This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3174413, IEEE Transactions on Multimedia

Learning Adaptive Patch Generators for Mask-Robust Image Inpainting

Hongyi Sun, Wanhua Li, Yueqi Duan, Jie Zhou, *Senior Member, IEEE*, and Jiwen Lu, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a Mask-Robust Inpainting Network (MRIN) to recover the masked areas of an image. Most existing methods learn a single model for image inpainting, under a basic assumption that all the masks belong to the same type. However, we discover that the masks are usually complex and exhibit various shapes and sizes at different locations of an image, where a single model cannot fully capture the large domain gap across different masks. To address this, we learn to decompose a complex mask area into several basic mask types and inpaint the damaged image in a patch-wise manner with a type-specific generator. More specifically, our MRIN consists of a mask-robust agent and an adaptive patch generative network. The mask-robust agent contains a mask selector and a patch locator, which generates mask attention maps to select a patch at each step. We train our mask-robust agent to learn the optimal inpainting patch route in a reinforcement learning manner by formulating the process of inpainting sequentially as a Markov decision process. Then, based on the predicted mask attention maps, the adaptive patch generative network inpaints the selected patch with the generators bank, so that it sequentially inpaints each patch with different patch generators according to its mask type. Extensive experiments demonstrate that our approach outperforms most state-of-the-art approaches on the Place2, CelebA, and Paris Street View datasets.**

***Index Terms*—Image inpainting, mask-robust agent, adaptive patch generators.**

I. Introduction

Image inpainting [1], [2], also known as image completion, aims to recover the masked areas for a damaged or missing image with plausible pixel values based on the information of the known areas. This task plays a significant role in the field of image processing and has been widely used in various applications [3]–[5] such as photo editing, object removal, and old photo restoration [6]–[8]. Thus, it has drawn continuous attention for decades [9]–[13].

Early image inpainting methods [14]–[19] aim to fill masked areas with the same shape, such as squared or oval holes. However, recovering damaged images with irregular masks is more common in real applications. To inpaint damaged images with irregular masks, many methods [13], [20] have been proposed over the past few years. Liu *et al.* [13] first proposed the efficacy of training inpainting models on irregularly shaped masks dataset. They also proposed an irregular mask dataset and partial convolutions to solve this problem, where the convolution is masked and renormalized to be conditioned on only valid pixels. To solve the issue of vanilla convolution that treats all input pixels as valid ones, Yu *et al.* [20] designed gated convolution that generalizes partial convolution by providing a learnable dynamic feature selection mechanism for each channel at each spatial location across all layers. Li *et al.* [21] presented a Recurrent Feature Reasoning (RFR) network which recurrently infers the hole boundaries and uses them as clues for further inference.

While extensive efforts have been devoted to image inpainting with irregular masks [13], [20], [22], these methods use a single network to inpaint images with complex masks types and ignore the variations in diverse masks. For example, a model trained with the bounding box masks may tend to use the remote pixels to recover the masked area. Similarly, one trained with stroke-like masks may exploit more details in nearby pixels of the masked area. Thus, a model trained with mask type A usually performs worse in a damaged image with mask type B. On the other hand, the mask of a damaged image is usually composed of various types. There may exist different mask shapes or ratios at different locations. It is infeasible for the progressive inpainting methods [14], [17], [23], [24] to ignore these domain gaps across different masks and train only one network.

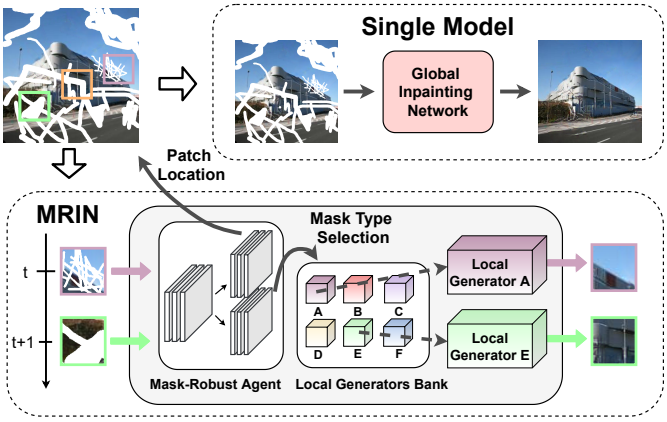

Fig. 1. Most existing inpainting methods regard the whole mask as the same type and train a single inpainting model. Our MRIN method inpaints the image in a patch-wise manner. At each step, the mask-robust agent selects a patch and decides its mask type. Then, the local generative networks explore the global perspective of the whole image and inpaints selected patch based on its mask shape and ratio. Different colors of the rectangle patch in the image denote different mask types.

Hongyi Sun, Wanhua Li, Jie Zhou, Jiwen Lu are with the Beijing National Research Center for Information Science and Technology (BNRist), and the Department of Automation, Tsinghua University, Beijing, 100084 China. E-mail: sunhongy18@mails.tsinghua.edu.cn; li-wh17@mails.tsinghua.edu.cn; jzhou@tsinghua.edu.cn; lujiwen@tsinghua.edu.cn.

Yueqi Duan is with the Beijing National Research Center for Information Science and Technology (BNRist), and the Department of Electronic Engineering, Tsinghua University, Beijing, 100084 China. E-mail: duanyueqi@tsinghua.edu.cn.

To address this challenge, we propose a Mask-Robust Inpainting Network (MRIN) to decompose an irregular mask into several basic mask types and inpaint the whole image in a patch-wise way, as shown in Fig. 1. Our MRIN consists of a mask selector, a patch locator, and an adaptive patch generative network. First, we simultaneously train a mask selector to generate mask attention maps and a patch locator to select a patch suitable for inpainting at a step. Then, we feed the selected patch into an adaptive patch generative network to generate the local inpainted result with the corresponding mask type. Finally, our MRIN updates the patch in the masked image by sequentially performing the above operations until recovering the whole masked image.

More specifically, we train the patch locator in a deep reinforcement learning way for its effectiveness in globally optimizing the sequential models without supervision for every step. It views the whole masked image and considers the previous inpainted result to decide which patch to inpaint at the next step. We formulate the mask type decomposition and the patch selection as a Markov decision process and define the global reward based on the overall performance of the inpainted result. In this way, the agent learns a policy to determine an optimal location of the next inpainted patch conditioning on the current inpainted result and the history actions, which are memorized to avoid the inference trapped in a repetitive action cycle and the history information is captured by utilizing a Long Short-Term Memory (LSTM) layer [25]. Meanwhile, the mask selector network composes the whole mask into several basic mask types, which are represented by n mask type attention heatmaps. Then, we feed the generated heatmaps to the adaptive patch generative network so that it can inpaint the selected patch conditioning on the corresponding mask type. The adaptive patch generative network employs N parallel patch generators, each corresponding to a mask type. The individual inpainted results are combined softly based on the mask attention maps to obtain the local inpainted result. We evaluate our approaches for image inpainting on the Place2 [26], CelebA [27], and Paris StreetView datasets [28], where the experimental results demonstrate that our proposed method outperforms most of the state-of-the-art approaches.

Our main contributions of this work are summarized as follows:

- We propose a Mask-Robust Inpainting Network (MRIN) framework for image inpainting. In contrast to existing works for image inpainting which ignore domain gaps across different masks and train a single model, we design adaptive patch generators to better handle different mask types. MRIN inpaints the damaged image in a patch-wise manner. We develop a patch locator to select a patch and the mask selector to generate its mask attention map. Then, we devise adaptive patch generators to sequentially inpaint an masked image. To the best of our knowledge, we are the first to demonstrate the efficacy of adaptively selecting different patch generators to inpaint an image with different mask types.
- We conduct experiments on three image inpainting datasets, Place2 [26], CelebA [27], and Paris StreetView datasets [28]. Extensive experimental results demonstrate that our proposed MRIN outperforms the state-of-the-art approaches.

II. Related Work

In this section, we briefly review two related topics: 1) image inpainting, and 2) deep reinforcement learning.

A. Image Inpainting

Image inpainting methods can be divided into two categories: traditional methods which inpaints the masked area by searching similar patches or propagating colors, and deep methods employing deep neural networks to learn semantics and texture from training datasets.

Traditional inpainting methods attempt to find patches from unmasked regions to inpaint the masked region [1], [8], [13], [29]–[36]. For example, Bertalmio *et al.* [1] employed anisotropic diffusion to propagate pixel colors and Ballester *et al.* [34] proposed to solve partial differential equations (PDEs). However, these methods mainly work on simple cases or thin hole regions where it is plausible to find a proper patch from the background area. Specifically, they tend to inpaint over-blurring results when the hole regions grow larger or more complex. Patch-based image inpainting methods [8], [37] solve the problem by finding similar unmasked patches and copying corresponding textures. Such methods tend to inpaint high-quality texture results. However, they cannot obtain a semantic understanding of complex objects and thus perform unsatisfactorily on complex scenes.

Deep convolutional networks [36] have been used for image inpainting for their learning capability of hierarchical features. Target mask types of recent methods can be roughly categorized into two types: 1) holes with the same shape, i.e., squared or oval holes, and 2) irregular holes or masks with various sizes. Generative adversarial networks (GANs) [38] are widely used for regular masks. Pathak*et al.* [15] employed Context-Encoder to learn scene representation along with inpainting, which demonstrated the potential of CNNs for inpainting tasks. Iizuka *et al.* [39] proposed 'Globally and Locally Consistent Image Completion'(GLCIC) that employs an extra discriminator to ensure local image coherency and render more detailed and sharper results. Yu *et al.* [19] incorporated contextual attention operations with an inpainting network to guide the model to leverage feature information from distant unmasked areas of the image. Yan *et al.* [40] proposed a U-Net architecture to learn distant information that employed a shift-connection from encoder to decoder. Lahiri *et al.* [41] proposed a prior guided GAN for semantic inpainting which converted the 'iterative-inference' pipeline to a single feed-forward framework. These methods target for filling holes with the same shape, i.e., squared masks, and cannot handle irregular holes well. To deal with inpainting on irregularly masked images, Liu *et al.* [13] and Yu *et al.* [20] proposed partial convolution and gated convolution, respectively. Li *et al.* [21] proposed a Recurrent Feature Reasoning (RFR) network which recurrently infers the hole boundaries and uses them as clues for further inference. Wang *et al.* [42] defined a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3174413, IEEE Transactions on Multimedia

new blind inpainting setting and proposed a two-stage visual consistency network (VCN) to estimate where to fill.

In [43], Wang *et al.* proposed an external-internal inpainting scheme with a monochromic bottleneck to help image inpainting models remove the artifacts. In [44], Zhou *et al.* proposed a multi-homography transformed fusion method (TransFill) to inpaint the mask area by referring to another source image that shares similar scene contents with the target. In [45], Liao *et al.* proposed a semantic-wise attention propagation module(SWAP) to generate semantically realistic textures by capturing distant relationships and referring to the texture feature of the same semantic in the feature maps.

Though much effort has been devoted to image inpainting research, few works pay attention to the model training with different mask types. These methods regard all masks as one type and are regardless of the differences between diverse masks as illustrated in Section 1. Model training in this way cannot exploit the information of mask type and handle the large domain gap across different masks.

B. Deep Reinforcement Learning

Deep reinforcement learning has been introduced to many computer vision tasks [46]–[50] recently. It teaches the agent a policy to select actions that can obtain the greatest feedback sequentially. One of the popular approaches of reinforcement learning is Policy Gradient [51], which directly learns a policy function by maximizing a cumulated reward. Silver *et al.* [52] proposed a deterministic algorithm that can be more efficient over the high dimension action space. Lillicrap [53] applied deep neural networks to deterministic policy gradients so that they can operate on the continuous action space. These works generally leverage deep neural networks to substitute the policy function for it has a more inspired performance [52], [53]. Some methods employ Q-learning to learn an optimal policy to locate the target. For example, Goodrich et al. [54] defined 32 actions to shift the focal point so that the agent can get a higher reward when finding the goal. Caicedo et al. [55] defined an action set that contains several transformations of the bounding box and the agent can get a higher reward when the bounding box is closer to the ground truth at each step. In [25], Cao et al. proposed an Attention-aware Face Hallucination framework for sequentially locating attended patches and exploiting the global interdependency of the image.

Inspired by the recent successes of reinforcement learning and recurrent models on computer vision tasks [25], [49], [55]–[61], we propose the Mask-Robust Inpainting Network that employs a mask-robust agent to select an image patch and decide its mask type and then inpaints it with adaptive patch generators sequentially. In this way, the information of the mask and the global interdependency of the image is fully exploited.

III. Proposed Apporoach

In this section, we first introduce an overview of the proposed MRIN. Then, we propose the settings of the Markov decision process (MDP) to show how to utilize deep reinforcement learning. Finally, we present the network architecture and loss function of each component and show how to optimize them.

A. Overview of Mask-Robust Inpainting Network

A masked/damaged image I_d can be regarded as generated from an original image I_{gt} and a binary mask M with zeros on masked pixels else ones, according to $I_d = I_{gt} \odot M$, where $\odot$ is the Hadamard operator. Then, our MRIN proposes to generate an inpainted image I_T with the input of M and I_d by learning the following projection function F:

$$I_T = F(I_d, M|\omega), \tag{1}$$

where ω denotes the parameters of the MRIN, which aims to make I_T and I_{gt} as similar as possible and I_T looks natural and pleasurable. Our MRIN sequentially locates and inpaints the attended image patch in each step, which is formulated as a deep reinforcement learning procedure. Specifically, the framework consists of three modules: the mask selector S that dynamically decompose the mask into several basic mask types, a patch locator L that determines an image patch to be inpainted at the current step, and the adaptive patch generative network G which aims to inpaint the selected patch according to the mask type decomposition and update the whole image with the local inpainted result.

Specifically, the whole image inpainting procedure of MRIN can be formulated as follows. Given an input masked image I_t and its corresponding mask M_t at the t-th step, the mask-robust agent employs a patch locator L to select one image patch I_{p_t} and a mask selector S to decompose the mask into several basic mask types H_{mt},

$$H_{mt} = S(M_t; \theta), \tag{2}$$

$$H_{lt} = L(I_t, M_t; \phi), \tag{3}$$

$$I_{p_t} = g(I_t, H_{l_t}), \tag{4}$$

where θ and ϕ denote the parameters of mask selector S and patch locator L respectively. $g(I_t, H_{l_t})$ denotes a randomly cropping operation that crops a fixed-size patch from I_t and the probability of the cropping center locating at (x, y) is $H_{lt_{x,y}}$. The patch size is set as 64×64. H_{mt} and H_{lt} are the mask decomposition result and location probability map detailed in Section 3.3.

Then, the adaptive patch generative network G inpaints the selected patch according to the mask attention map H_{mt}. It employs N parallel local generators each one regarding the selected patch as a certain mask type, *ie* regarding as i-th type, an inpainted result can be generated by $p_i = G(I_p, i; \psi)$ where ψ denotes the parameters of the adaptive generative networks. The local inpainted result $I_{p_{t+1}}$ can be computed as:

$$I_{p_{t+1}} = \sum_{i=1}^{N} (p_i \odot H_m(i)). \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3174413, IEEE Transactions on Multimedia

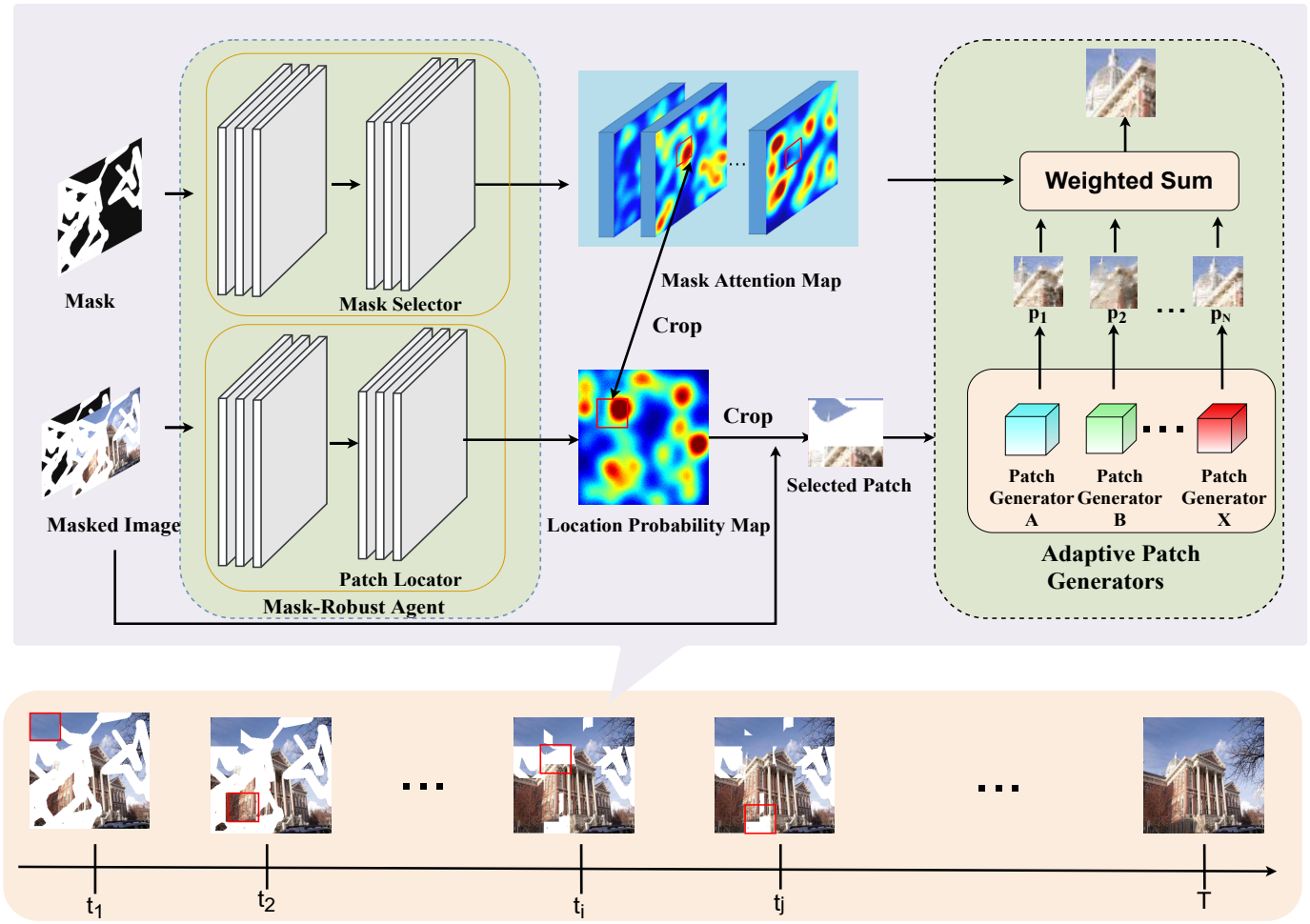

Fig. 2. Illustration of the Mask-Robust Inpainting Network. It is mainly composed of a mask-robust agent and an adaptive patch generative network. The mask-robust agent employs a mask selector to decompose a mask into several mask types and a patch locator to select the patch to be inpainted at the current step. The local inpainting model employs several adaptive patch generators, each one inpainting the patch regarding its mask as a certain type. Then, the local inpainted results are added weighted by the mask type heatmap as the local inpainted result to update the selected patch at the current step.

Finally, the current image updates the patch I_{p_t} with the local inpainted result $I_{p_{t+1}}$. Thus, the image inpainting process at a time step can be written as:

$$I_{t+1}(x,y) = \begin{cases} I_{p_{t+1}}(x',y'), & (x,y) \quad in \quad P \\ I_t(x,y) \quad , & otherwise \end{cases} \tag{6}$$

where P denotes the selected patch and (x', y') are the corresponding coordinates in the local patch. Then, the updated I_{t+1} replaces I_t as the input of the next recurrence and repeat this process for T times, where T is the maximum iteration numbers of the patch selecting procedure. Therefore, our whole sequential inpainting process of MRIN can be written as follows:

$$\begin{cases} I_0 &= I_d \\ I_t &= F(I_{t-1};\theta), \qquad 0 \le t \le T \\ I_{pred} &= I_T \end{cases} \tag{7}$$

where $\theta = \{\theta, \phi, \psi\}$ denoting the parameter of MRIN, and $F = S, L, G$ denoting the whole MRIN model. The overview of our MRIN is illustrated in Fig. 2.

B. Markov Decision Process Formulation

We train the patch locator in a reinforcement learning way because the patch locating and cropping process involves $argmax$ or $argsoftmax$ operators, which are non-differentiable and the conventional back-propagation algorithm is hard to be directly used for training. Moreover, reinforcement learning shows great effectiveness in globally optimizing the sequential models without supervision for every step.

Our MRIN performs the image inpainting in a patch-wise way, which can be treated as a decision-making process at discrete time intervals. The mask-robust agent takes action to determine an optimal image patch to be inpainted and predict its mask decomposition based on the current image at each step. Given the selected patch, the adaptive patch generative network inpaints the extracted local patch based on its mask type. During each time step, the state is updated by recovering the selected patch with the local inpainted result. The whole MRIN recurrently selects and inpaints local patches until the maximum time step T is achieved. At the end of this sequence, a delayed global reward is utilized to guide the training of the mask-robust agent, which measures the similarity between the

predicted image and the ground truth. Thus, the mask-robust agent can explore an optimal inpainting route for the masked image by maximizing the global holistic reward. Specifically, we formulate the process of selecting and inpainting an image patch as an MDP, where a state space, an action space, and a reward function of the agent are defined as follows.

State: The state s_t is defined as a tuple of (I_t, h_t), composed of two parts: 1) the image I_t updated by the local inpainting network from the previous step, which provides information for the agent to decide the next patch to be inpainted and choose the mask type. 2) The latent variable h_t incorporating previous actions, which is obtained by forwarding the last encoded history action vector h_{t_1} into the ConvLSTM layers. In this way, the agent is provided enough history information without looking back on previous steps. I_0 is initialized as the masked image I_d, and $h_0 \in \mathbb{R}^{128\times h\times w}$ is randomly initialized with Gaussian distribution.

Action: Given state I_t, h_t at each step t, the mask-robust agent encodes the current face image with convolution layers and fuses it with the history vector h_t utilizing an LSTM unit. Then, it employs the patch locator L and the mask selector S to select the image patch to be inpainted and decompose the mask into several basic mask types respectively. Thus, the action space composes of two parts: selecting the patch a_p and generating the mask attention map a_m.

The first heatmap $H_l \in \mathbb{R}^{h\times w}$ decides the next image patch to be inpainted, where $H_{lx,y}$ denotes the probability of the pixel (x, y) is chosen as the patch center. The second heatmap $H_m \in \mathbb{R}^{n\times h\times w}$ decides the mask type decomposition result, where n is the number of mask types and the probability $P(i) = H_{mi,x,y}$ denotes the probability distribution of the pixel (x, y) belongs to the i-th mask type.

Reward: The reward guides the mask-robust agent to exploit the optimal sequence. The whole model aims at making the final inpainted result I_t look similar to the ground truth unmasked image I_{gt}. Thus, we define the reward conditioning on an image similarity evaluation function L_{SIM} (detailed in Section 3.4), which evaluates the similarity between two images. Supposing the local inpainting network is fixed and T local patches are selected, we can get the final inpainted result I_T by sequentially inpainting the list of the local patches. Then, the reward is defined as

$$r_t = \begin{cases} 0 & t < T \\ -L_{SIM}(I_{gt}, I_T) & t = T. \end{cases} \tag{8}$$

Let γ denote the discounted factor, then the total discounted reward can be calculated simply as

$$R = -\gamma^T \cdot L_{SIM}(I_{gt}, I_T). \tag{9}$$

State Transition: Once the agent selects an image patch and decides its mask type, the adaptive local inpainting network will take it as input and generate an inpainted result patch based on the mask type. According to the local inpainted result, the whole masked image will update this patch. Suppose f_i denotes the local inpainting network. The next state will be $\{I_{t+1}, h_{t+1}\}$, where I_{t+1} is computed with (6), which can be regarded as the state transition.

C. Mask-Robust Agent

As we have discussed in section 1, there exist several mask types in a damaged image. To recover such an image, we propose to use different patch generators to inpaint patches with different mask types. Motivated by this, a natural idea is to inpaint the image in a patch-wise way. Specifically, we discover that the whole mask consists of several basic mask types so we can divide the whole image into several patches. In a certain patch, we suppose its mask roughly belongs to one type so that it can be inpainted by the corresponding patch generator. Therefore, we train a mask-robust agent to achieve the patch-wise inpainting process, which selects a patch and decides its mask type at a step.

The mask-robust agent consists of the mask selector S and the patch locator L. The patch locator takes a masked image and its corresponding mask as input and targets to select a patch to be inpainted. To exploit the information of the mask, the mask selector only takes the mask as input. They both start with three ConvLSTM layers to extract features from the whole image or the mask and two convolution layers are followed to generate expected heatmaps. Thus, the network splits into two branches and each aims at a task. The output of the first branch, mask selector module, is mask attention heatmaps $H_m \in \mathbb{R}^{n\times h\times w}$ corresponding to the action a_m, where $H_{mk,i,j}$ denotes the possibility of the pixel located at (i, j) classified as the k-th mask type. The output of the second branch, patch locator, is a heatmap $H_l \in \mathbb{R}^{h\times w}$ corresponding to the action a_p, where $H_{lx,y}$ denotes the probability of the pixel (x,y) is chosen as the patch center.

We define a mask decomposition loss function L_{mask} to guide the mask selector S. The loss L_{mask} consists of three parts: locally consistent loss L_c, mask type-specific loss L_e, and l_2 loss L_d, where

$$L_c = \sum_{k=1}^{N}\sum_{i,j}((H_{k,i+1,j} - H_{k,i,j})^2 + (H_{k,i,j+1} - H_{k,i,j})^2). \tag{10}$$

For simplicity, we omit the subscript m here. The loss L_c employs total variation (TV) loss to encourage spatial smoothness in the generated mask maps. This is because we believe the points in a region are supposed to belong to the same mask type.

$$L_e = \sum_{x=1}^{h}\sum_{y=1}^{w} Entropy(P(i)), \tag{11}$$

where $P(i) = H_{mi,x,y}$ denotes the possibility distribution of mask type at the location (x,y), and $Entropy(P(x)) = \sum_{i=1}^{n}(P(i) \cdot log(P(i)))$ computes the entropy of the distribution. Thus, the loss L_e will be large when the possibility distribution $P(i) = H_{mi,x,y}$ is close to the uniform distribution. In this way, the loss L_e encourages that the predicted possibility at a location belongs to a certain mask type.

To guarantee that the predicted heatmaps are based on mask types rather than some other issues accidentally, we provide the labels for the mask selector as the supervisory signal. We compute the l_2 distance between the predicted heatmaps and the ground truth of the mask types. The mask is randomly

generated with N types. Some examples are shown in Figure 8. During the training phase, the numbers of different mask types are equal. Thus, the l_2 loss L_d is computed as

$$L_d = \sum_{k=1}^{N} \|H_k - H_k^{gt}\|_2 \tag{12}$$

In summary, the loss function L_{mask} is computed as

$$L_{mask} = \lambda_c L_c + \lambda_e L_e + \lambda_d L_d. \tag{13}$$

Besides the loss function L_{mask} above, the mask decomposition network is also guided by the loss L_{SIM} at the same time as the local inpainting network optimization.

The patch searcher branch is guided by the reward (9) in a reinforcement learning way. At each step, the agent learns a policy to determine an optimal location of the next inpainted patch conditioning on the current inpainted result and the history actions. The history actions are memorized to avoid the inference trapped in a repetitive action cycle and the history information is captured by utilizing Long Short-Term Memory (LSTM) layers [25].

D. Adaptive Patch Generative Network

As the mask-robust agent has selected a patch and predicts its mask type at a step, recovering the patch becomes a traditional image inpainting task. Specifically, the adaptive patch generative network G is employed to inpaint the selected image patch based on the mask type decided by the mask selector. Its input consists of two parts: 1) The whole image I_t and its corresponding mask M_t, 2) The selected image patch I_p and its corresponding mask M_p. For a specific patch generator, we use a similar network architecture as the existing generator models. The whole image and its mask are concatenated and passed into three convolution layers to generate a feature map that shares the same size as the selected image patch. Then, the feature map is concatenated with the image patch and fed into convolution layers to extract the local and global features. Finally, decoder convolutions are employed to obtain the inpainted result of the local patch. The adaptive patch generative network has N decoder convolutions that share the same architecture, each one inpainting for a specific mask type, and N denotes the number of basic mask types. The network will output N inpainted results $\boldsymbol{p} = \{p_1, p_2, \cdots, p_N\}$ where p_i corresponds to i-th mask type respectively. The final local inpainted result is the weighted sum of the N results according to (5).

The patch generative network G is guided by the similarity loss L_{SIM} that measures the similarity between the inpainted patch and the corresponding ground truth patch cropped from the unmasked image. The L_{SIM} consists of three parts: perceptual loss $L_{perceptual}$, SSIM loss L_{SSIM}, weighted mean L1 loss $L_{weightedl1}$. Given the local inpainted result I_{pred} and its corresponding ground truth I_{gt} and mask M, the perceptual loss is computed as:

$$L_{perceptual} = \sum_{i=1}^{N} \frac{1}{H_i W_i C_i} |\phi_{pool_i}^{gt} - \phi_{pool_i}^{pred}|_1, \tag{14}$$

where ϕ_{pool_i} denotes the feature maps extracted from the i-th pooling layer in VGG-16 and H_i,W_i,C_i denote its height, weight and channel size respectively. The perceptual loss measures the differences between the deep feature map of the predicted image and the ground truth, which can effectively guide the model to explore structural and textural information of the image.

The SSIM loss is computed by:

$$L_{SSIM} = 1 - \frac{(2\mu_{gt}\mu_{pred} + C_1)(2\sigma_{gt,pred} + C_2)}{(\mu_{gt}^2 + \mu_{pred}^2 + C_1)((\sigma_{gt}^2 + \sigma_{pred}^2 + C_2))}, \tag{15}$$

where μ_{gt} and μ_{pred} denotes the mean value of the ground-truth image and predicted image respectively. Similarly, σ_{gt} and σ_{pred} denoting the unbiased estimate of their standard deviation,$\sigma_{gt,pred} = \frac{1}{N-1}\sum_{i=1}^{N}(gt_i - \mu_{gt})(pred_i - \mu_{pred})$ denoting their correlation coefficient, C_1,C_2, and C_3 are constant parameters to avoid the instability caused by extreme value. The SSIM loss measures the similarity of two images from the three aspects of brightness, contrast, and structure, which is more consistent with human visual perception.

The weighted mean l_1 loss is defined as

$$\begin{aligned} L_{weightedl1} = {} & \lambda_{mask}|(I_{pred} - I_{gt}) \odot M|_1 \\ & + \lambda_{unmask}|(I_{pred} - I_{gt}) \odot (1 - M)|_1. \end{aligned} \tag{16}$$

The weighted mean l_1 loss calculates the l_1 difference between the predicted image and ground truth. The trade-off λ of the masked area is usually higher than the unmasked area because the main object of the model is to inpaint the masked area. In summary, the total loss L_{SIM} is calculated as

$$\begin{aligned} L_{SIM} = {} & \lambda_{perceptual} L_{perceptual} + \lambda_{SSIM} L_{SSIM} \\ & + L_{weightedl1}. \end{aligned} \tag{17}$$

Amongst these, the perceptual loss compares the differences between the deep feature maps of two images, and SSIM loss measures the structural similarity. Thus, loss L_{SIM} can effectively guide the model to learn the structural and textural information of images.

E. Optimization

The parameters $\{\theta, \phi\}$ of the mask-robust agent and ψ of the adaptive generative networks are jointly trained by a reinforcement learning scheme. The patch locator is optimized with the reinforcement learning algorithm [51], which is guided by the reward given at the end of sequential inpainting. The local inpainting network and mask decomposition network are optimized with L_{SIM} loss between the inpainted patch and the corresponding ground truth patch cropped from the unmasked image. The L_{SIM} loss is calculated at each step and minimized based on back-propagation. Besides the L_{SIM} loss, the mask decomposition network is also guided by the mask decomposition loss L_{mask}. The mask-robust agent and patch generative networks are trained jointly. Thus, the change of parameters in the patch generators will affect the final inpainted result, which causes a non-stationary objective for the mask-robust agent. Similar to [56], we employ a variance reduction strategy to reduce variance caused by the moving rewards at the training phase. The whole optimization process

Algorithm 1: MRIN for Image Inpainting

Input: Training unmasked image data $\boldsymbol{I} = \{I\}$, mask data $\boldsymbol{M} = \{M\}$ and corresponding mask type labels $\boldsymbol{H}_m^{gt}$, maximal training iterative number S, maximal iterative number of training a batch T

Output: The parameters of Patch Locator ϕ, Mask Selector θ, Adaptive Generative Networks ψ

Randomly initialize ϕ, θ, ψ.

for i = *1,2,*$\cdots$,S **do**

  Randomly select a batch of images $I_{n=1:N}$ from $\boldsymbol{I}$ and randomly select a batch of masks $M_{n=1:N}$ from $\boldsymbol{M}$, calculate the masked image with $I_d = I \odot M$.

  for t = *1,2,*$\cdots$,T **do**

    Generate mask attention map H_{mt} with (2).

    Generate location probility map H_{lt} with (3).

    Crop the local patch according to H_{lt} with (4).

    Inpaint selected the local patch with (5).

    Compute loss L_{mask} with (13).

    Compute loss L_{SIM} with (17).

    Update $\psi \leftarrow \frac{\partial}{\partial \psi}(L_{SIM})$.

    Update $\theta \leftarrow \frac{\partial}{\partial \theta}(L_{SIM} + L_{mask})$.

    Update I with (10).

  end

  Calculate the reward R with (9).

  Update ϕ with R according to Reinforcement Learning algorithm [51].

end

Return: θ, ψ and ϕ.

is detailed in Algorithm 1. More training details are described in Section 4.

IV. EXPERIMENTS

In this section, we conducted experiments on the three datasets and compared the qualitative and quantitative results with state-of-the-art methods. Further, we conducted ablation studies to verify the effects of the components in our model.

A. Datasets and Evaluation Metrics

Datasets: We conduct experiments on three public image datasets: Place2 [26], CelebA [27], and Paris StreetView [28]. *Place2 Challenge Dataset* [26] released by MIT contains over 8,000,000 images from over 365 scenes. We utilized images as training samples and 255,464 images for testing. The model trained on this dataset can obtain a better perception of natural scenes and performs well in building inpainting tasks.

CelebA Dataset [27] is a human face image dataset containing over 200,000 images. We trained a face inpainting model on 182,600 images and evaluated it on 20,000 test images. All the CelebA images are aligned at the post-processing phase to enable the model to learn the distribution from various face images.

Paris StreetView Dataset [28] is collected from street views of Paris, containing 14,900 training images and 100 test images. Similar to the Place2 dataset, the model trained on this dataset mainly aims at inpainting tasks of buildings in the city.

TABLE I
QUANTITATIVE COMPARISON ON PLACES2. *HIGHER IS BETTER. †LOWER IS BETTER.

Dataset		Places2		
Mask Ratio		10%-20%	30%-40%	50%-60%
SSIM*	PIC	0.932	0.786	0.494
	PConv	0.934	0.803	0.555
	EdgeConnect	0.933	0.802	0.553
	PRVS	0.936	0.810	0.574
	RFR-Net	0.939	0.819	0.596
	MRIN (Ours)	**0.941**	**0.827**	**0.602**
PSNR*	PIC	27.14	21.72	17.17
	PConv	27.29	22.12	18.29
	EdgeConnect	27.17	22.18	18.35
	PRVS	27.41	22.36	18.67
	RFR-Net	**27.75**	**22.63**	18.92
	MRIN (Ours)	27.67	22.41	**19.10**
Mean $l_1^\dagger$	PIC	0.0161	0.0441	0.0944
	PConv	0.0154	0.0409	0.0824
	EdgeConnect	0.0157	0.0408	0.0821
	PRVS	0.0148	0.0390	0.0778
	RFR-Net	0.0142	0.0381	0.0761
	MRIN (Ours)	**0.0128**	**0.0339**	**0.0576**

Standard Evaluation Protocols: We evaluate our experiments result with quantitative comparisons and qualitative comparisons.

Quantitative Comparisons. We quantitatively evaluate our experiments result with three evaluation metrics: Structural similarity index (SSIM), peak signal-to-noise ratio (PSNR), and mean l_1 loss. Higher SSIM and PSNR or lower mean l_1 loss mean the result is better.

Qualitative Comparisons. Inpainted result comparison on three datasets. between our method and several state-of-the-art approaches are shown in Fig. 3, 4, and 5. The compared methods include: PIC [62], PConv [13], GIP [19], PRVS [24], and RFR-Net [21].

Implementation Details: We set the maximum iteration steps $T = 35$ and the local patch size $H \times W = 64 \times 64$ for all datasets. All images in our experiments were resized to 256×256. The model was trained with the Adam [63] optimizer. The learning rate was set as to 1×10^{-4} and is divided by 2 every 20 epoch. The hyper-parameters in mask loss $\{\lambda_c, \lambda_e\}$ were set to $\{1, 0.01\}$, and the hyper-parameters in similarity loss $\{\lambda_{perceptual}, \lambda_{SSIM}, \lambda_{mask}, \lambda_{unmask}\}$ were set to $\{1, 0.01, 5, 1\}$ respectively. The parameter $\{C_1, C_2, C_3\}$ were set to {6.5,58.5,29.3} respectively. The discounted factor γ was set as 1 in the total discounted reward, and the number of mask type N was set as 5. More discussions of the parameter settings are detailed in ablation study. We train our model with Pytorch on 4 1080Ti GPUs.

B. Quantitative and Qualitative Comparisons

As shown in Table I,II,and III, our method achieves the lowest mean l_1 loss and highest SSIM on the Place2, CelebA, and Paris StreetView datasets with various mask ratio settings (10%$\sim$ 20%, 30%$\sim$40%, and 50%$\sim$60%) and produces comparable PSNR results comparing to the state-of-art methods. For the mean l_1 on the challenging dataset Paris2, our method outperforms the state-of-the-art method by a large margin,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3174413, IEEE Transactions on Multimedia

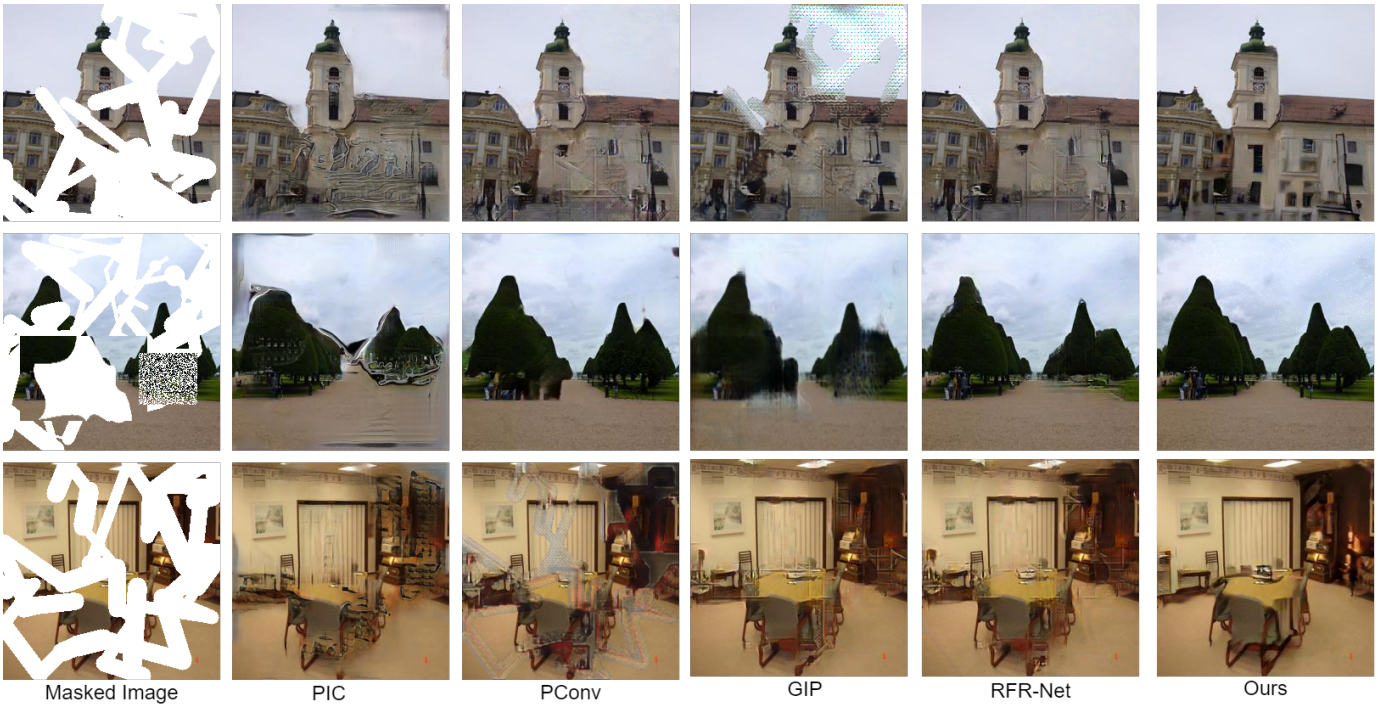

Fig. 3. Qualitative comparisons on Places2. From left to right are: masked image, inpainted results by PIC, PConv, GIP, RFR-Net, MRIN (ours), accordingly.

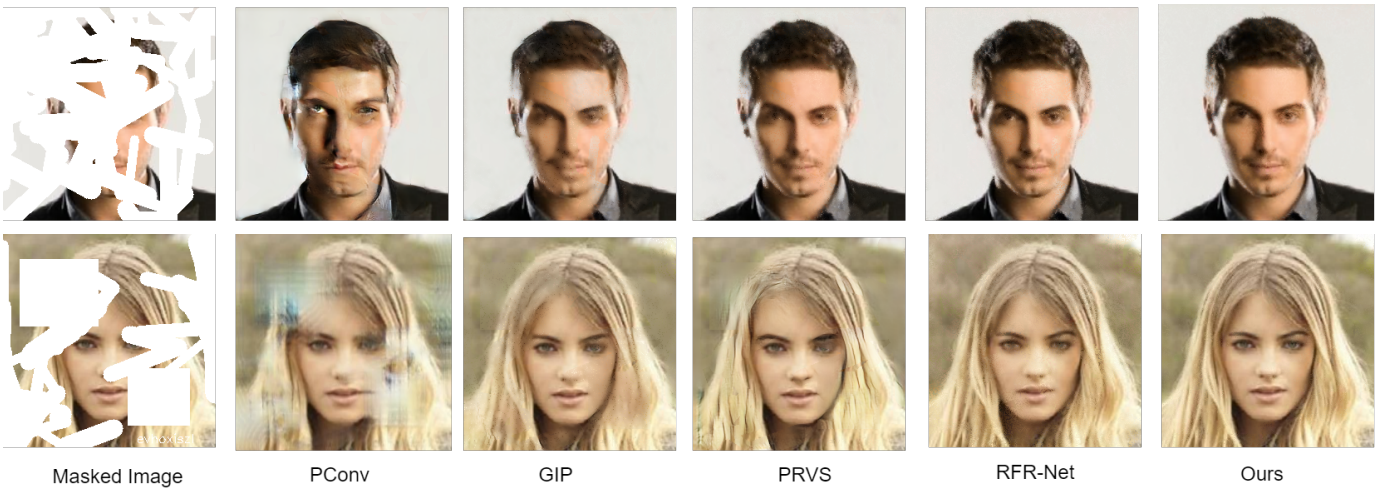

Fig. 4. Qualitative comparisons on CelebA. From left to right are: masked image, inpainted results by PIC, PConv, GIP, RFR-Net, MRIN (ours), accordingly.

5.76% vs 7.61%, with the mask ratio of 50%-60%. Several inpainted results comparisons in three datasets are shown in Fig. I, II, and III respectively. The selected images are challenging for they are occluded by large ratio masks, *e.g.* the right part of the house in the second image in Fig. I is almost completely obscured. We can see that the results generated by our algorithm are more semantically plausible compared to the other methods. For the damaged images with large mask ratios and different mask shapes at different locations, our model can still reason the structure and restore the image.

C. Discussion

Different Mask Types: We classify the masks into several categories. Fig. 8 shows some examples. It includes object-like, bounding box type, stroke type, random noise, and watermark type. We conduct experiments to figure out the shortcomings of the single model. The experiment settings are designed as follows: Train several models with different masks. Each mask is employed for a single model especially. We compare the results from two aspects: 1) Test a model with different mask types. 2) For a certain mask type, test different models trained with different mask types. We show experimental results in the upper part of the Table IV. We can see that model trained with a single mask type generally performs worse when testing with other mask types. This experiment validates our motivation that it is meaningful to inpaint different mask types with different models. Thus, we propose to integrate these N models with an ensemble model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3174413, IEEE Transactions on Multimedia

Fig. 5. Comparisons on Paris StreetView. From the left to the right: masked image, inpainted results by PIC, PConv, GIP, RFR-Net, MRIN (ours), accordingly.

TABLE II
QUANTITATIVE COMPARISON ON CELEBA. *HIGHER IS BETTER. †LOWER IS BETTER.

Dataset		CelebA		
Mask Ratio		10%-20%	30%-40%	50%-60%
SSIM*	PIC	0.965	0.881	0.672
	PConv	0.977	0.922	0.791
	GatedConv	0.973	0.914	0.767
	EdgeConnect	0.975	0.915	0.759
	PRVS	0.978	0.926	0.799
	RFR-Net	0.981	0.934	0.819
	MRIN (Ours)	**0.983**	**0.937**	**0.823**
PSNR*	PIC	30.67	24.74	19.29
	PConv	32.77	26.94	22.14
	GatedConv	32.56	26.72	21.47
	EdgeConnect	32.48	26.62	21.49
	PRVS	33.05	27.24	22.37
	RFR-Net	**33.56**	**27.76**	**22.88**
	MRIN (Ours)	32.96	27.14	22.45
Mean $l_1^\dagger$	PIC	0.0111	0.0314	0.0749
	PConv	0.0083	0.0236	0.0524
	GatedConv	0.0088	0.0245	0.0561
	EdgeConnect	0.0088	0.0247	0.0572
	PRVS	0.0079	0.0224	0.0500
	RFR-Net	0.0075	0.0212	0.0470
	MRIN (Ours)	**0.0072**	**0.0208**	**0.0461**

TABLE III
QUANTITATIVE COMPARISON ON PARIS STREET VIEW. *HIGHER IS BETTER. †LOWER IS BETTER.

Dataset		Paris Street View		
Mask Ratio		10%-20%	30%-40%	50%-60%
SSIM*	PIC	0.930	0.785	0.519
	PConv	0.947	0.835	0.619
	GatedConv	0.953	0.849	0.621
	EdgeConnect	0.950	0.849	0.646
	PRVS	0.953	0.854	0.659
	RFR-Net	**0.954**	0.862	0.681
	MRIN (Ours)	0.953	**0.865**	**0.693**
PSNR*	PIC	29.35	23.97	19.52
	PConv	30.76	25.46	21.39
	GatedConv	31.32	25.54	20.61
	EdgeConnect	31.19	26.04	21.89
	PRVS	31.49	26.17	22.07
	RFR-Net	31.71	26.44	22.40
	MRIN (Ours)	**31.76**	**26.62**	**22.57**
Mean $l_1^\dagger$	PIC	0.0140	0.0379	0.0799
	PConv	0.0123	0.0313	0.0623
	GatedConv	0.0120	0.0309	0.0660
	EdgeConnect	**0.0110**	0.0286	0.0582
	PRVS	0.0111	0.0281	0.0562
	RFR-Net	**0.0110**	0.0275	0.0546
	MRIN (Ours)	0.0111	**0.0269**	**0.0543**

When testing with a certain mask type A, its corresponding model A will play a leading role in inpainted results. Specifically, the weight heatmap $H_m(i)$ corresponding to model A will be close to 1 in Equation 5. Such an ensemble model is more capable of handling different mask types, comparing with any single model. To validate this motivation, we conduct experiments to compare our model with RFR-Net [21] and GIP [19] with a mixture of multiple mask types. Specifically, the mask of the training set consists of 5 types, each of which accounts for 20%. The quantitative comparison results are shown in the bottom half of the Table IV. We can see that our model outperforms all other models. It is difficult for a single model to grasp the difference between different mask types. Therefore, these methods still perform badly though they are trained with different mask types simultaneously. Our model can utilize different patch generators to handle different mask types so it shows promise performance compared with the best single model.

Different Patch Generators: To validate that the proposed MRIN can actually learn to separate different mask types and tackle them differently, we show local inpainted results generated by the patch generators for several mask types. As shown in Fig. 7, there exist several mask types in the damaged image. For example, the mask types include object-like, bounding box, and stroke types. We select some patches from the testing images and show different generated results in Fig. 7. We show the local inpainted results predicted by several mask selectors. We can see that for a specific patch, there exists a patch generator that can inpaint better results than others. In the final weighted summation process, the weights denoting this patch generator will be larger. Therefore, it will play a dominant role in the inpainted results so that our MRIN can separate and tackle different mask types. Thus, MRIN can generate more promising results for the damaged image with complex mask types than the existing single model.

TABLE IV
QUANTITATIVE COMPARISON OF DIFFERENT MASK TYPES. THE NUMBER IN THE i-TH ROW AND j-TH COLUMN IS THE L_1 LOSS OF A MODEL TRAINED WITH THE i-TH MASK TYPE AND TEST ON THE j-TH TYPE.

Mask Type	1	2	3	4	5
Object-like	**0.064**	0.103	0.043	0.062	0.091
Bounding Box	0.182	**0.101**	0.203	0.075	0.115
Stroke Type	0.143	0.153	**0.013**	0.102	0.098
Random Noise	0.076	0.124	0.061	**0.045**	0.085
Watermark Type	0.091	0.115	0.086	0.079	**0.067**
Mixture (RFR)	0.073	0.101	0.017	0.062	0.071
Mixture (GIP)	0.075	0.103	0.022	0.067	0.076
MRIN	0.066	0.101	0.014	0.048	0.068

Computation Complexity of MRIN: The average inference time of an image in a single GPU is 650 ms (batch size

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3174413, IEEE Transactions on Multimedia

Fig. 6. An overview of the patch-wise inpainting process. Each picture shows the inpainted result at a step. The red bounding box indicates the patch to be inpainted at the current step.

TABLE V
THE EFFECTS OF THE COMPONENTS IN MRIN

Setting	SSIM	PSNR	Mean l_1
Random Patch	0.594	18.92	0.0647
Sliding Windows	0.600	19.02	0.0593
Whole Image	0.583	18.84	0.0681
Random Mask Map	0.588	18.96	0.0714
Same Mask Type	0.585	18.79	0.0753
Without L_{mask}	0.596	19.02	0.0649
Without L_m	0.599	19.07	0.0614
Without L_c	0.598	19.05	0.0591
MRIN	**0.602**	**19.10**	**0.0576**

TABLE VI
THE EFFECTS OF THE RECURSIVE STEP NUMBER T

T	15	20	25	30	35	45
SSIM	0.581	0.593	0.598	0.601	**0.602**	0.602
PSNR	18.37	18.74	18.96	19.05	**19.10**	19.08
mean l1	0.0774	0.0691	0.0653	0.0607	0.0576	**0.0571**

TABLE VII
THE EFFECTS OF THE MASK TYPE NUMBER N

N	2	3	4	5	7	10
SSIM	0.577	0.583	0.589	0.592	0.593	**0.594**
PSNR	18.63	18.66	18.87	**18.90**	18.89	18.89
mean l1	0.0693	0.0617	0.0615	0.0606	0.0603	**0.0600**

of 2, resolution of 256×256). Though the MRIN requires multiple steps, it still achieves comparable inference time with other state-of-the-art methods. This is because the inpainting model just inpaints a small patch of the image at a step. Additionally, the mask-robust agent uses a simple ConvLSTM architecture and its forwarding process performs much faster than the inpainting model.

D. Ablation Studies

To validate the effects of several components in our module and the effects of some hyper-parameters, we perform ablation studies on Place2 dataset with the mask ratio of 50%-60%.

Effects of the Patch Locator: To verify the ability of the patch locator to exploit optimal patch sequence and the effectiveness of the patch-wise manner, we conduct experiments in three settings: 1) We randomly pick the patch at each time step rather than the heatmap generated by the patch selector, named as "random patch". To guarantee that the selected patches can cover all the masks in the image in limited steps, we only picked the patch containing new regions. 2) We use sliding windows to pick the patch. 3) We remove the patch selector and feed the adaptive generative networks with the whole image, named as "whole image". The quantitative comparisons are shown in Table V. The mean l_1 loss increases from 5.76% to 6.81% and 6.47% in the two settings, which indicates the effectiveness of the patch locator in exploring meaningful sequence patches.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3174413, IEEE Transactions on Multimedia

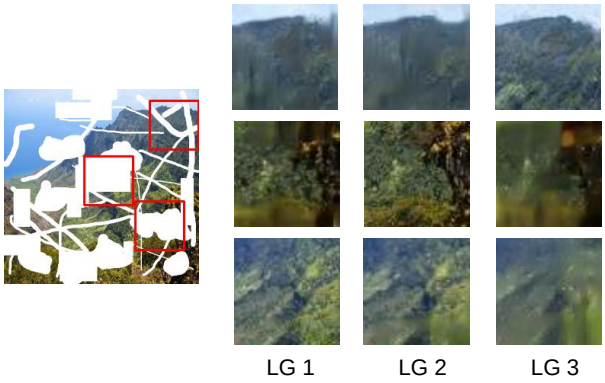

Fig. 7. The local inpainted results generated by different patch generators. LG is short for local generator.

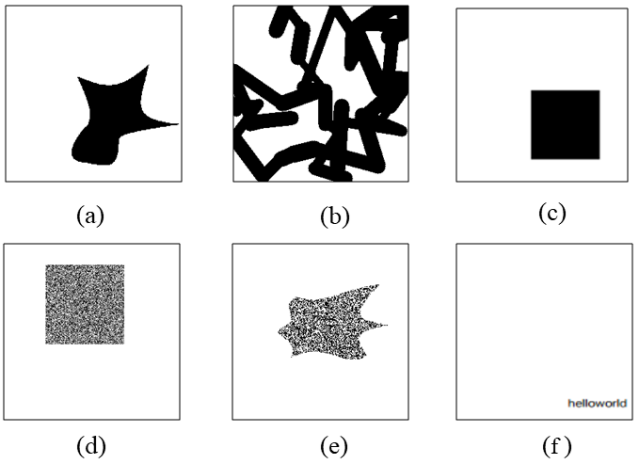

Fig. 8. Examples of some mask types. They are (a) object-like, (b) stroke type, (c) bounding box, (d) and (e) random noise, (f) watermark type, respectively.

Effects of the Mask Selector: To verify the effect of the mask selector, we conduct experiments in two settings: 1) We remove the mask selector module and set the number of the decoder branches in the patch generative networks as 1, named as "the same mask type". 2) We randomly generate the mask attention map H_m, named as "random mask map". As shown in Table V, in the two settings, the mean L_1 loss increases by a large margin, and SSIM/PSNR drop from 0.602/19.10 to 0.585/18.79 and 0.588/18.96, respectively, which verifies the effect of the mask selector and demonstrates that the mask type can be decomposed into several basic mask types.

Effects of Components of the Mask Loss: The mask selector assists the adaptive generative networks to inpaint for different mask types, which is guided by the mask Loss L_{mask}. To verify the effect of its components, we conduct experiments in three settings: 1) Remove L_e from the mask loss. 2) Remove L_c from the mask loss. 3) We train the mask selector only with similarity loss L_{SIM} and without L_{mask}. The quantitative comparisons are shown in Table V. We can find that two components of max loss are essential for the mask selector to assist the local generators in inpainting the patch adaptively. With the guide of mask loss, the mask selector can explore the local mask types and is not just an ensemble model. To validate the effectiveness of the l_2 loss in Equation 13, we remove this loss term and conduct experiment with different mask type number N from 2 to 10. This setting aims to exploit whether the mask selector can learn information on mask types autonomously. We show results in Table VII. We can see that without the labels of mask types, the performance decrease to a certain degree. However, the mask selector still exploits the differences in a masked image and guides the mask decomposition process with the supervisory signals of L_c and L_e. the mask decomposition ability of the mask selector S is stronger with the N becoming larger. Thus, the performance improves obviously when the mask num is below 5. The improvement becomes minor when the N is large enough for mask select to express various mask types. Too larger N will lead to a waste of inference time and memory resources.

Effects of Maximum Time Steps T: The performance variations with different parameter settings T are shown in Table VI. When T is relatively low, the performance is bad for the selected patches are unable to cover the whole image. The performance gradually improves with T increasing. The improvement becomes minor when T is relatively large especially T is larger than 30, because the extracted patches can cover the whole image. We visualize the patch selection order for a few examples to exploit the questions whether the process starts with patches with small or large mask ratios, whether the order follow the spatial order of the patches. Fig. 6 displays the patch-wise inpainting procedure. We can see that our MRIN tends to first inpaint the marginal area of the image. The reason may be that these regions are easy to be inpainted without prior knowledge for they are usually background areas. Secondly, the MRIN turns to inpaint patches with large mask ratios. This is because the agent can get a higher reward by inpainting such patches. Completing patches with large mask ratios can significantly reduce the loss L_{SIM} so that the agent is more likely to inpaint the whole image in fewer steps. Finally, the model refines some detailed areas in the picture at the last few steps to complete the inpainting process.

V. Conclusions

In this paper, we have proposed the Mask-Robust Inpainting Network (MRIN) which sequentially decomposes the mask and selects a patch to inpaint at each step. The mask-robust agent is optimized in a deep reinforcement learning way to explore the optimal inpainting route and generate mask attention maps. Then, the adaptive patch generators inpaint the selected patch based on its mask type. Thus, our MRIN inpaints different mask types with different patch generators in a patch-wise way. Extensive experiments on three popular datasets demonstrate our model achieves state-of-the-art performance with quantitative and qualitative comparisons, which validates the effectiveness of our method. In the future work, we will increase the interpretability of the model to figure out how different adaptive patches can learn the generation of different mask patches.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62125603 and Grant U1813218, and in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: CGIT. (2000) 417–424

[2] Liao, L., Xiao, J., Wang, Z., Lin, C.w., Satoh, S.: Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. (2020)

[3] Dai, Q., Chopp, H., Pouyet, E., Cossairt, O., Walton, M., Katsaggelos, A.K.: Adaptive image sampling using deep learning and its application on x-ray fluorescence image reconstruction. TMM **22**(10) (2019) 2564–2578

[4] Tang, N.C., Hsu, C.T., Su, C.W., Shih, T.K., Liao, H.Y.M.: Video inpainting on digitized vintage films via maintaining spatiotemporal continuity. TMM **13**(4) (2011) 602–614

[5] Wang, Q., Fan, H., Sun, G., Ren, W., Tang, Y.: Recurrent generative adversarial network for face completion. TMM **23** (2020) 429–442

[6] Song, L., Cao, J., Song, L., Hu, Y., He, R.: Geometry-aware face completion and editing. In: AAAI. Volume 33. (2019) 2506–2513

[7] Shetty, R.R., Fritz, M., Schiele, B.: Adversarial scene editing: Automatic object removal from weak supervision. In: NIPS. (2018) 7706–7716

[8] Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM ToG **28**(3) (2009) 24

[9] Zeng, Y., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. (2020)

[10] Liu, H., Jiang, B., Song, Y., Huang, W., Yang, C.: Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. (2020)

[11] Liu, J., Yang, S., Fang, Y., Guo, Z.: Structure-guided image inpainting using homography transformation. TMM **20**(12) (2018) 3252–3265

[12] Schmeing, M., Jiang, X.: Faithful disocclusion filling in depth image based rendering using superpixel-based inpainting. TMM **17**(12) (2015) 2160–2173

[13] Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: ECCV. (2018) 85–100

[14] Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019)

[15] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR. (2016) 2536–2544

[16] Vo, H.V., Duong, N.Q., Pérez, P.: Structural inpainting. In: ACM MM. (2018) 1948–1956

[17] Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., Luo, J.: Foreground-aware image inpainting. In: CVPR. (2019) 5840–5848

[18] Yeh, R.A., Chen, C., Yian Lim, T., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with deep generative models. In: CVPR. (2017) 5485–5493

[19] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: CVPR. (2018) 5505–5514

[20] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: ICCV. (2019) 4471–4480

[21] Li, J., Wang, N., Zhang, L., Du, B., Tao, D.: Recurrent feature reasoning for image inpainting. In: CVPR. (2020) 7760–7768

[22] Guo, Z., Chen, Z., Yu, T., Chen, J., Liu, S.: Progressive image inpainting with full-resolution residual network. In: ACM MM. (2019) 2496–2504

[23] Oh, S.W., Lee, S., Lee, J.Y., Kim, S.J.: Onion-peel networks for deep video completion. In: ICCV. (2019) 4403–4412

[24] Li, J., He, F., Zhang, L., Du, B., Tao, D.: Progressive reconstruction of visual structure for image inpainting. In: ICCV. (2019) 5962–5971

[25] Cao, Q., Lin, L., Shi, Y., Liang, X., Li, G.: Attention-aware face hallucination via deep reinforcement learning. In: CVPR. (2017) 690–698

[26] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. TPAMI **40**(6) (2017) 1452–1464

[27] Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV. (2015) 3730–3738

[28] Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.A.: What makes paris look like paris? ACM ToG **31**(4) (2012) 1–9

[29] Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. TIP **12**(8) (2003) 882–889

[30] Chan, T.F., Shen, J.: Nontexture inpainting by curvature-driven diffusions. JVCIR **12**(4) (2001) 436–449

[31] Ding, D., Ram, S., Rodríguez, J.J.: Image inpainting using nonlocal texture matching and nonlinear filtering. TIP **28**(4) (2018) 1705–1719

[32] Fang, Y., Yu, K., Cheng, R., Lakshmanan, L.V., Lin, X.: Efficient algorithms for densest subgraph discovery. arXiv preprint arXiv:1906.00341 (2019)

[33] Li, K., Wei, Y., Yang, Z., Wei, W.: Image inpainting algorithm based on tv model and evolutionary algorithm. Soft Computing **20**(3) (2016) 885–893

[34] Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-in by joint interpolation of vector fields and gray levels. TIP **10**(8) (2001) 1200–1211

[35] Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: CGIT. (2001) 341–346

[36] Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: ICCV. Volume 2., IEEE (1999) 1033–1038

[37] Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. TPAMI **29**(3) (2007) 463–476

[38] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014) 2672–2680

[39] Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM ToG **36**(4) (2017) 1–14

[40] Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-net: Image inpainting via deep feature rearrangement. In: ECCV. (2018) 1–17

[41] Lahiri, A., Jain, A.K., Agrawal, S., Mitra, P., Biswas, P.K.: Prior guided gan based semantic inpainting. In: CVPR. (2020) 13696–13705

[42] Wang, Y., Chen, Y.C., Tao, X., Jia, J.: Vcnet: A robust approach to blind image inpainting. (2020)

[43] Wang, T., Ouyang, H., Chen, Q.: Image inpainting with external-internal learning and monochromic bottleneck. In: CVPR. (2021) 5120–5129

[44] Zhou, Y., Barnes, C., Shechtman, E., Amirghodsi, S.: Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations. In: CVPR. (2021) 2266–2276

[45] Liao, L., Xiao, J., Wang, Z., Lin, C.W., Satoh, S.: Image inpainting guided by coherence priors of semantics and textures. In: CVPR. (2021) 6539–6548

[46] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540) (2015) 529

[47] Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: AAAI. (2016)

[48] Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N.: Dueling network architectures for deep reinforcement learning. arXiv preprint arXiv:1511.06581 (2015)

[49] Jie, Z., Liang, X., Feng, J., Jin, X., Lu, W., Yan, S.: Tree-structured reinforcement learning for sequential object localization. In: NIPS. (2016) 127–135

[50] Krull, A., Brachmann, E., Nowozin, S., Michel, F., Shotton, J., Rother, C.: Poseagent: Budget-constrained 6d object pose estimation via reinforcement learning. In: CVPR. (2017) 6702–6710

[51] Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3-4) (1992) 229–256

[52] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. (2014)

[53] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)

[54] Goodrich, B., Arel, I.: Reinforcement learning based visual attention with application to face detection. In: CVPRW, IEEE (2012) 19–24

[55] Caicedo, J.C., Lazebnik, S.: Active object localization with deep reinforcement learning. In: ICCV. (2015) 2488–2496

[56] Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: NIPS. (2014) 2204–2212

[57] Gregor, K., Danihelka, I., Graves, A., Rezende, D.J., Wierstra, D.: Draw: A recurrent neural network for image generation. arXiv preprint arXiv:1502.04623 (2015)

[58] Sun, Y., Liang, D., Wang, X., Tang, X.: Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873 (2015)

[59] Liang, X., Lee, L., Xing, E.P.: Deep variation-structured reinforcement learning for visual relationship and attribute detection. In: CVPR. (2017) 848–857

[60] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML, PMLR (2015) 2048–2057

[61] Xiong, C., Merity, S., Socher, R.: Dynamic memory networks for visual and textual question answering. In: ICML, PMLR (2016) 2397–2406

[62] Zheng, C., Cham, T.J., Cai, J.: Pluralistic image completion. In: CVPR. (2019) 1438–1447
[63] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. (2015)