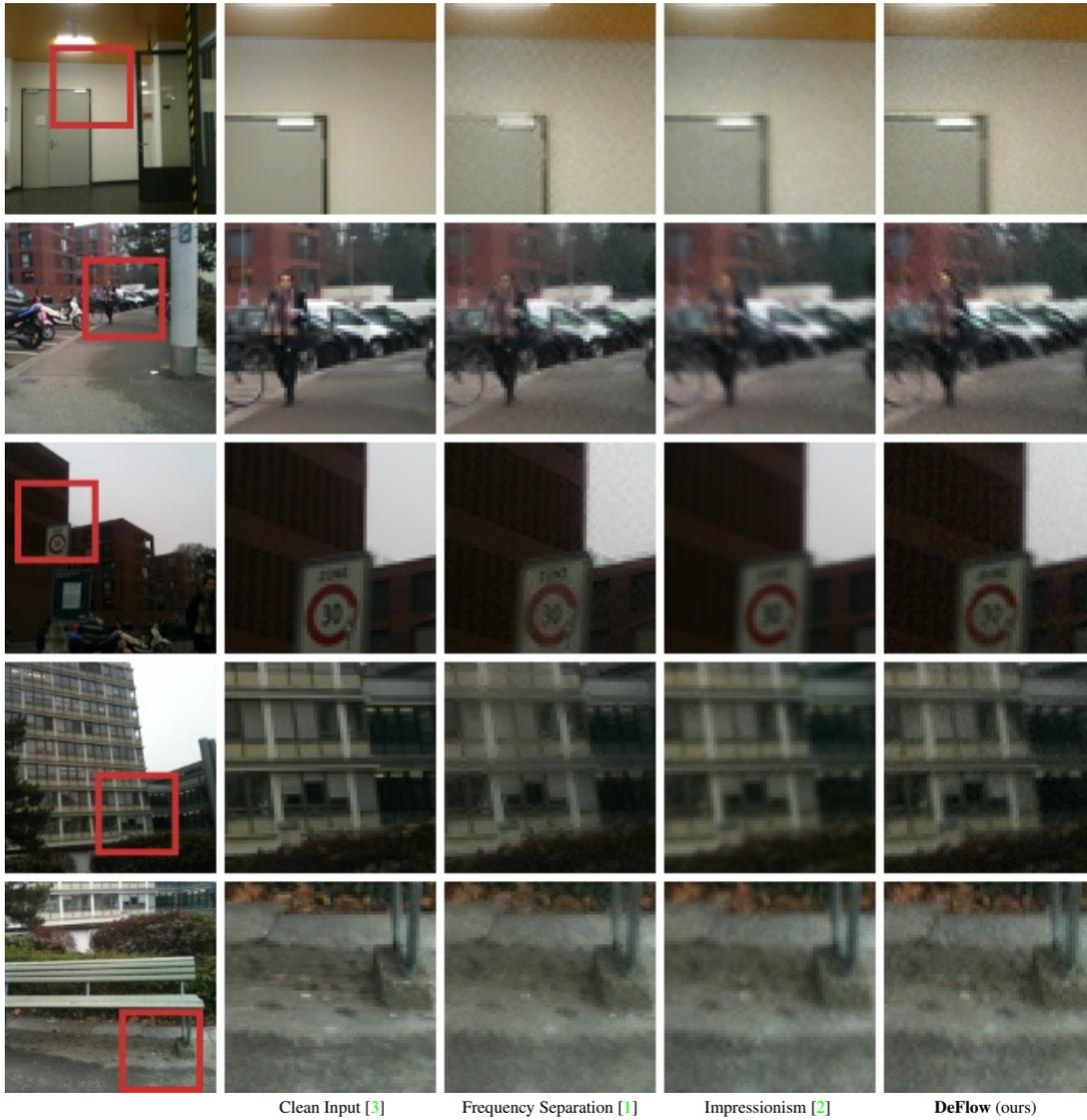
Impressionism [2]

DeFlow (ours)

Figure 6. NTIRE-RWSR: examples of clean inputs and corresponding synthetically degraded versions from different domain adaption methods.

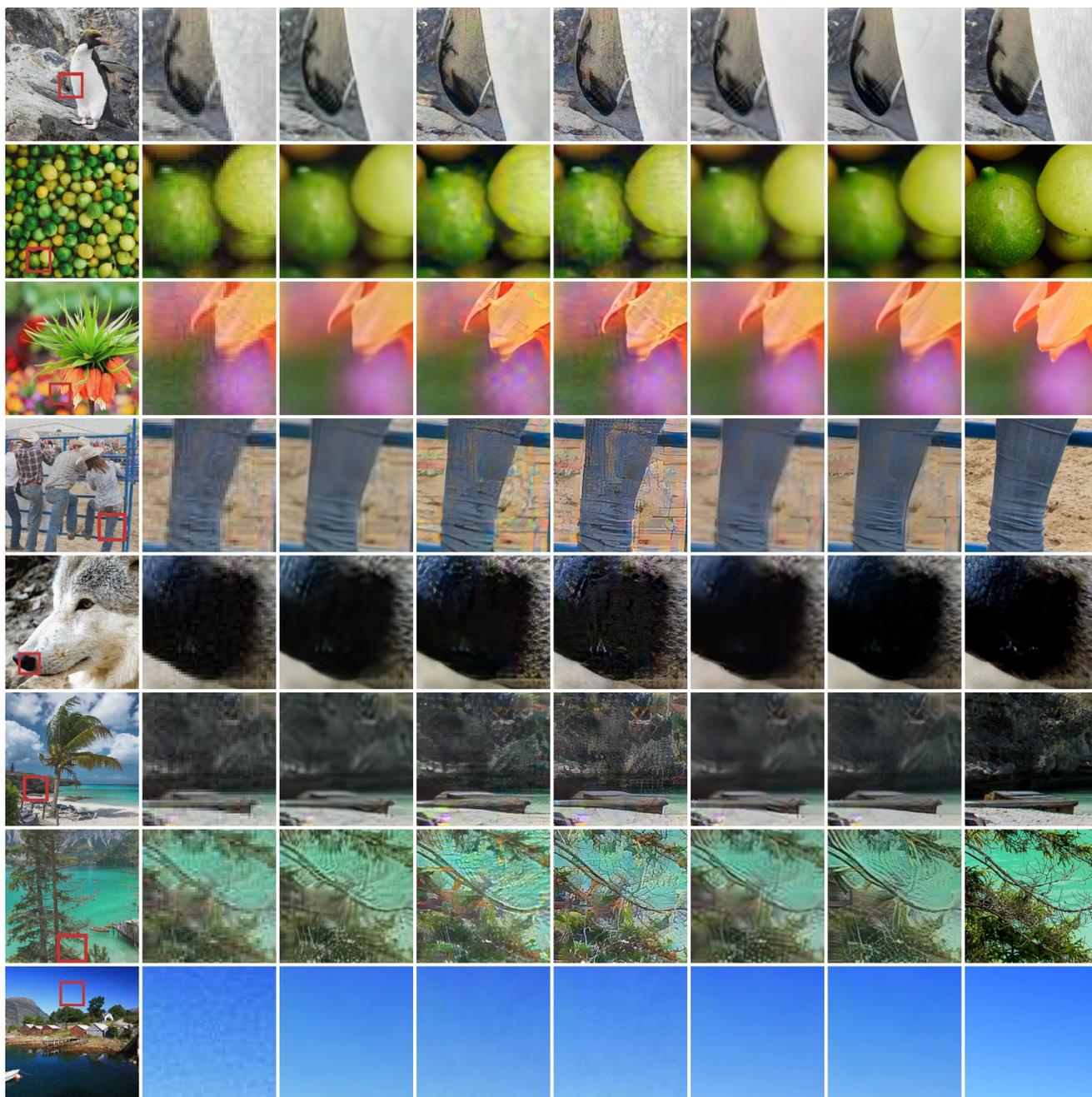


Figure 7. DPED-RWSR: examples of noisy image patches.



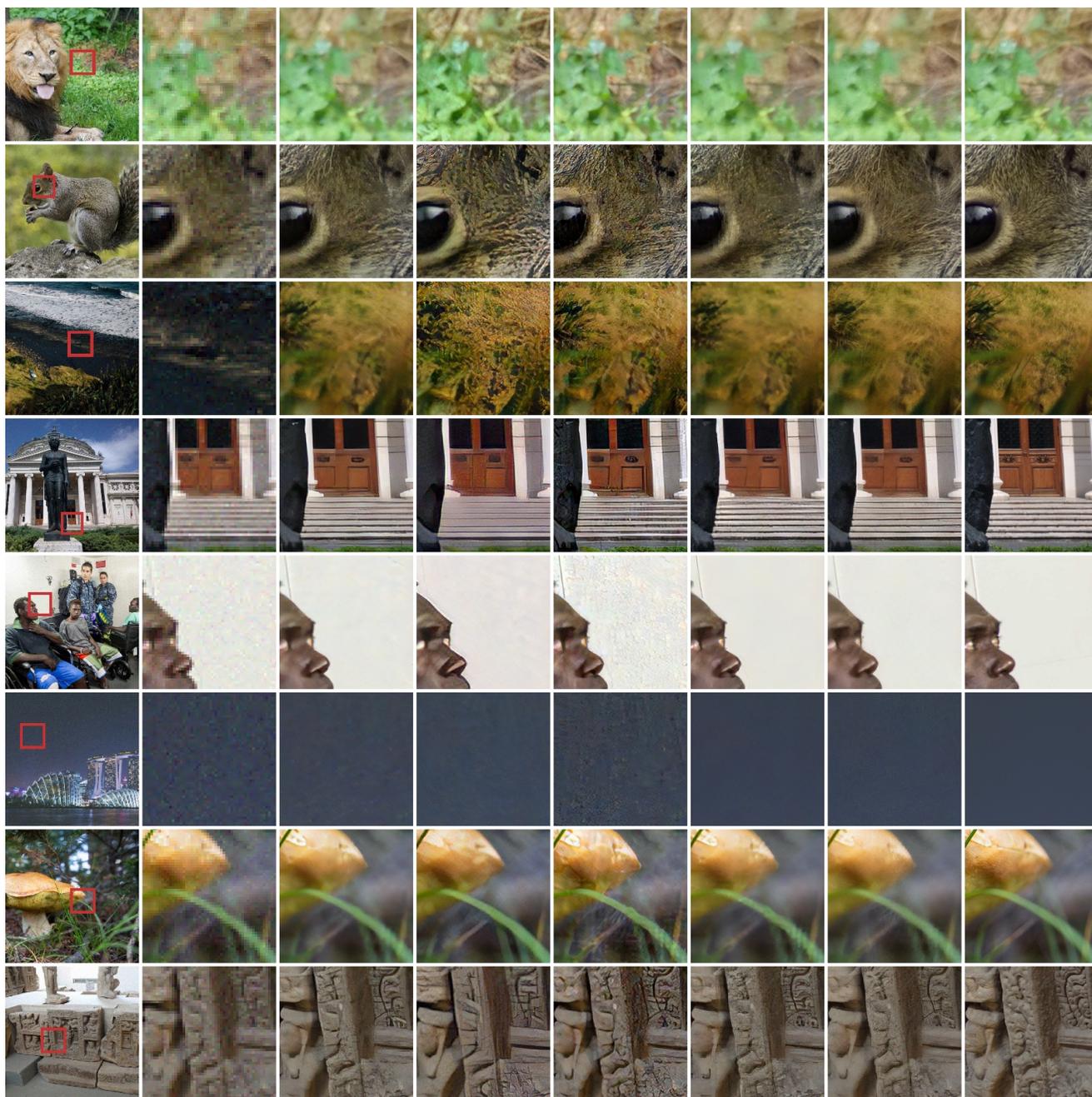
Clean Input [3]      Frequency Separation [1]      Impressionism [2]      **DeFlow (ours)**

Figure 8. DPED RWSR: examples of clean inputs and corresponding synthetically degraded versions from different domain adaption methods. Note, that we did not include CycleGAN [3] as differing to the other approaches it is trained to degrade images from DIV2k with DPED noise instead of down-sampled DPED images.



LR      White Noise  $\sigma=0.04$       DASR<sup>†</sup> [5]      Frequency Separation<sup>†</sup> [1]      Impressionism<sup>†</sup> [2]      **DeFlow (ours)**      GT

Figure 9. AIM-RWSR: Super-resolution results on the validation set. Methods marked with † employ the same SR pipeline as DeFlow and the baselines. Crops were chosen at random for an unbiased comparison.



LR      White Noise  $\sigma \sim \mathcal{U}(0, 0.06)$       DASR<sup>†</sup> [5]      Frequency Separation<sup>†</sup> [1]      Impressionism<sup>†</sup> [2]      DeFlow (ours)      GT

Figure 10. NTIRE-RWSR: Super-resolution results on the validation set. Methods marked with † employ the same SR pipeline as DeFlow and the baselines. Crops were chosen at random for an unbiased comparison.

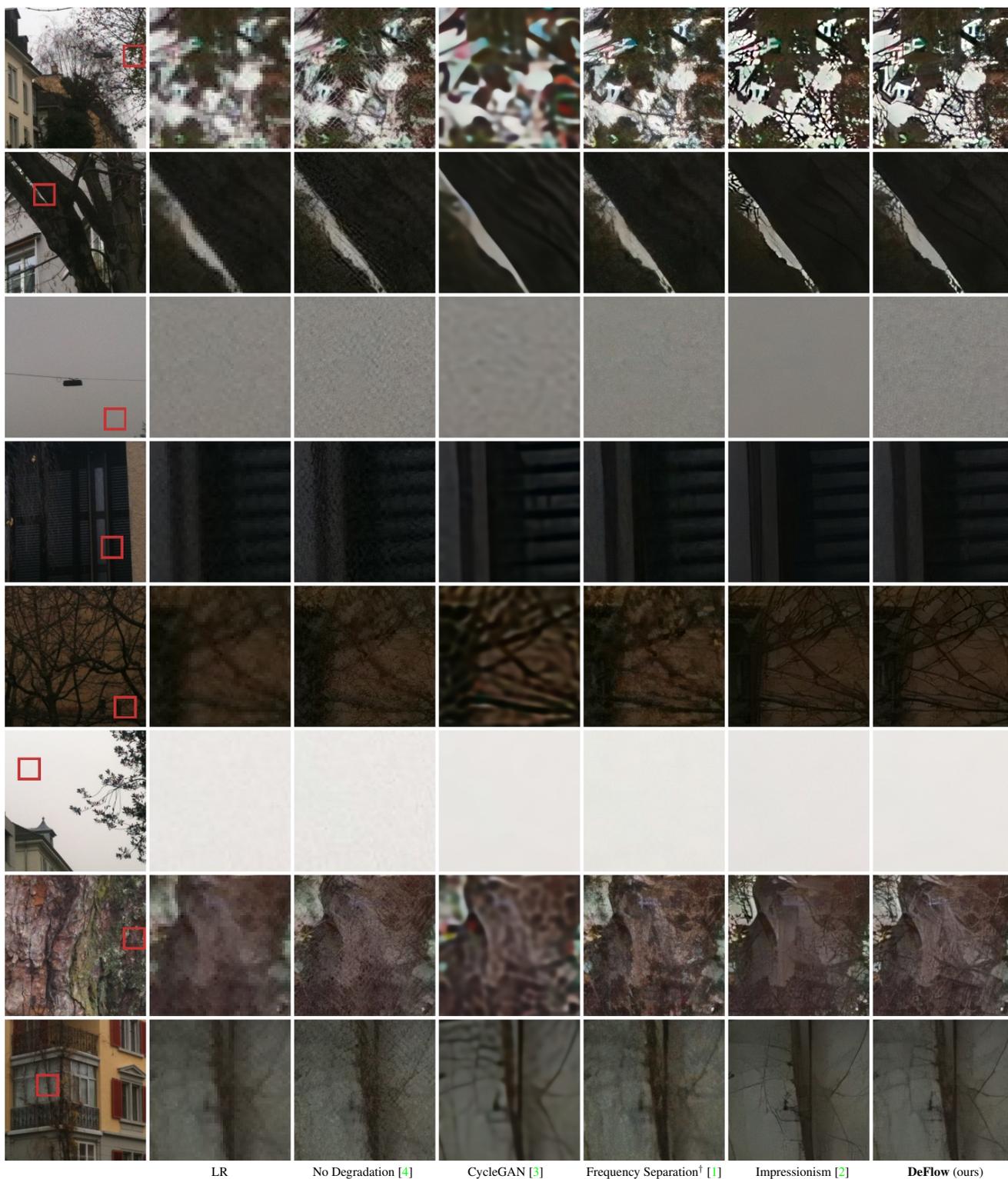


Figure 11. DPED-RWSR: Super-resolution results on the validation set. Ground truth reference images do not exist for this dataset as it consists of real-world low-quality smartphone photos. Methods marked with † employ the same SR pipeline as DeFlow and the baselines. Crops were chosen at random for an unbiased comparison.