This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# ABPN: Adaptive Blend Pyramid Network for Real-Time Local Retouching of Ultra High-Resolution Photo

Biwen Lei†, Xiefan Guo*, Hongyu Yang, Miaomiao Cui†, Xuansong Xie†, Di Huang
†DAMO Academy, Alibaba Group
biwen.lbw@alibaba-inc.com, {guoxiefan, hongyu.yang.cv}@gmail.com,
miaomiao.cmm@alibaba-inc.com, xingtong.xxs@taobao.com, dhuang.cv@outlook.com

## Abstract

*Photo retouching finds many applications in various fields. However, most existing methods are designed for global retouching and seldom pay attention to the local region, while the latter is actually much more tedious and time-consuming in photography pipelines. In this paper, we propose a novel adaptive blend pyramid network, which aims to achieve fast local retouching on ultra high-resolution photos. The network is mainly composed of two components: a context-aware local retouching layer (LRL) and an adaptive blend pyramid layer (BPL). The LRL is designed to implement local retouching on low-resolution images, giving full consideration of the global context and local texture information, and the BPL is then developed to progressively expand the low-resolution results to the higher ones, with the help of the proposed adaptive blend module and refining module. Our method outperforms the existing methods by a large margin on two local photo retouching tasks and exhibits excellent performance in terms of running speed, achieving real-time inference on 4K images with a single NVIDIA Tesla P100 GPU. Moreover, we introduce the first high-definition cloth retouching dataset CRHD-3K to promote the research on local photo retouching. The dataset is available at https://github.com/youngLBW/CRHD-3K.*

## 1. Introduction

Photo retouching [25], especially portrait photo retouching, finds a vast range of applications in photography scenarios including wedding, advertisement, personal recording, etc. While extensive works [5, 12, 14, 21, 46, 57] yield impressive results on photo retouching, most of them manipulate the attributes of the entire image, such as color, illumination, and exposure. Few methods deal with the local region in photos (*e.g.*, face, clothing, and commodity), which is actually the most tedious and time-consuming step in professional photography pipelines.

(a) Input (b) Ours (c) Target

Figure 1. High-fidelity retouched photos. From left to right: (a) raw photos, (b) our retouched results, and (c) ground-truth images.

To focus on this kind of problem, we summarize them as the **L**ocal **P**hoto **R**etouching (LPR) task, whose goal is to edit the target region in the photo and keep the rest area unchanged. Different from general local image editing tasks (such as image inpainting and rain removal), LPR pays more attention to enhancing the aesthetic perception and visual quality of the target object. Fig. 1 gives some LPR examples.

We conclude three main challenges of the LPR task as: (1) accurate localization of the target region; (2) local generation with global consistency and detail fidelity; and (3) efficient processing of ultra high-resolution images. The first two are brought by the characteristics of the task itself, while the last one is determined by the application scenarios of LPR. As ultra high-resolution photos have been widely used in various photographic scenes, the ability to process them becomes a key factor of LPR methods in practice. Given these challenges above, we in this paper analyze the applicability of existing methods to the LPR task and attempt to propose a more suitable solution to it.

*This work was done while Xiefan Guo was an intern at the DAMO.

In recent years, massive works have devoted to the image-to-image translation task and achieve impressive results in style transfer [11, 16, 19, 45], semantic image synthesis [7, 18, 37], etc. Most of them adopt a deep network with an encoding-decoding paradigm to fulfill faithful translation, which results in a heavy computational, thus severely limiting their applications in some high-resolution scenarios. Some methods [12, 25, 47, 52] try to accelerate the models by transferring the computational burden from high-resolution maps to low-resolution ones and successfully accomplish global translation on high-resolution images. However, due to the lack of attention to local regions, few of them well adapt to the LPR task.

Instead of performing global translation, a number of works focus on the local image editing task, such as image inpainting [28, 39, 55], shadow removal [15, 32, 33], and rain removal [40–42, 48, 49]. Most of them rely on the masks that indicate the target region as input, while in the LPR task, accurately acquiring such masks is itself a quite challenging issue. Though some methods resort to the deep generative networks and perform local editing without specifying the masks, they are hardly capable of processing ultra high-resolution images directly. Besides, AutoRetouch [46] employs a sliding window strategy to achieve local modeling and retouching, but it fails to capture the global context, especially in the case of high resolution.

Based on the observations, we propose a novel adaptive blend pyramid network (ABPN) for local retouching of ultra high-resolution photos, as shown in Fig. 3. The network addresses the three challenges aforementioned via two components: a context-aware local retouching layer (LRL) and an adaptive blend pyramid layer (BPL). In general, given a high-resolution image, the LRL performs local retouching on its thumbnail and the subsequent BPL expands the outputs of LRL to the original size of the input. For LRL, specifically, we design a novel multi-task architecture to fulfill mask prediction of the target region and local generation simultaneously. A local attentive module (LAM) is proposed, where the local semantics and texture of the target region and the global context can be fully captured and aggregated to achieve consistent local retouching. For BPL, inspired by the *blend layer* in digital image editing, we develop a light-weight adaptive blend module (ABM) and its reverse version (R-ABM) to implement the fast expansion from the low-resolution results to the higher ones, ensuring great extensibility and detail fidelity. Extensive experiments on two LPR tasks reveal that our method outperforms the existing methods by a large margin in terms of retouching quality and processing efficiency, demonstrating its superiority in the LPR task.

Moreover, since the editing work is usually time-consuming and requires high image processing skills, there are few publicly available datasets for the LPR task. Accordingly, we build and release the first high-definition cloth retouching dataset (CRHD-3K) to facilitate the research.

Our main contributions in this work are as follows:

**(A)** We propose a novel framework ABPN for local retouching of ultra high-resolution photos, which exhibits the remarkable efficiency performance (real-time inference on 4K images with a single NVIDIA Tesla P100 GPU) and superior retouching quality to the existing methods.

**(B)** We present a local attentive module (LAM), which is effective in capturing and aggregating the global context and local texture.

**(C)** We design an adaptive blend module (ABM), which provides powerful extensibility to the framework, allowing the fast expansion from low-resolution results to the higher ones.

**(D)** To boost the research on LPR (*e.g.*, cloth retouching), we introduce the first high-definition cloth retouching dataset CRHD-3K.

## 2. Related Work

**Photo Retouching.** Benefiting from the development of deep convolutional neural networks, learning-based methods [5, 10, 12, 14, 21, 46, 50, 57] have recently been presented to produce exciting results on photo retouching. Most of those, however, are limited by the heavy computational and memory costs when the photo resolution is increased. In addition, these methods are designed for global photo retouching and do not well fit for the LPR task.

**Image-to-Image Translation.** Image-to-image translation was originally defined by [18], in which many computer-vision tasks were summarized as a pixel-to-pixel predicting job and a conditional GANs-based framework was developed as a general solution. Following [18], various methods have been proposed to address the image translation problem, using paired images [7, 18, 27, 37, 43, 47, 52] or unpaired images [3, 8, 9, 16, 17, 23, 25, 30, 36, 38, 59]. Several works focus on a specific image translation task (such as semantic image synthesis [7, 18, 37] and style transfer [11, 16, 19, 45]) and achieve impressive performance. However, the works above mainly concentrate on global transformation and give less attention to the local region, which limits their capability in the LPR task.

**Image Inpainting.** Image Inpainting is the closest task to LPR, which refers to the process of reconstructing missing regions of an image given a corresponding mask. The deep generative methods [13, 22, 26, 28, 29, 35, 39, 51, 53–56, 58] have achieved significant progress, owing to their powerful feature learning ability. However, acquiring accurate masks is itself a very challenging issue, and taking unreasonable masks tends to incur large errors in filled results. Recently, the blind image inpainting methods [6, 31, 53] relax the restriction by completing the visual contents without specifying masks for missing regions. Nevertheless, those methods

Figure 2. Examples from the CRHD-3K Dataset (zoom in for a better view). *Left*: raw photos, *right*: retouched results by professional staffs with high image processing expertise.

assume the contamination with simple data distributions or undesired images, which makes them fail to take full advantages of the inherent semantics and textures of the image for LPR. Moreover, the existing methods can only handle low-resolution inputs, ultra high-resolution image inpainting is still extremely challenging. There are also some local image editing tasks that aim to restore the local region in the image, including shadow removal [15, 32, 33], rain removal [40–42, 48, 49], etc. Unfortunately, due to the strong specificity of these methods, few of them are adaptive for the common LPR task.

**High-resolution Image Editing.** To enable translation on high-resolution images, [12, 25, 47, 52] attempt to alleviate the space and time burden by shifting the major computation from high-resolution maps to low-resolution ones. Though yielding impressive efficiency performance, it is still problematic when applied to LPR as the lack of attention to the local regions.

## 3. The CRHD-3K Dataset

Photo retouching [24] refers to the process of enhancing the visual aesthetic quality of an image, and cloth retouching is one of the most representative tasks, which is conventionally achieved via hand-craft operations. However, the process of manual retouching is tedious and time-consuming. In order to facilitate the learning-based retouching methods, we introduce the first large-scale high-definition cloth retouching (CRHD-3K) dataset.

**Data collection.** We initially collected more than 60,000 raw photos from Unsplash[1], and further carefully checked them one by one, where outliers (*e.g.*, severe motion blur) and duplicates (*e.g.*, same content) were removed. The CRHD-3K dataset finally includes 3,022 high-definition raw portrait photos.

**Data labeling.** To obtain high-quality retouched photos, the process is accomplished by a team of professional image editors, with the goal of removing the wrinkles, creases, and other blemishes on the clothes to make them look more smooth and beautiful. The retouching time for each photo is 3 to 5 minutes. Some retouched examples are shown in Fig. 2.

[1]https://unsplash.dogedoge.com

**Data statistics.** The CRHD-3K dataset consists of 3,022 pairs of raw and retouched photos, of which 2,522 are for training and 500 for testing. The resolutions mainly vary in the range of 4K to 6K.

**Ethics guidelines.** To avoid the attendant risk of harm from the data, we blurred and cropped the personally identifiable information contained in the photos (*e.g.*, faces), and kept only the clothing components as much as possible.

Cloth retouching is a typical and quite challenging LPR task due to the diversity of clothing patterns and the subjectivity of wrinkle judgment. More importantly, ultra high-resolution images from the CRHD-3K dataset place extremely strict requirements on the time and space efficiency of the model.

## 4. Methods

### 4.1. Overview

As discussed above, subject to the lack of attention to local regions or the high computational costs, the existing methods are difficult to cope with the LPR task. To solve these problems, we develop an adaptive blend pyramid network for local retouching of ultra high-resolution photos. Fig. 3 shows an overview of our framework. The network is mainly composed of two components: a context-aware local retouching layer (LRL) and an adaptive blend pyramid layer (BPL). Given an image $\boldsymbol{I}_0 \in \mathbb{R}^{h\times w\times 3}$, we first build an image pyramid $\boldsymbol{P_I} = [\boldsymbol{I}_0, \boldsymbol{I}_1, \cdots, \boldsymbol{I}_l]$ and a high-frequency component pyramid $\boldsymbol{P_H} = [\boldsymbol{H}_0, \boldsymbol{H}_1, \cdots, \boldsymbol{H}_{l-1}]$, where $\boldsymbol{P_H}$ is acquired following Laplacian Pyramid [4] and $l$ is the number of downsampling operations ($l = 2$ as default in Fig. 3). Then LRL is applied to $\boldsymbol{I}_l \in \mathbb{R}^{\frac{h}{2^l}\times\frac{w}{2^l}\times 3}$ to predict the target region mask $\boldsymbol{M}$ and generate the retouched results $\boldsymbol{R}_l \in \mathbb{R}^{\frac{h}{2^l}\times\frac{w}{2^l}\times 3}$. After that, we employ BPL to expand the low-resolution outputs $\boldsymbol{R}_l$ to the original size of $\boldsymbol{I}_0$. Specifically, the reverse adaptive blend module (R-ABM) is introduced to generate the blend layer $\boldsymbol{B}_l \in \mathbb{R}^{\frac{h}{2^l}\times\frac{w}{2^l}\times 3}$, which records the translation information from $\boldsymbol{I}_l$ to $\boldsymbol{R}_l$. By progressively upsampling and refining, the blend layer $\boldsymbol{B}_0$ with high resolutions and abundant details is obtained. At last, we utilize the adaptive blend module (ABM) to apply $\boldsymbol{B}_0$ to $\boldsymbol{I}_0$ to generate the final results $\boldsymbol{R}_0$.

We introduce these sub-networks and loss functions used

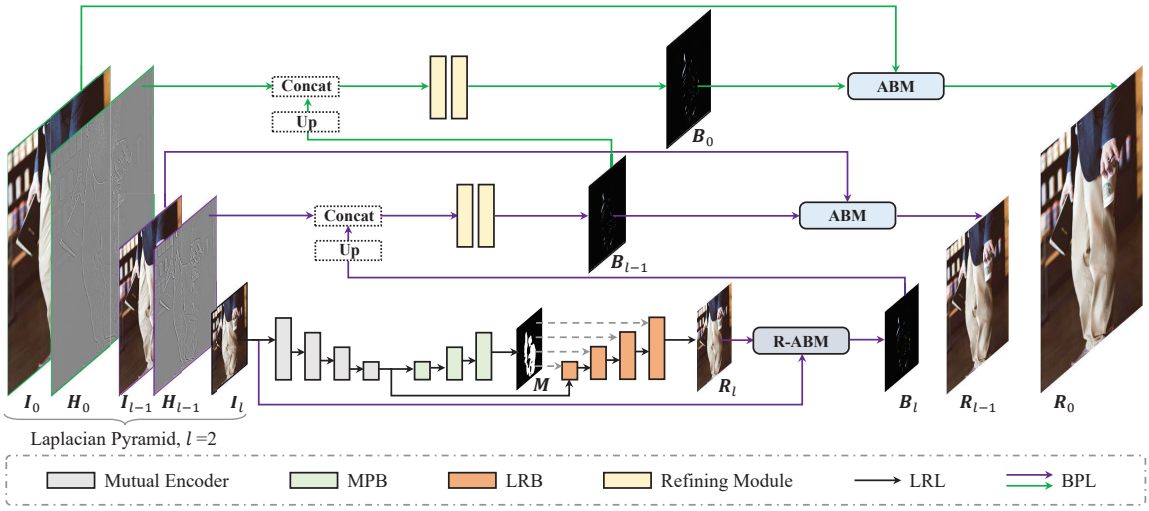



Figure 3. Overview of the proposed Adaptive Blend Pyramid Network (ABPN).

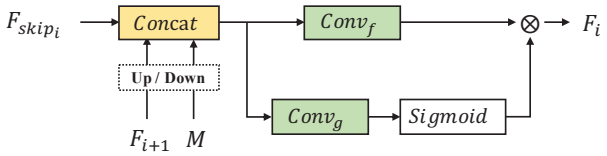



Figure 4. The details of the local attentive module (LAM).

for training in detail in the following sections, including LRL in Sec. 4.2, BPL in Sec. 4.3, and loss functions in Sec. 4.4.

### 4.2. Context-aware Local Retouching Layer

In this section, we propose a context-aware local retouching layer (LRL) to address the first two challenges mentioned in Sec. 1: accurate localization of the target region and local generation with global consistency. As shown in Fig. 3, the LRL adopts a multi-task architecture and consists of a mutual encoder, a mask prediction branch (MPB) and a local retouching branch (LRB).

**Mutual Encoder.** The mutual encoder is composed of six simple convolution blocks ($3 \times 3$ convolutions, batch normalization, and ReLU) in series, and the output of each convolution block composes a feature pyramid $P_F = [\boldsymbol{F}_{skip_i} \in \mathbb{R}^{\frac{h}{2^{l+i}} \times \frac{w}{2^{l+i}} \times c_i}]_{i=0}^{6}$, where $c_i$ denotes the number of channels and $\boldsymbol{F}_{skip_0} = \boldsymbol{I}_0$. Sharing the encoder with the subsequent MPB and LRB is feasible because both of the two branches rely on the semantic features and contextual information to generate their results. It also greatly reduces the computational complexity of the model.

**Mask Prediction Branch.** MPB aims to automatically predict the mask $\boldsymbol{M} \in \mathbb{R}^{\frac{h}{2^{l+2}} \times \frac{w}{2^{l+2}} \times 1}$ of the target region to guide subsequent local region generation. It consists of four convolution blocks ($3 \times 3$ convolutions, batch normalization, and leakyReLU) and a classification head. Besides, we employ skip connections [44] to incorporate low-level features to improve the accuracy of segmentation. Note that $\boldsymbol{M}$ is $4\times$ smaller than $\boldsymbol{I}_l$ but it is sufficient for the guidance of LRB, without sacrifice to the overall performance. Although most of the datasets do not directly provide the target region mask $\boldsymbol{M}_{gt}$ for supervision, owing to the characteristics of the LPR task, we can obtain the $\boldsymbol{M}_{gt}$ by taking a difference between $\boldsymbol{I}_l$ and its target and applying a threshold to it.

The contributions of MPB to the network are two-fold. First, the predicted mask $\boldsymbol{M}$ itself can help LRB focus on the target region to enhance the retouching quality. Second, through joint training, the global context and semantic information can be better perceived, thereby achieving consistent generation results.

**Local Retouching Branch.** Most image translation methods adopt a traditional encoder-decoder architecture to implement global translation, which leads to insufficient attention to the target regions. Based on the gated convolution (GConv) [55], we thus design a local attentive module (LAM) to improve capturing local semantics and textures, as shown in Fig 4. Different from image inpainting, the target region in LPR contains rich texture information, which is essential to generate detailed and realistic results. In this case, we apply skip connections to incorporate low-level features $\boldsymbol{F}_{skip_i}$ from the mutual encoder. Besides, instead of only involving the binary mask in the first or the last block of LRB, we concatenate the soft mask $\boldsymbol{M}$ in every LAM to guide feature fusion and update at different levels. Owing to the gating mechanism of GConv, spatial attention and channel attention are simultaneously employed to fully fuse the features and capture the semantics and textures of the target region. By stacking LAM, LRB is then able to produce consistent and faithful local retouched results. Note that although the predicted mask may have errors, the final retouching area could still not be affected as the mask is only used as soft guidance in LRB.

### 4.3. Adaptive Blend Pyramid Layer

LRL achieves local retouching on a low-resolution image, and the following objective is to extend the result to a larger scale while simultaneously enhancing its detail fidelity. Inspired by the concept of *blend layer* (or *top layer*) in the digital image editing, we propose an adaptive blend module (ABM) and its reverse one (R-ABM) to achieve lossless transformation between two images with a sparse and smooth blend layer. Then, we build a pyramid to progressively upsample and refine the blend layer and finally apply it to the original input to generate the final result. We describe the implementation details of these components below.

**Adaptive Blend Module.** The blend layer is often utilized to be blended with the image (or *base layer*) in various modes [1] to fulfill different image editing tasks, such as contrast adjustment, dodge and burn. Generally, given an input image $\boldsymbol{I} \in \mathbb{R}^{h\times w\times 3}$ and a blend layer $\boldsymbol{B} \in \mathbb{R}^{h\times w\times 3}$, we blend the two layers to produce the result $\boldsymbol{R} \in \mathbb{R}^{h\times w\times 3}$ as:

$$\boldsymbol{R} = f(\boldsymbol{I}, \boldsymbol{B}) \tag{1}$$

where $f$ is a pixel-wise function and denotes the mapping formula determined by the blend mode. Limited by the translation ability, a certain blend mode with the fixed function $f$ is difficult to apply to various image editing tasks. To better adapt to the data distribution and the transformation patterns of different tasks, we refer to the Soft Light blend mode [2] and design an adaptive blend module (ABM) as follows:

$$g(\boldsymbol{I}, i) = \boldsymbol{E}\underbrace{\odot \boldsymbol{I} \odot \boldsymbol{I} \cdots \odot \boldsymbol{I}}_{i} \tag{2}$$

$$\boldsymbol{R} = f_a(\boldsymbol{I}, \boldsymbol{B}) = \sum_{i=0}^{2}((j_i\boldsymbol{B} + k_i\boldsymbol{E}) \odot g(\boldsymbol{I}, i)) \tag{3}$$

where $\odot$ indicates the Hadamard product, $j_i$ and $k_i$ are learnable parameters shared by ABMs and R-ABM in the framework, and $\boldsymbol{E} \in \mathbb{R}^{h\times w\times 3}$ denotes a constant matrix with the value $\mathbf{1}$ for all items.

**Reverse Adaptive Blend Module.** ABM is based on the prerequisite of the blend layer $\boldsymbol{B}$. However, we only obtain the low-resolution results $\boldsymbol{R}_l$ in the previous LRL. To acquire the blend layer $\boldsymbol{B}$, we solve Eq. (3) and build a reverse adaptive blend module (R-ABM) as:

$$\boldsymbol{B} = f_r(\boldsymbol{I}, \boldsymbol{R}) = \frac{\boldsymbol{R} - \sum_{i=0}^{2}(k_i g(\boldsymbol{I}, i))}{\sum_{i=0}^{2}(j_i g(\boldsymbol{I}, i))} \tag{4}$$

where $j_i$, $k_i$ and $g$ are consistent with those in Eq. (3).

In general, utilizing the blend layer as an intermediate medium, ABM and R-ABM offer an adaptive transformation between the image $\boldsymbol{I}$ and the result $\boldsymbol{R}$. Instead of directly expanding the low-resolution result, we employ the blend layer to achieve this goal, which has its advantages on two aspects: (1) In the LPR task, the blend layer mainly records local transformation between two images. That means it contains less irrelevant information and can be readily refined with a light-weight network. (2) The blend layer is to be blended with the original image to implement final retouching, which makes full use of the information of the image itself, thereby delivering local retouching with a high detail fidelity.

Actually, there are plenty of alternative functions or strategies to achieve adaptive blend. An intuitive way is to utilize two networks composed of $1 \times 1$ convolutions and nonlinear activation layers to replace Eq. (3) and Eq. (4) respectively. However, the transformations from the two networks are irreversible and may increase the difficulty in training. In contrast, good reversibility and consistency between ABM and R-ABM ensure that all the blend layers lie in the same domain, which effectively reduces the burden on the model. Moreover, Eq. (3) is a generalized form of the Pegtop's formula [2], which is easy to optimize and tends to produce a smooth and sparse blend layer (see Fig. 7 and Fig. 8). As in our framework, we fulfill the expansion by progressively upsampling and refining the blend layer. Smoothness and sparseness mean a smaller information gap between the low-resolution blend layer and its high-resolution target, which greatly eases the burden on the refining module. See experimental results toward ABM in Sec. 5.4 for its superiority.

ABM and R-ABM hold simple structures but fully consider the characteristics of the LPR task and provide powerful extensibility to the framework, facilitating fast expansion of the low-resolution results at a negligible cost.

**Refining Module.** To apply the low-resolution blend layers to high-resolution images, the refining module is essential to compensating the information loss caused by downsampling. Since the blend layer is initially generated from the low-resolution result, it is short of the transformation information of high-frequency components. We thus include the high-frequency component of the image as an additional input for the refining module. Owing to the smoothness and sparsity of the blend layer produced from R-ABM, we can build a light-weight refining module as:

$$\boldsymbol{B}_i = \phi_2(h(\phi_1(\text{Cat}(\text{up}(\boldsymbol{B}_{i+1}), \boldsymbol{H}_i)))) + \text{up}(\boldsymbol{B}_{i+1}) \tag{5}$$

where up denotes bilinear interpolation, Cat is channel-wise concatenation, $\boldsymbol{H}_i$ ($i \in \{0, 1, ..., l-1\}$) is the high-frequency component of image $\boldsymbol{I}_i$, $\phi_1$ and $\phi_2$ are $3 \times 3$ convolutions with 16 and 3 filters respectively, and $h$ indicates leaky ReLU with negative slop 0.2.

Given the input and output of LRL, we first adopt Eq. (4) to calculate a primitive blend layer $\boldsymbol{B}_l$. By continuously upsampling and refining the blend layer, we then obtain a high-resolution blend layer $\boldsymbol{B}_0$ with detailed transformation

(a) Input (b) VCNet (c) AutoRetouch (d) pix2pixHD (e) ASAPNet (f) LPTN (g) Ours (h) Target

Figure 5. Qualitative comparison on FFHQR and CRHD-3K (zoom in for a better view): (a) original images, (b) VCNet [53], (c) AutoRetouch [46], (d) pix2pixHD [52], (e) ASAPNet [47], (f) LPTN [25], (g) Ours, and (h) ground-truth images.

information. At last, Eq. (3) is applied to $\boldsymbol{B}_0$ and $\boldsymbol{I}_0$ to deliver the final result.

### 4.4. Loss Functions

The model is trained in an end-to-end manner, and the loss functions that we utilize for training consist of (i) the multi-scale mean squared-error (MSE) loss $\mathcal{L}_{mse} = \sum_{i=0}^{l} ||\boldsymbol{R}_{gt_i} - \boldsymbol{R}_i||_2^2$, (ii) the perceptual loss $\mathcal{L}_{perc}$ [19] is only applied to the low-resolution outputs $\boldsymbol{R}_l$ for saving training memory cost, (iii) the adversarial loss $\mathcal{L}_{adv}$ [18] for the final outputs $\boldsymbol{R}_0$, (iv) the dice loss $\mathcal{L}_{dice}$ [34] for the predicted mask $\boldsymbol{M}$ of MPB, and (v) the total variation loss $\mathcal{L}_{tv}$ [19] for each blend layer $\boldsymbol{B_i}$ ($i \in \{0, 1, ..., l\}$). In summary, the joint loss is written as:

$$\begin{aligned}\mathcal{L}_{joint} = \lambda_1\mathcal{L}_{mse} + \lambda_2\mathcal{L}_{perc} + \lambda_3\mathcal{L}_{adv} \\ + \lambda_4\mathcal{L}_{dice} + \lambda_5\mathcal{L}_{tv},\end{aligned} \tag{6}$$

where $\lambda_1 = \lambda_4 = 1$ and $\lambda_2 = \lambda_3 = \lambda_5 = 0.1$ as default.

## 5. Experiments

### 5.1. Experimental Settings

**Datasets.** To verify the effectiveness and generalization of our model in LPR, we conduct experiments on two typical and popular local retouching scenarios: cloth retouching (CRHD-3K) and face retouching (FFHQR). The CRHD-3K dataset is described in Sec. 3. FFHQR [46] is a large-scale face retouching dataset based on FFHQ [20], which contains 70,000 high-definition face images from FFHQ and their corresponding retouched images. To enable comparison with the methods having diverse inference ability, we pad and resize all the images to $1024 \times 1024$ for training and evaluation in our experiments. Besides, we show the performance of the proposed network on CRHD-3K in the case of different resolutions (from 480p to 4K) in Sec. 5.5. CRHD-3K is randomly divided into a training set of 2,522 images and a test set of 500 images, and FFHQR is split into train/val/test set as in [46].

**Implementation details.** Our model and baselines are implemented using PyTorch 1.0 on Python 3.6 and trained on a single NVIDIA Tesla P100 GPU. We train our model using the Adam optimizer. With a batch size of 8, the learning rate is $5 \times 10^{-4}$ initially and reduced by $10\times$ after 100 epochs. We set $l$ at 2 as default in our experiments. Training the whole framework to convergence takes about 18 hours on CRHD-3K and about 70 hours on FFHQR.

### 5.2. Qualitative Comparison

Fig. 5 compares the images generated by the proposed model with those by the current state-of-the-art methods on the FFHQR [46] and CRHD-3K datasets. As we can see, pix2pixHD [52], ASAPNet [47], and LPTN [25] are limited in handling the LPR task, and fail to distinguish the retouching regions, resulting in global transfer. Moreover, visual artifacts are observed in the results of pix2pixHD [52] and ASAPNet [47]. VCNet [53] and AutoRetouch [46] yield competitive results; however, the details are still less elegant than ours. To sum up, the proposed model outperforms


Figure 6. Ablation study toward MPB and LAM on CRHD-3K. The masks presented in the upper right corner of the last four columns show the changing area relative to the input, following the same processing method illustrated in Sec. 4.2.

| Datasets | FFHQR [46] | | | | CRHD-3K | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Metrics | LPIPS$^\dagger$ | PSNR$^\P$ | SSIM$^\P$ | User Study$^\P$ | LPIPS$^\dagger$ | PSNR$^\P$ | SSIM$^\P$ | User Study$^\P$ | Time$^\dagger$ |
| VCNet [53] | 0.039 | 38.36 | 0.973 | 13.3% | 0.084 | 31.99 | 0.902 | 6.0% | 0.197 |
| AutoRetouch [46] | 0.025 | 41.83 | 0.986 | 18.0% | 0.081 | 32.70 | 0.907 | 7.3% | 0.057 |
| pix2pixHD [52] | 0.053 | 31.39 | 0.952 | 2.0% | 0.101 | 27.23 | 0.892 | 1.3% | 0.055 |
| ASAPNet [47] | 0.163 | 26.21 | 0.910 | 0.0% | 0.101 | 30.31 | 0.887 | 4.7% | 0.015 |
| LPTN [25] | 0.069 | 37.42 | 0.949 | 4.0% | 0.042 | 35.09 | 0.963 | 20.0% | 0.035 |
| Ours | **0.018** | **44.35** | **0.993** | **62.7%** | **0.029** | **37.35** | **0.971** | **60.7%** | **0.009** |

Table 1. Objective quantitative comparison ($^\dagger$Lower is better; $^\P$Higher is better).

the counterparts with reasonable retouched results of high detail fidelity.

### 5.3. Quantitative Comparison

**Objective evaluation.** We quantitatively evaluate the proposed method with three metrics: LPIPS, PSNR and SSIM. Table 1 shows the results achieved on the FFHQR [46] and CRHD-3K datasets, where the proposed method achieves the best results compared with the other approaches, clearly demonstrating its effectiveness.

**User study.** We evaluate the proposed method via a human subjective study. 10 volunteers with image processing expertise were invited to choose the most elegant image from those generated by the proposed method and the state-of-the-art approaches. Specifically, each participant has 15 questions from FFHQR [46] and 15 questions from CRHD-3K. We tally the votes and show the statistics in Table 1. Our method performs favorably against the other methods.

**Inference time.** We evaluate the inference time of all the models on images of $1024 \times 1024$ pixels with a single NVIDIA Tesla P100 GPU (16 GB). As shown in Table 1, VCNet [53], AutoRetouch [46] and pix2pixHD [52] are computationally expensive on high-resolution images. Thanks to the proposed adaptive blend pyramid architecture, our model outperforms the other methods regarding the time consumption.

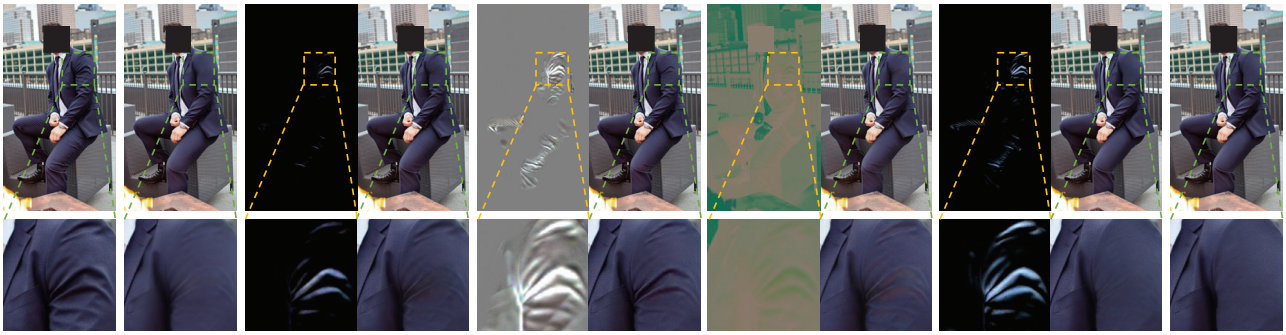
### 5.4. Ablation Study

In order to verify the rationality and effectiveness of the proposed components, we conduct extensive ablation experiments on the CRHD-3K dataset. Table 2 shows the quantitative results, including ablation comparison for MPB, LAM, the refining module (RM), and some major blend methods. As revealed in the table, MPB plays a key role in the architecture, contributing a $\sim$4% improvement. We replace LAM with PCB proposed in VCNet [53], and the results show that LAM achieves a $\sim$1% improvement. RM produces a $\sim$2.5% improvement. We also compare the results by adopting different blend modes for image translation, and ABM yields an improvement of 1$\sim$1.5% compared to other methods. Below we analyze the effectiveness of each module in detail based on the visualization results.

**On MPB.** MPB realizes the localization of the target region to guide local retouching. With the assistance of the mask predicted by MPB, LRB achieves a better semantic perception of the image under a limited model capacity. As shown in Fig. 6, without MPB (column b), the model produces a certain blur effect in the non-target region (the local region on the top), and it is susceptible to background distraction. The changing areas of the results show that MPB helps to keep the non-target region intact to a large extent. Moreover, thanks to the attention to the local target region, precise retouched results are obtained.

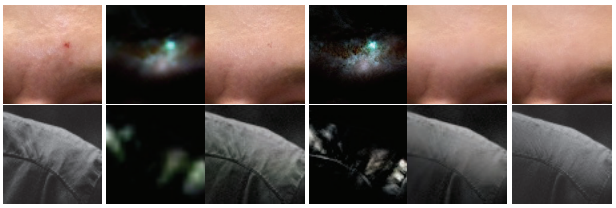
**On LAM.** We compare LAM with PCB [53], which exhibits its effectiveness in the image inpainting task. As shown in Fig. 6 (column c), the network with PCB fails to make full use of the textures of the target region and results in the loss of details that should be preserved. In contrast, our LAM renders local retouching with high detail fidelity.

**On ABM.** To validate the effectiveness of ABM for extending local retouched results from low resolution to high resolution, we compare it with various blend methods as well as other translation strategies. As shown in Fig. 7, directly upsampling and refining the RGB results loses plenty of



(a) Input (b) Refine RGB (c) Addition (d) Soft Light (e) Convs (f) Ours (g) Target

Figure 7. Visual comparison among different blend methods on CRHD-3K, including (b) refining RGB directly, (c) Addition [1], (d) Soft Light [2], (e) adaptive blend with convolutions and (f) ours. To facilitate visualization, we scale all the blend layer values to $0 \sim 255$.



(a) Input (b) w/o Refining Module (c) Ours (d) Target

Figure 8. Ablation study toward the Refining Module on FFHQR and CRHD-3K. For better observation, we only present some local regions of the blend layers and the corresponding RGB results.

details, resulting in blurred effects. We adopt some existing blend modes with fixed functions used in digital image editing, such as Addition [1] and Soft Light [2]. The Addition blend mode that adopts linear translation is unable to fit well when the color of the local region changes severely. Limited by the transformation ability, the soft light blend mode cannot greatly change the pixel value near 0 and 255 (as shown in the column d). We also design two 3-layer convolutional networks to replace Eq. (3) and Eq. (4) respectively for adaptive blend. However, subject to the irreversibility of the two translations, it is prone to produce a color difference. With powerful transformation capabilities and good reversibility, the proposed ABM module achieves much more smooth and realistic results.

**On RM.** The refining module is proposed to progressively compensate for the deficiency of details in the low-resolution blend layer. As shown in Fig. 8, RM gains massive details for the blend layer, so as to complete precise retouching of the local region.

### 5.5. High-resolution Expansion Capability

BPL has a powerful ability to expand upward. By increasing $l$ in Fig. 3, we can achieve local retouching on ultra high-resolution photos at a very low cost. Table 3 shows the quantitative results and runtime of our model at different resolutions. It can be seen that even for 4K resolution images, the model still achieves good retouched results at a super fast speed. Visual examples of 4K images are provided in the supplementary material.

| MPB | LAM | Blend methods | | | | | RM | PSNR |
|---|---|---|---|---|---|---|---|---|
| | | RGB | Addition | Soft Light | Convs | Ours | | |
| | ✓ | | | | | ✓ | ✓ | 33.02 |
| ✓ | | | | | | ✓ | ✓ | 36.24 |
| ✓ | ✓ | | | | | ✓ | | 34.78 |
| ✓ | ✓ | ✓ | | | | | ✓ | 35.76 |
| ✓ | ✓ | | ✓ | | | | ✓ | 36.57 |
| ✓ | ✓ | | | ✓ | | | ✓ | 36.10 |
| ✓ | ✓ | | | | ✓ | | ✓ | 35.88 |
| ✓ | ✓ | | | | | ✓ | ✓ | **37.35** |

Table 2. Quantitative ablation experiments on CRHD-3K.

| Resolution | LPIPS† | PSNR¶ | SSIM¶ | Runtime | Memory |
|---|---|---|---|---|---|
| 512×512 ($l = 1$) | 0.027 | 37.50 | 0.971 | 0.008 | 1043MB |
| 1024×1024 ($l = 2$) | 0.029 | 37.35 | 0.971 | 0.009 | 1329MB |
| 2048×2048 ($l = 3$) | 0.029 | 37.24 | 0.968 | 0.010 | 2505MB |
| 4096×4096 ($l = 4$) | 0.030 | 37.19 | 0.969 | 0.014 | 7191MB |

Table 3. Comparison of evaluation metrics, runtime, and memory consumption of our model in the case of different resolutions on CRHD-3K. The runtime denotes the average inference time of all test samples on a single NVIDIA Tesla P100 GPU (16 GB).

## 6. Conclusion

We summarize a kind of photo retouching as the local photo retouching (LPR) task and develop a novel solution to it, giving full consideration to the intrinsic characteristics of the task itself. Specifically, we design a context-aware local retouching layer based on a multi-task architecture to implement mask prediction and local retouching simultaneously. By utilizing the predicted mask as guidance, global context and local texture can be fully captured to render consistent retouching. Then, we build a pyramid based on the adaptive blend module and the refining module to expand the low-resolution results to the high-resolution ones progressively, showing great extensibility and high fidelity. Consequently, our method exhibits excellent performance in terms of the retouching quality as well as the running speed, achieving real-time inference on 4K images with a single NVIDIA Tesla P100 GPU. In addition, we introduce the first high-definition clothing retouching dataset CRHD-3K to promote the research on clothing retouching and LPR.

## References


[1] Blend modes. https://en.wikipedia.org/wiki/Blend_modes. 5, 8

[2] Pegtop blend modes: soft light. http://www.pegtop.net/delphi/articles/blendmodes/softlight.htm. 5, 8

[3] Kyungjune Baek, Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Hyunjung Shim. Rethinking the truly unsupervised image-to-image translation. In *ICCV*, 2021. 2

[4] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *Readings in Computer Vision*, 31(4):671–679, 1987. 3

[5] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *TIP*, 2018. 1, 2

[6] Nian Cai, Zhenghang Su, Zhineng Lin, Han Wang, Zhijing Yang, and Bingo Wing-Kuen Ling. Blind inpainting using the fully convolutional neural network. *The Visual Computer*, 2017. 2

[7] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. 2

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 2

[9] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. pages 8188–8197, 2020. 2

[10] Yubin Deng, Chen Change Loy, and Xiaoou Tang. Aesthetic-driven image enhancement by adversarial learning. In *ACM MM*, 2018. 2

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 2

[12] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *TOG*, 2017. 1, 2, 3

[13] Xiefan Guo, Hongyu Yang, and Di Huang. Image inpainting via conditional texture and structure dual generation. In *ICCV*, 2021. 2

[14] Jingwen He, Yihao Liu, Yu Qiao, and Chao Dong. Conditional sequential modulation for efficient global image retouching. In *ECCV*, 2020. 1, 2

[15] Xiaowei Hu, Yitong Jiang, Chi-Wing Fu, and Pheng-Ann Heng. Mask-shadowgan: Learning to remove shadows from unpaired data. In *ICCV*, 2019. 2, 3

[16] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2

[17] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 6

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2, 6

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 6

[21] Satoshi Kosugi and Toshihiko Yamasaki. Unpaired image enhancement featuring reinforcement-learning-controlled image editing software. In *AAAI*, 2020. 1, 2

[22] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *CVPR*, 2020. 2

[23] Xinyang Li, Shengchuan Zhang, Jie Hu, Liujuan Cao, Xiaopeng Hong, Xudong Mao, Feiyue Huang, Yongjian Wu, and Rongrong Ji. Image-to-image translation via hierarchical style disentanglement. In *CVPR*, 2021. 2

[24] Jie Liang, Hui Zeng, Miaomiao Cui, Xuansong Xie, and Lei Zhang. Ppr10k: A large-scale portrait photo retouching dataset with human-region mask and group-level consistency. In *CVPR*, 2021. 3

[25] Jie Liang, Hui Zeng, and Lei Zhang. High-resolution photo-realistic image translation in real-time: A laplacian pyramid translation network. In *CVPR*, 2021. 1, 2, 3, 6, 7

[26] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Image inpainting guided by coherence priors of semantics and textures. In *CVPR*, 2021. 2

[27] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In *CVPR*, 2021. 2

[28] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018. 2

[29] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *ECCV*, 2020. 2

[30] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, 2017. 2

[31] Yang Liu, Jinshan Pan, and Zhixun Su. Deep blind image inpainting. In *IScIDE*, 2019. 2

[32] Zhihao Liu, Hui Yin, Yang Mi, Mengyang Pu, and Song Wang. Shadow removal by a lightness-guided network with training on unpaired data. *TIP*, 2021. 2, 3

[33] Zhihao Liu, Hui Yin, Xinyi Wu, Zhenyao Wu, Yang Mi, and Song Wang. From shadow generation to shadow removal. In *CVPR*, 2021. 2, 3

[34] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 6

[35] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *ICCVW*, 2019. 2

[36] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *ECCV*, 2020. 2

[37] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 2

[38] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *NeurIPS*, 2020. 2

[39] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[40] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *CVPR*, 2018. 2, 3

[41] Ruijie Quan, Xin Yu, Yuanzhi Liang, and Yi Yang. Removing raindrops and rain streaks in one go. In *CVPR*, 2021. 2, 3

[42] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, 2019. 2, 3

[43] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*, 2021. 2

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 4

[45] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Bjorn Ommer. A style-aware content loss for real-time hd style transfer. In *ECCV*, 2018. 2

[46] Alireza Shafaei, James J Little, and Mark Schmidt. Autoretouch: Automatic professional face retouching. In *WACV*, 2021. 1, 2, 6, 7

[47] Tamar Rott Shaham, Michaël Gharbi, Richard Zhang, Eli Shechtman, and Tomer Michaeli. Spatially-adaptive pixelwise networks for fast image translation. In *CVPR*, 2021. 2, 3, 6, 7

[48] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, 2020. 2, 3

[49] Hong Wang, Zongsheng Yue, Qi Xie, Qian Zhao, Yefeng Zheng, and Deyu Meng. From rain generation to rain removal. In *CVPR*, 2021. 2, 3

[50] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *CVPR*, 2019. 2

[51] Tengfei Wang, Hao Ouyang, and Qifeng Chen. Image inpainting with external-internal learning and monochromic bottleneck. In *CVPR*, 2021. 2

[52] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 2, 3, 6, 7

[53] Yi Wang, Ying-Cong Chen, Xin Tao, and Jiaya Jia. Vcnet: A robust approach to blind image inpainting. In *ECCV*, 2020. 2, 6, 7

[54] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. 2

[55] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019. 2, 4

[56] Yingchen Yu, Fangneng Zhan, Shijian Lu, Jianxiong Pan, Feiying Ma, Xuansong Xie, and Chunyan Miao. Wavefill: A wavelet-based generation network for image inpainting. In *ICCV*, 2021. 2

[57] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *TPAMI*, 2020. 1, 2

[58] Yu Zeng, Zhe Lin, Huchuan Lu, and Vishal M Patel. Cr-fill: Generative image inpainting with auxiliary contextual reconstruction. In *ICCV*, 2021. 2

[59] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2