## Supplementary Material

**Context-aware Pretraining for Efficient Blind Image Decomposition**

Chao Wang\(^1,2\), Zhedong Zheng\(^3\), Ruijie Quan\(^1\), Yifan Sun\(^2\), Yi Yang\(^1\†
\(^1\)ReLER, CCAI, Zhejiang University
\(^2\)Baidu Inc.
\(^3\)Sea-NExT Joint Lab, School of Computing, National University of Singapore

### 1. Implementation Details

**Appearance Flow Sampling:** As described in Section 3.2, we adopt the similar Gaussian sampling strategy [13] to avoid the bad local minima problem. The sampling process can be written as follow:

\[
F_{out} = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i,j} F_{i,j},
\]

where \(n\) is the kernel size. \(F_{i,j}\) represents the features around the sample center and \(F_{out}\) is the output feature. \(\alpha_{i,j}\) is the sampling weight calculated as:

\[
\alpha_{i,j} = \exp \left( \frac{-\Delta h^2 + \Delta v^2}{2\sigma^2} \right),
\]

where \(\Delta h\) and \(\Delta v\) are the horizontal and vertical distance between the sampling center and feature \(F_{i,j}\), respectively. \(\sigma\) is the variance of the sampling kernel.

**Noise Construction:** During the pre-processing stage, we randomly add different combinations of degradation to the clean images. For a fair comparison, we use the same setting for raindrop construction as BIDeN [5]. The model for rainstreak and snow is:

\[
I_{\text{rain/snow}}(x) = J(x)(1 - m(x)) + A \ast m(x),
\]

and the model for haze is:

\[
I_{\text{haze}}(x) = J(x)t(x) + A \ast (1 - t(x)),
\]

where \(x\) is the image pixel, \(J\) is the observed intensity, \(J\) is the scene radiance, \(A\) is the global atmospheric light, and \(m\) is the mask of rain streak and snow. \(t\) denotes the transmission map.

The model for reflection is formulated as:

\[
I_{\text{reflection}}(x) = T(x) + R(x) \ast V(x),
\]

and the model for watermark composition is:

\[
I_{\text{watermark}}(x) = J(x)(1 - w(x)) + A \ast w(x),
\]

where \(T\) is the transmission layer, \(R\) is the reflection layer, and \(V\) denotes the vignette mask. \(J\) is the scene radiance, \(A\) is the global atmospheric light, and \(w\) represents the watermark image.

As for shadow removal, we use the same setting on SRD [12] dataset with the masks generated by DHAN [3]. The reflection image \(R\) is processed by a Gaussian smoothing kernel with a random kernel size, where the size is in the range of 3 to 17 pixels during training, and fixed to 11 pixels during testing. For both Eqs. (9), (10) and (12), we set \(A\) between [0.8, 1.0] during training, and fix \(A = 0.9\) for testing. Before the random combination, all masks are randomly rotated with angles \([0^\circ, 90^\circ, 180^\circ, 270^\circ]\) for more robust pretraining.

**Dataset & Experimental Settings:** We compared the proposed CPNet mainly with Restormer [20], All-in-one [7], BIDeN [5], MPRNet [21], RCDNet [15], DHAN [3] and Auto-Exposure [4] method. The performance of the compared methods is acquired through former publicly available pretrained models or implementation codes *1 2 3 4 5 6*. Note that since the original code of All-in-one [7] is not publicly available, the qualitative results are based on our implementation. For a fair comparison, we trained and tested all methods under the same setting as BIDeN [5] with the same mask dataset.

In particular, Task-I is based on the CityScape [2] dataset, where we use the original test set as our training set (2975), and the validation set as our test set (500). The test set for all source components contains a fixed number of 500 images. The mask dataset contains four different masks with various intensities, including rainstreak (1620), rain-

\(^1\)https://github.com/lefthoman/Restormer
\(^2\)https://github.com/JunlinHan/BID
\(^3\)https://github.com/swa38/MPRNet
\(^4\)https://github.com/hongwang01/RCDNet
\(^5\)https://github.com/vinthony/ghost-free-shadow-removal
\(^6\)https://github.com/tsinqqguo/exposure-fusion-shadow-removal
\(^*\) Work done during an internship at Baidu.
\(^†\) Corresponding author: Yi Yang.
Table 6. Quantitative results of several structural variants on raindrop + snow removal on CityScape [2] dataset. The best performances under each case are marked in bold. (S)W-MSA represents the two successive Swin-Transformer blocks [10].

<table>
<thead>
<tr>
<th>Module</th>
<th>Vanilla Encoder</th>
<th>BIDeN</th>
<th>MHA (S)W-MSA</th>
<th>MDTA</th>
<th>FFN</th>
<th>Ours MDTA+GDFN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR ↑</td>
<td>28.71</td>
<td>29.16</td>
<td>32.25</td>
<td>31.77</td>
<td>31.15</td>
<td>32.02</td>
</tr>
<tr>
<td>SSIM ↑</td>
<td>0.822</td>
<td>0.881</td>
<td>0.918</td>
<td>0.887</td>
<td>0.864</td>
<td>0.910</td>
</tr>
<tr>
<td>FLOPs ↓</td>
<td>325G</td>
<td>344G</td>
<td>216G</td>
<td>90G</td>
<td>67G</td>
<td>102G</td>
</tr>
</tbody>
</table>

For raindrop masks, we model the droplet shape and property using the meta-ball model [1], following the same setting as BIDeN [3]. Haze masks are acquired from Foggy CityScape [14] with three different intensities, while the masks for snow are selected from Snow100K [9].

As for Task-II, the training set contains 3661 natural images as ground truth, where 861 images are adopted from the training set of [11]. 1800 images are borrowed from the training set of Rain1800 [17], and the rest 1000 images are selected from the training set of Snow100K [9]. The training masks are identical to Task-I. After training under these three cases, these models are respectively tested on three task-specific real-world datasets, which consist of 1000 images from SPAData [16] test set for rainstreak, 249 images from DeRaindrop [11] for raindrop, and 1329 images from DeSnowNet [9] for snow removal.

As for Task-III, we select the shadow dataset from [5]. The dataset is based on SRD [12], which consists of 2680 paired shadow masks, shadow-free images, and shadow images as the training set, and the test set contains 408 images for each type of degradation. The shadow masks are generated with DHAN [3] method. The algorithm for adding reflection to images is acquired from [22], we select 3120 images from the reflection subset as the reflection layer. We impose further supervised constraint on image details such as texture, which is deployed as:

$$\mathcal{L}_{\ell_2}^{t} = \|I_{\text{gen}} - I_{\text{gt}}\|_2,$$

where $I_{\text{gen}}$ is the predicted images and $I_{\text{gt}}$ means the ground truth. Meanwhile, we adopt a default perceptual loss [6] with a pretrained VGG16, which consists of two parts, i.e., the content loss $\mathcal{L}_{c}^{t}$ and style loss $\mathcal{L}_{s}^{t}$:

$$\mathcal{L}_{c}^{t} = \sum_{i=1}^{N} \frac{1}{HW} \left\| \phi_{\text{pool}}^{t} - \phi_{\text{pool}}^{\text{gen}} \right\|_1,$$

$$\mathcal{L}_{s}^{t} = \sum_{i=1}^{N} \frac{1}{C_{F} + C} \left\| \frac{1}{HW} \left( \phi_{\text{pool}}^{t} - \phi_{\text{pool}}^{\text{gen}} \right) \right\|_1,$$

where $\phi_{\text{pool}}^{t}$ represents the feature map after $pool_{i}$ layer, $\phi_{\text{pool}}^{\text{gen}} = \phi_{\text{pool}}^{t}$, and $N$ is the number of feature maps. The overall training loss for the refinement network $\mathcal{N}_r$ is $\mathcal{L}_{\ell_2}^{t} + \lambda_1 (\mathcal{L}_{c}^{t} + \mathcal{L}_{s}^{t})$, $\lambda_1$ denotes the hyper-parameter which are set as 0.01 in this paper.

![Figure 10. Architecture of the texture refinement network.](image-url)
Table 7. Quantitative comparisons on different fine-tuning strategies. \(N_r\) is the texture refinement network. MHP represents the multi-head prediction module.

<table>
<thead>
<tr>
<th>Finetuning</th>
<th>Input</th>
<th>MPRNNet</th>
<th>(N_r)</th>
<th>(N_r + ) MHP</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIQE ↓</td>
<td>4.87</td>
<td>4.10</td>
<td>4.13</td>
<td>4.02</td>
</tr>
<tr>
<td>BRISQUE ↓</td>
<td>27.82</td>
<td>28.66</td>
<td>25.58</td>
<td>25.11</td>
</tr>
</tbody>
</table>

Table 8. Detailed architectures of the proposed modules, in which \(H_s\) means the appearance head, \(H_t\) is the structure head, \(D_{img}\) is the discriminator judging whether the image is true or false, and \(D_{att}\) represents the discriminative branch predicting the attribute label of degradation type with dimension \(n\). \(\text{Conv}(c,k,s)\) denotes a standard convolutional layer with channel \(c\), kernel size \(k\) and stride \(s\). \(\text{GConv()}\) represents the gated convolution layer [18] and \(\sigma\) is the sigmoid activation.

<table>
<thead>
<tr>
<th>(H_t)</th>
<th>(H_s)</th>
<th>(D_{img})</th>
<th>(D_{att})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{GConv}(256,3,1),BN,\text{GELU})</td>
<td>(\times 3) (\text{GConv}(256,3,1),BN,\text{GELU})</td>
<td>(\times 3) (\text{Conv}(64,3,2),\text{IN,RELU})</td>
<td>(\text{Conv}(256,3,2),\text{IN,RELU})</td>
</tr>
<tr>
<td>with skip connection</td>
<td>with skip connection</td>
<td>(\text{Conv}(256,3,2),\text{IN,RELU})</td>
<td>(\text{Conv}(256,3,2),\text{IN,RELU})</td>
</tr>
<tr>
<td>(\text{GConv}(256,3,1),BN,\text{GELU})</td>
<td>(\text{GConv}(256,3,1),BN,\text{GELU})</td>
<td>(\text{Conv}(512,3,2),\text{IN,RELU})</td>
<td>(\text{Conv}(512,3,2),\text{IN,RELU})</td>
</tr>
<tr>
<td>(\text{GConv}(128,3,1),BN,\text{GELU})</td>
<td>(\text{GConv}(128,3,1),BN,\text{GELU})</td>
<td>(\text{Conv}(1024,3,2),\text{IN,RELU})</td>
<td>(\text{Conv}(1024,3,2),\text{IN,RELU})</td>
</tr>
<tr>
<td>(\text{GConv}(64,3,1),BN,\text{GELU})</td>
<td>(\text{GConv}(64,3,1),BN,\text{GELU})</td>
<td>(\text{FC}(1024),\text{IN,RELU})</td>
<td>(\text{FC}(1024),\text{IN,RELU})</td>
</tr>
<tr>
<td>(\text{GConv}(32,3,1),BN,\text{tanh})</td>
<td>(\text{GConv}(32,3,1),BN,\text{tanh})</td>
<td>(\text{FC}(1))</td>
<td>(\text{FC}(n), \sigma)</td>
</tr>
</tbody>
</table>

As shown in Table 7, we further unlock the Multi-Head Prediction (MHP) module along with refinement network \(N_r\) for fine-tuning, leading to a higher performance since the gradients from the textual information can directly guide the training of the appearance flow map. We also observe a marginal performance improvement with a more complex refinement network, which further indicates the effectiveness of our pre-trained model. We also show the detailed architecture of the multi-head prediction module as well as the multi-head discriminator in Table 8.

3. More Results

Figure 11 and Figure 12 show more qualitative comparisons and results for Task-I. More results on real-world Task-II are given in Figure 13, Figure 14 and Figure 15. Figure 16 and Figure 17 show the results for Task-III under seven different noise combinations. It can be seen that our method stably produces well-structured results with finer details while remaining robust to different noise combinations and training strategies.
Figure 12. More qualitative results of the proposed CPNet under each case in Task-I. Case (1): rain streak, (2): rain streak + snow, (3): rain streak + light haze, (4): rain streak + heavy haze, (5): rain streak + moderate haze + raindrop, (6) rain streak + snow + moderate haze + raindrop. Every two rows represent one case. Please zoom in to see the details.
Figure 13. More qualitative comparisons on real-world rainstreak removal scenario. BIDeN(3) represents the model that is jointly trained with three noise combinations: rainstreak + raindrop + snow. Please zoom in to see the details.
Figure 14. More qualitative comparisons on real-world raindrop removal scenario. Please zoom in to see the details.
Figure 15. More qualitative comparisons on real-world snow removal scenario. Please zoom in to see the details.
Figure 16. More qualitative results on Task-III. Please zoom in to see the details.
Figure 17. More results on Task-III with different noise combinations. Please zoom in to see the details.
References


    R. Benenson, U. Franke, S. Roth, and B. Schiele. The 
    cityscapes dataset for semantic urban scene understanding. 
    In *CVPR*, 2016.

    shadow removal via dual hierarchical aggregation network 


[5] J. Han, W. Li, P. Fang, C. Sun, J. Hong, M. A. Armin, L. 
    Petersson, and H. Li. Blind image decomposition. In *ECCV*, 
    2022.


[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and 
    B. Guo. Swin transformer: Hierarchical vision transformer 

    generative adversarial network for raindrop removal from a 

    ownet: A multi-context embedding deep network for shadow 


