SketchEdit: Mask-Free Local Image Manipulation with Partial Sketches

Yu Zeng
Johns Hopkins University
yzeng22@jhu.edu

Zhe Lin
Adobe Research
zlin@adobe.com

Vishal M. Patel
Johns Hopkins University
vpatel36@jhu.edu

Figure 1. Our system allows users to interactively manipulate an image by sketching desired contours directly on top of the image.

Abstract

Sketch-based image manipulation is an interactive image editing task to modify an image based on input sketches from users. Existing methods typically formulate this task as a conditional inpainting problem, which requires users to draw an extra mask indicating the region to modify in addition to sketches. The masked regions are regarded as holes and filled by an inpainting model conditioned on the sketch. With this formulation, paired training data can be easily obtained by randomly creating masks and extracting edges or contours. Although this setup simplifies data preparation and model design, it complicates user interaction and discards useful information in masked regions. To this end, we investigate a new paradigm of sketch-based image manipulation: mask-free local image manipulation, which only requires sketch inputs from users and utilizes the entire original image. Given an image and sketch, our model automatically predicts the target modification region and encodes it into a structure agnostic style vector. A generator then synthesizes the new image content based on the style vector and sketch. The manipulated image is finally produced by blending the generator output into the modification region of the original image. Our model can be trained in a self-supervised fashion by learning the reconstruction of an image region from the style vector and sketch. The proposed method offers simpler and more intuitive user workflows for sketch-based image manipulation and provides better results than previous approaches. More results, code and interactive demo will be available at https://zengxianyu.github.io/sketchedit.

1. Introduction

Recently there have been increasing efforts and demand for building interactive photo editing tools on devices with touch interfaces. Being expressive and easily editable, sketching is one of the most straightforward ways that people illustrate their creative ideas and interact with the apps [2, 10, 25, 44]. Sketch-based image editing is an emerging research topic where the goal is to build models which can manipulate holistic or local structures of an image according to the user-drawn sketches. It has made a significant progress in the last few years with advancements in deep learning and generative models, *e.g.*, GANs [11].

There are two challenges in sketch-based image manipulation: (1) The input sketch roughly indicates where to modify, but the precise modification region is unknown, and

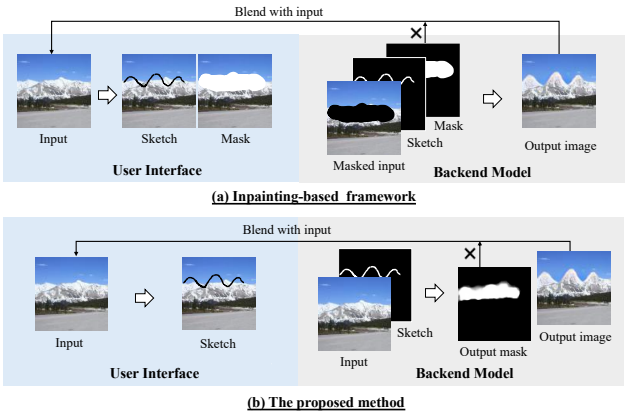

Figure 2. Illustration of the previous inpainting-based image manipulation systems and the proposed one with only sketch input.

(2) it is difficult to collect large-scale image pair data (*i.e.* images before and after manipulation) for training. Previous approaches avoid these obstacles by converting sketch-based image manipulation into a conditional inpainting problem. Users are required to draw an extra mask to indicate the modification regions in addition to the sketch. The masked regions are regarded as holes and filled by an inpainting model conditioned on the sketch, , as illustrated in Fig. 2 (a). Under this inpainting framework, paired training data can be easily obtained by randomly generating masks as in general image inpainting [17, 21, 24, 26, 33, 35, 38, 40, 42, 47, 48, 51, 53, 54, 56, 59–61, 64] and extracting edges/contours in the masked regions as surrogate sketches using edge detection algorithms [1, 5, 6, 15, 23, 27, 28, 41, 46, 50, 52]. Then the problem can be solved by training an inpainting model to predict the original image given the masked image, mask, and sketch as input. This clever formulation simplifies model design and training setup, and has been explored extensively in the recent literature [19, 29, 36, 55, 58].

However, the inpainting framework is sub-optimal for sketch-based image manipulation. First, although it eases model design and training, it leaves extra work to users and results in a complicated user interface. After users sketch in an image region, they have to draw a mask again around the same location. This procedure is tedious and redundant. An untrained user may not be able to draw a good mask that exactly covers the desired modification area, making the model difficult to produce reasonable results. Second, with this framework, masked regions in an input image has to be removed to match the training condition of the inpainting model. However, since the original content in a modification region is usually highly correlated with the desired result, ignoring this information will degrade the quality of the manipulated image and lead to unwanted changes in the modification region.

To this end, we investigate a new paradigm of sketch-based image manipulation, which requires only sketch inputs from users while leveraging the entire original image. Our system provides a more straightforward and user friendly interface for image manipulation: to solely sketch directly on top of the original image, as illustrated in Fig. 2 (b). Moreover, as the system takes the entire image as input, information in the modification region can be reserved, resulting in more consistent appearance (*e.g.* color, texture) with the original content in the modification region.

For local editing, it is desired to preserve most of the original image content and only modify the relevant image region surrounding the sketch input. Therefore, we first predict the modification region with a mask estimator and then synthesize new content inside with a generator. The manipulated image is produced by blending the generator output into the original image using the predicted mask. To encourage the structure of the synthesized content to follow the sketch while only keeping the style of the original content, we encode the modification region of an input image into a structure agnostic style vector with a style encoder. The system can be trained in a self-supervised fashion by learning to reconstruct the target modification region based on the style vectors and sketches.

We evaluate our method on multiple datasets, including CelebAHQ, Places2, and newly constructed datasets SketchFace and SketchImg containing sketches and masks drawn by users. Extensive experiments demonstrate that the proposed method outperforms the state-of-the-art approaches. To show the advantage of our new framework in terms of user interaction, we include an interactive demo in the supplementary material. Our contributions are as follow,

- Investigation of a new paradigm of sketch-based image manipulation: mask-free local image manipulation that requires only partial sketch inputs from users.
- The first system for mask-free sketch-based local image manipulation, including the network architecture, data acquisition, and training strategy.
- Extensive experiments are conducted on multiple datasets to demonstrate the superiority of our approach over the related methods.

2. Related Work

Sketch-guided image completion. Previous research on sketch-based image manipulation is studied mainly under a conditional image inpainting framework. Yu *et al.* [58] propose DeepFill-v2 which can perform sketch-guided image inpainting of general images as well as face images. The gated convolution layer is introduced to select useful features from the incomplete input dynamically. Portenier *et al.* [36] propose FaceShop for sketch-based face images manipulation through conditional image completion. FaceShop allows users to modify the local shape and color in a face

image by drawing the mask, sketch, and a few color strokes. Jo and Park [19] explore face manipulation with sketch and color strokes through image completion. They use free-form masks and style loss to train the image completion model to handle irregular and possibly large missing areas. Yang *et al*. [55] focus on the adaptation of the face manipulation model trained on sketches generated by edge detection to the human-drawn sketches. A sketch refinement strategy is proposed, which first dilates then refines a user-drawn sketch to close to an edge detection result. Our work is inspired by these methods but focuses on the manipulation of complete images. We do not require the users to draw an extra mask to construct an incomplete image when editing a photo. Instead, the region to edit is automatically discovered and modified based on where and what they draw.

Image-to-image translation. Image translation aims to learn the mapping between different image domains, *e.g*. to generate images from semantic label maps, edges, *etc*. Isola *et al*. [18] propose an image-to-image translation framework, called Pix2Pix, to generate images from label maps or edge maps. Zhu *et al*. [65] propose CycleGAN, which allows training an image translation model on unpaired data with a cycle consistency constraint. Park *et al*. [34] propose spatially-adaptive normalization for image generation conditioned on semantic layout, which modulates the activations using input semantic layouts to propagate semantic information throughout the network better. Zhang *et al*. [62] propose a framework for exemplar-based image translation with cross-domain correspondence by jointly learning cross-domain correspondence and image translation. When dealing with the edge input, image translation methods usually require a complete edge map and generate a brand new image. In contrast, our method focuses on local image manipulation with partial sketches as input.

Contour-based image manipulation is another line of research where an image is edited in the contour domain. Approaches in this category typically encode an image into an edge-based representation and decode after users modify the edges. Elder *et al*. [8] encode images with the brightness on edges and perform image editing operations (crop, paste, delete) in the contour domain. Dekel *et al*. [3] propose to encode images with learned features at contour points to perform complex changes in the image domain by simple edits of contours. Vinker *et al*. [43] explore deep image manipulation with a single training sample by extensive augmentation of the training sample through thin-plate splines (TPS) warping. A conditional generator trained on the augmented samples can map the edited edges and segmentation to the modified image. These methods require the contour representation to be complete from which the image can be faithfully reconstructed and assume that the users have basic photo editing skills to perform certain editing operations on a pre-computed edge map. In comparison, our method allows a broader range of users to manipulate an image more easily, *i.e*. by directly sketching on top of the input image.

3. Proposed Method

We focus on local image manipulation based on free-form sketches. Given an input image x and partial sketch c indicating the target contour, our model modifies a local region of x to match the sketch, resulting in a manipulated image y. Due to the lack of data for supervised training, we propose a self-supervised learning approach to jointly learn mask estimation and masked region reconstruction.

3.1. Model

Our model consists of three components: mask estimator, structure agnostic style encoder, and generator. Fig. 3 shows the overall structure of our model. Given an input image x and sketch c, the generator synthesizes new content based on the image, sketch, and a structure agnostic style vector produced by the style encoder. The mask estimator predicts a mask to blend the synthesized content into the original image at a proper location. In what follows, we describe each of these components in detail.

Mask estimator. The mask estimator M is an encoder-decoder style network that maps an image x and a partial sketch map c to a mask m, *i.e*. $m = M(x, c)$. The mask is produced by applying the sigmoid function on the output of a convolutional layer, which is a gray-scale map of range $[0, 1]$. We use the term mask for consistency with previous inpainting-based approaches. Each element of the mask represents how likely the corresponding pixel of the input image should be modified to match the sketch. Thus, we can separate the modification region and non-modification static region by multiplying an input image x with m and $1 - m$ element-wise, respectively. Since we are interested in the style information in the modification region, we define a style partial image as $x^{sty} = m \odot x$, where $\odot$ represents element-wise multiplication. We define a static partial image $x^{sta} = (m - 1) \odot x$ as the non-modification region of the input image.

Structure agnostic style encoder. The goal of the style encoder S is to extract the style information from the modification region of the input image but get rid of the structure information. It is composed of stacked convolutional layers followed by a global max pooling, which produces a d-dimensional style vector v^{sty} given the style partial image x^{sty} and the mask m, *i.e*. $v^{sty} = S(x^{sty}, m)$. By repeating v^{sty} by $h \times w$ times, we can obtain a style feature $\hat{v}^{sty}_{(d \times h \times w)}$ of arbitrary spatial size $h \times w$ while keeping position invariant. However, it may still be sensitive to spatial transformation due to the local connectivity of convolutional layers. To further encourage the extracted style features to be structure agnostic, we apply random warping and deterioration to x^{sty} in training, which will be introduced in Sec. 3.2.

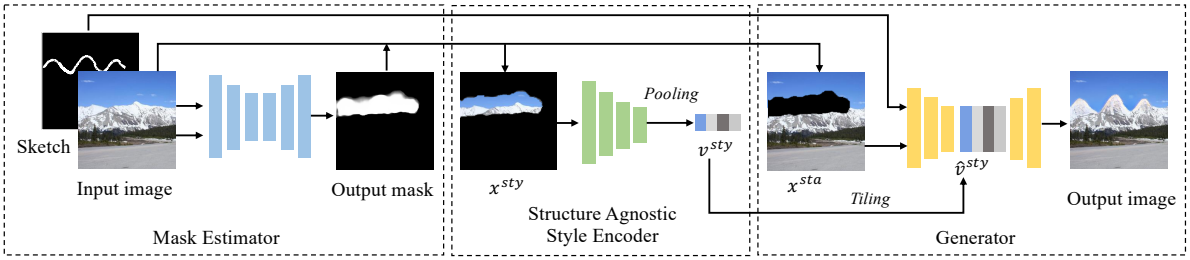

Figure 3. Model structure of the proposed method. Given an input image x and sketch c, the mask estimator predicts a mask m indicating the region to edit. The style partial image x^{sty} and the static partial image x^{sta} can be created by multiplying x with m and $1-m$ element-wise, respectively. Then the generator synthesizes new content in the modification region based on the static image x^{sta}, sketch c, and a structure agnostic style feature extracted from x^{sty} by the style encoder.

Generator. The proposed framework is general and does not depend on the specific architecture of the generator. We can combine it with most existing inpainting-based models by using the estimated modification regions as missing regions and tiling the style vector into style feature maps of the suitable size as a generator's input. In our implementation, we incorporate the network architecture of DeepFill-v2 [58] due to its simplicity and lightweight. The generator G consists of a coarse stage G_0 and refinement stage network G_1. The coarse stage takes the partial image, estimated mask, and sketch as input. We use the style feature maps $\hat{v}^{sty}$ as the side input at the bottleneck of the coarse stage network. The output of the coarse stage is denoted as y_0, which is passed through the refinement stage to produce a refined output y_1:

$$\begin{aligned} y_0 &= G_0(x^{sta}, m, \hat{v}^{sty}) \\ y_1 &= G(x^{sta}, m, \hat{v}^{sty}) = G_1(y_0). \end{aligned} \tag{1}$$

The final result y is produced by blending the refinement stage output into the original image using the estimated mask

$$y = y_1 \odot m + x \odot (1 - m). \tag{2}$$

To train the generator, we minimize the L1 error between the output y_0, y_1, y and the original image x. We also use adversarial training with a discriminator to encourage visual realism of the blended image y. The detailed training strategy and loss functions will be described in the next section.

3.2. Learning by reconstruction

Due to the lack of training data, previous inpainting-based approaches resort to self-supervised learning, specifically, by learning to reconstruct the original image from a masked image and sketch. In this work, we also seek to train our model using self-supervision. However, as our model infers the mask, directly learning by reconstructing the original image will lead to a trivial solution. The mask estimator will constantly predict an all-zero mask; the generator becomes an identity mapping; the style encoder does not need to learn any meaningful representation. We solve this issue by deteriorating the input images with random local warping and regional dropout before passing them through the model. Fig. 4 shows a training iteration of the proposed method.

Local warping. Before passing an input image through the model in training, we warp x in a random area using triangular local warping. We construct a triangular mesh by placing vertices on the image boundary, boundary of a randomly sampled region, and inside the region. Then a warping flow can be created by moving interior vertices while keeping other vertices unmoved. For general images, we place and move the interior vertices randomly. For face images, to make the warped faces remain visually plausible, we use facial landmarks and blendshapes to guide the selection and movements of the interior vertices. The detailed warping algorithm can be found in the supplementary material. Fig. 5 shows samples augmented by warping. We can see that the warped images have the same color and texture as the original ones but with different structures in warped regions. Therefore, we can construct the original image with structure information from the sketch and style information extracted from a warped image. Also, as a warped image is identical to the original image in non-warped regions, the mask estimator will be encouraged to predict zero where no sketch appears. Thus the regions irrelevant to sketches can be preserved.

Bi-directional mask regularization. Since we warp the input images in training, the mask predictor may end up producing masks by simply detecting warping artifacts and fail in the inference stage, as shown in thee third column of Fig. 6. To avoid this behavior, we train the mask estimator with a bi-directional image reconstruction loss as regularization. Let $f(x)$ denote the image produced by distorting x with the warping flow f. By applying the same warping flow to the sketch c, we obtain a sketch $f(c)$ corresponding to the warped image. During training, we add an extra decoder to the mask estimator to predict an image, with the goal of estimating the original image x from the warped image $f(x)$ and the original sketch c, and the opposite, to estimate $f(x)$ from x and $f(c)$. Thus the mask estimator M has an extra output in addition to the mask output $m = M(x, c)$. For input im-

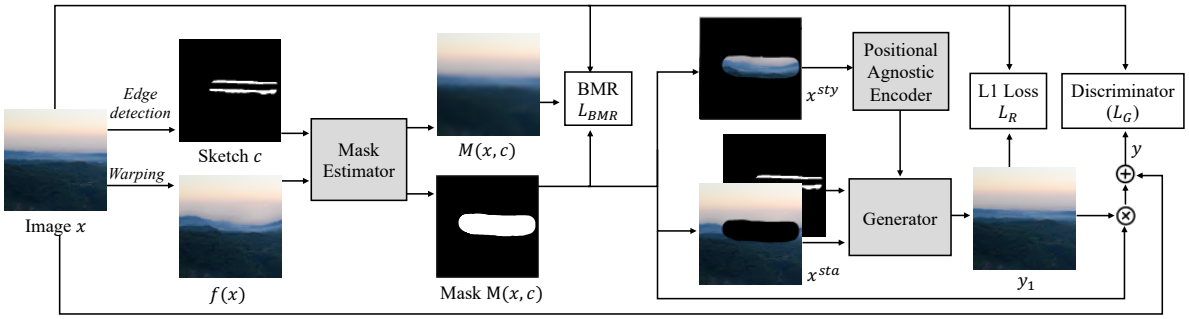

Figure 4. Illustration of a training iteration. The mask estimator, style encoder, and generator are trained jointly by learning to reconstruct the original images. To avoid the trivial solution of generating all-zero masks, we warp the input image in a local areal before passing them through the mask estimator. A bi-directional mask regularization loss (BMR) is applied to help the mask estimator produce reasonable masks rather than simply detecting warping artifacts.

Figure 5. Example images augmented by local warping. Top: original images, bottom: warped images.

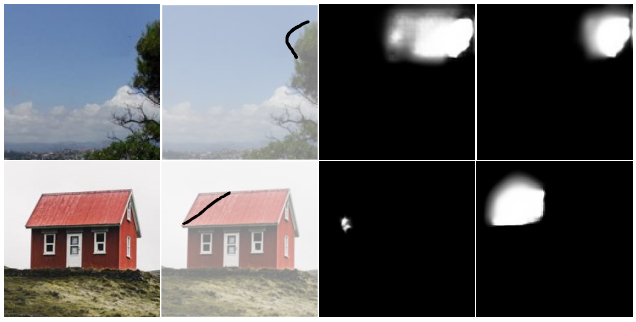

Figure 6. The predicted masks with and without the bi-directional mask regularization (BMR). Top: warped images, bottom: real images. BMR helps the mask estimator generalize better to real image inputs.

age x and sketch c, we let $\bar{M}(x,c)$ represent the estimated image and $\hat{M}(x,c)$ represent the image blended using the mask, *i.e.* $\hat{M}(x,c) = \bar{M}(x,c) \odot M(x,c) + x \odot (1 - M(x,c))$. The bi-directional mask regularization (BMR) loss is defined as follows:

$$L_{BMR} = \left\| \bar{M}(f(x),c) - x \right\|_1 + \left\| \bar{M}(x, f(c)) - f(x) \right\|_1 + \left\| \hat{M}(f(x),c) - x \right\|_1 + \left\| \hat{M}(x, f(c)) - f(x) \right\|_1 .$$

Since L_{BMR} is computed for both the warped and the original images, the mask estimator can be adapted to natural image input and thus can predict reasonable masks in the inference stage, as shown in the forth column of Fig. 6.

Regional dropout. Warping helps the style encoder learn structure agnostic representation by creating structural variation while preserving the style in a local region. However, the warped image and the original image usually have identical visual elements, while real cases might involve the generation of new elements. Therefore, during training, we randomly dropout sampled regions in the style image x^{sty} before passing it through the style encoder to reduce the correlation between v^{sty} and x.

Loss functions. For a pair of image x and sketch map c in the training set, we compute the following loss functions to jointly train the mask estimator, style encoder, and the generator:

$$L_R = \|y_0 - x\|_1 + \|y_1 - x\|_1 + \|y - x\|_1 . \quad (3)$$

Eqn. 3 measures the error of the coarse stage output y_0 and refinement stage output y_1, as well as the error of the manipulated image y, which is the combination of x and y_1 blended using the estimated mask. In addition,

$$L_G = \text{ReLU}[\mathbb{1} - D(y,c)], \quad (4)$$

where D represents the discriminator. For the discriminator, we use the hinge loss and spectral normalization [32] in [57, 58]. Eqn. 4 defines the adversarial loss inspired by GANs. To minimize L_G, the generator and mask estimator have to cooperate reasonably to fool the discriminator. It requires the generator to synthesize visually realistic content and the mask estimator to provide the mask that blends the synthesized content into a proper region in the original image seamlessly. The overall loss function is the sum of L_G,

L_R and the mask regularization term L_{BMR}:

$$L_{total} = L_R + L_G + L_{BMR}. \quad (5)$$

4. Experiments

As introduced in Sec. 3.1, our framework can be combined with most inpainting-based models to enable mask-free manipulation. In our implementation, we plug the DeepFill-v2 generator into our framework due to its simplicity and efficiency. To verify the effectiveness of the proposed method, we compare the results under our mask-free framework to the original DeepFill-v2 as a baseline. We also compare with other state-of-the-art inpainting-based face manipulation methods including SC-FEGAN [19] and DeepPS [55] to demonstrate the superiority of our approach.

4.1. Datasets & Evaluation Metric

We train our general image manipulation model on the training split of Places [39]. For face manipulation, we train the model on CelebHQ [20] for consistency with the inpainting-based approaches. We randomly sample 2,000 images as the validation set and use the remaining data for training.

For evaluation, we use edge maps extracted by [15] as surrogate sketches for Places and obtain the sketches in face regions by connecting detected facial landmarks for the CelebAHQ dataset. We randomly sample modification regions for each test image and obtain partial sketches by discarding all edges outside this region. For evaluation of our method, we apply local warping in modification regions. For inpainting-based methods, we set pixel values in modification regions equal to zero according to their training condition. For synthetic samples, the original images can be seen as the ground-truth. Hence, we evaluate the performance using PSNR, L1 error, SSIM, and FID [16] between the results and the original images. PSNR, L1 error and SSIM are based on pixel-wise errors, often used in image inpainting and restoration tasks [7, 12–14, 22, 31, 37, 45, 49, 57, 63]. FID measures the distance between the distribution of the deep features of generated images and real images. It is the current standard metric for assessing the quality of generative models [30].

Apart from using synthetic samples for evaluation, to reflect the performance in real cases, we construct two test sets SketchImg and SketchFace for face and general image manipulation, respectively. These two datasets contain 100 natural images and 400 face images as well as the corresponding sketches and masks drawn by human users. The samples are created under the conditional inpainting framework, where the users draw a sketch indicating the target contour and the mask that marks the region to modify. To evaluate our method, we simply ignore the masks and only use the images and sketches as input. For the real cases, the results are supposed to have a similar style to the input images. Therefore, we evaluate the performance using style loss [9], *i.e.* the mean squared error between the Gram matrices of the deep features of the input image and results. We use the feature maps from the `relu_1` and `relu_2` of a VGG [39] network pretrained on ImageNet [4]. We denote the obtained style loss as SL_1 and SL_1, respectively. We also report FID between the results and input images as a reflection of the image quality.

4.2. Qualitative Evaluation

Fig. 7 presents the face manipulation results of our method and the state-of-the-art methods. As our model leverages entire images as input, it can better reserve the original appearance in the modification region of the input image, *e.g.* the eyebrows in the firs row and the lip color in the second row. Our method can also produces visually realistic results for more challenging general image manipulation. Fig. 8 shows the general image manipulation results of our method and DeepFill-v2. It can be seen that our method well preserves the colors and textures of the original images. In contrast, DeepFill-v2 often adds extra changes in the modification regions, *e.g.* it changes the colors of the island, arch, and the beak in the first, second, and forth row, respectively. In addition, as our framework prevents the information loss caused by masking out the modification area, our results are less blurry and have more details.

4.3. Quantitative Evaluation

Table 1 reports the quantitative evaluation results on the synthetic samples from the CelebAHQ and Places validation sets. It can be seen that our results have larger PSNR, SSIM and smaller L1 error than previous approaches, which imply a smaller difference between the generated images and the corresponding ground-truth. Our results also have smaller FID, indicating a closer distance to the distribution of original images. Table 2 reports the quantitative evaluation results on real samples. It can be seen that the style loss of our results are significantly smaller than previous approaches, as our mask-free framework effectively keeps the style of the input images. Our method also outperforms previous approaches in terms of FID on real samples.

4.4. Ablation Study

Mask estimator. The mask estimator is an important component in our framework to enable local manipulation and encourage the model to learn better generative capability. Without the mask predictor, we may train a common conditional GAN on training samples synthesized by warping. However, a model trained to mimic warping often fails in challenging cases involving generating new elements or deleting existing elements. For example, as shown in Fig. 9, the generator jointly trained with the mask estimator correctly deletes and

Figure 7. Visual comparison of face manipulation results. Input*: input sketch and masks for inpainting-based methods. Input: input sketch for our method.

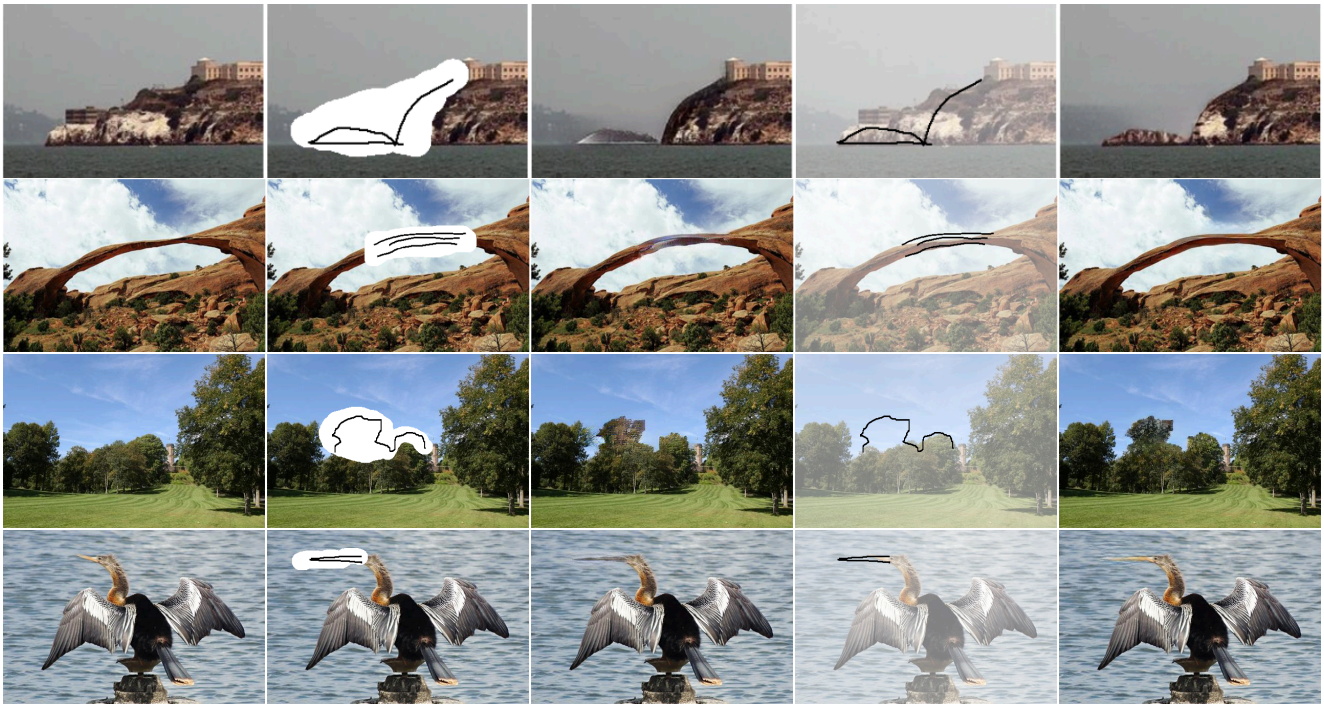

Figure 8. Visual comparison of general image manipulation results. Input*: input sketch and masks for inpainting-based methods. Input: input sketch for our method.

Table 1. Quantitative comparison on synthetic samples from CelebAHQ (top) and Places (bottom) validation sets.

CelebAHQ				
Method	L1 error↓	PSNR↑	SSIM↑	FID↓
SC-FEGAN	0.0110	30.39	0.9408	2.693
DeepPS	0.0125	30.54	0.9340	2.460
DeepFill-v2	0.0115	30.59	0.9370	2.345
Ours	**0.0072**	**34.15**	**0.9604**	**0.844**
Places				
Method	L1 error↓	PSNR↑	SSIM↑	FID↓
DeepFill-v2	0.0293	24.06	0.8442	2.238
Ours	**0.0223**	**26.35**	**0.8527**	**1.500**

Table 2. Quantitative comparison on SketchFace and SketchIMG containing sketches and masks drawn by human users.

	SketchFace				SketchImg	
Method	SC-FEGAN	DeepPS	DeepFill-v2	Ours	DeepFill-v2	Ours
FID↓	4.092	5.285	4.382	**2.94**	27.11	**24.27**
SL_1↓	2484	3797	2859	**2001**	5314	**4374**
SL_2↓	13145	15585	13970	**10850**	24169	**21575**

Image Sketch w/o ME with ME

Figure 9. Results with and without the mask estimator (ME).

adds teeth in manipulation of mouths. In contrast, a common conditional GAN only squeezes or stretches the mouth. The mask estimator also helps to better preserve the non-modification regions. Without the mask estimator, slight changes in the static region is almost inevitable. This is reflected by the larger L1 error and smaller PSNR in the third row of Table 3. Another baseline without the mask estimator is to use masks generated based on rules. The third row reports the results with the rule-based masks given by the minimum bounding boxes enclosing sketches. By comparing the first row and the third row, we can see the significant improvements brought by the mask estimator.

Style encoder. The style encoder is essential to keep the appearance of the modification region. The model without the style encoder has no access to information in the modification region and generates new content based on prior, resulting in unexpected changes. Fig. 10 shows example results for face manipulation. It can be seen that the model

Table 3. Effect of each components. Rule-based Mask: masks given by the minimum bounding boxes enclosing sketches, ME: mask estimator, SE: style encoder. The first row corresponds to the full model with both ME and SE.

w/o Mask	Rule-based Mask	ME	SE	L1 error↓	PSNR↑
		✓	✓	0.0072	34.15
✓			✓	0.0127	32.50
	✓		✓	0.0102	31.87
		✓		0.0074	33.98

Image Sketch w/o SE with SE

Figure 10. Results with and without the style encoder (SE).

with the style encoder reserves the colors of the lips, while the one without the style encoder changes the colors. The quantitative evaluation results also reflect the effect of the style encoder. The first and the forth row of Table 3 report the quantitative evaluation results with and without the style encoder, respectively. It can be seen that the model with the style encoder results in smaller L1 error and larger PSNR.

5. Conclusion, Limitations and Future Work

This paper presents a new paradigm of sketch-based image manipulation, *i.e.* mask-free local image manipulation, and propose a complete system implementation for this task including the network architecture, data acquisition and training strategy. The proposed method offers simpler and more robust user workflows for sketch-based image manipulation and provides consistently better results than the related approaches. A limitation is that the current framework only supports structure editing based on monochrome sketches. Color editing based on colored sketches under the proposed new mask-free framework is an interesting topic for future work. Potential negative social impact of this work might include: fake news can be created by manipulating photos; The spreading of manipulated images may distort of people's perception of body images and beauty. The negative impact can be mitigated by embedding digital watermarks in manipulated images to make them easy to identify.

Acknowledgement This work was supported by an ARO grant W911NF-21-1-0135.

References

[1] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[2] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018.

[3] Tali Dekel, Chuang Gan, Dilip Krishnan, Ce Liu, and William T Freeman. Sparse, smart contours to represent and edit images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image batabase. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[5] Ruoxi Deng and Shengjun Liu. Deep structural contour detection. In *ACM Int. Conf. Multimedia*, 2020.

[6] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *European Conference on Computer Vision*, 2018.

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2015.

[8] James H Elder and Richard M Goldberg. Image editing in the contour domain. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1998.

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[10] Arnab Ghosh, Richard Zhang, Puneet K Dokania, Oliver Wang, Alexei A Efros, Philip HS Torr, and Eli Shechtman. Interactive sketch & fill: Multiclass sketch-to-image translation. In *International Conference on Computer Vision*, 2019.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Conference on Neural Information Processing Systems*, 27, 2014.

[12] Pengfei Guo, Jeya Maria Jose Valanarasu, Puyang Wang, Jinyuan Zhou, Shanshan Jiang, and Vishal M. Patel. Over-and-under complete convolutional rnn for mri reconstruction. In Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 13–23, Cham, 2021. Springer International Publishing.

[13] Pengfei Guo, Puyang Wang, Jinyuan Zhou, Shanshan Jiang, and Vishal M. Patel. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2423–2432, June 2021.

[14] Wei Han, Shiyu Chang, Ding Liu, Mo Yu, Michael Witbrock, and Thomas S Huang. Image super-resolution via dual-state recurrent networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[15] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3828–3837, 2019.

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Conference on Neural Information Processing Systems*, 30, 2017.

[17] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36(4):1–14, 2017.

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[19] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *International Conference on Computer Vision*, pages 1745–1753, 2019.

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[21] Avisek Lahiri, Arnav Kumar Jain, Sanskar Agrawal, Pabitra Mitra, and Prabir Kumar Biswas. Prior guided gan based semantic inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[22] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *European Conference on Computer Vision*, 2018.

[23] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In *European Conference on Computer Vision*, 2020.

[24] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Image inpainting guided by coherence priors of semantics and textures. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[25] Fang Liu, Xiaoming Deng, Yu-Kun Lai, Yong-Jin Liu, Cuixia Ma, and Hongan Wang. Sketchgan: Joint sketch completion and recognition with generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[26] 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 *European Conference on Computer Vision*, 2018.

[27] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *European Conference on Computer Vision*, 2020.

[28] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. Pd-gan: Probabilistic diverse gan for image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[29] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, Jing Liao, Bin Jiang, and Wei Liu. Deflocnet: Deep

image editing via flexible low-level controls. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[30] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Conference on Neural Information Processing Systems*, 2018.

[31] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, June 2021.

[32] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[33] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

[34] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[35] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[36] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Transactions on Graphics*, 2018.

[37] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[38] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *International Conference on Computer Vision*, 2019.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[40] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C-C Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. In *European Conference on Computer Vision*, pages 3–19, 2018.

[41] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikainen, and Li Liu. Pixel difference networks for efficient edge detection. In *International Conference on Computer Vision*, 2021.

[42] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Distillation-guided image inpainting. In *International Conference on Computer Vision*, pages 2481–2490, 2021.

[43] Yael Vinker, Eliahu Horwitz, Nir Zabari, and Yedid Hoshen. Deep single image manipulation. *arXiv preprint arXiv:2007.01289*, 2020.

[44] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *International Conference on Computer Vision*, pages 14050–14060, 2021.

[45] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[46] Yi Wang, Ying-Cong Chen, Xin Tao, and Jiaya Jia. Vcnet: A robust approach to blind image inpainting. In *European Conference on Computer Vision*, 2020.

[47] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *Conference on Neural Information Processing Systems*, 2018.

[48] Jing Xiao, Liang Liao, Qiegen Liu, and Ruimin Hu. Cisi-net: Explicit latent content inference and imitated style rendering for image inpainting. In *he AAAI Conference on Artificial Intelligence*, 2019.

[49] Minshan Xie, Menghan Xia, and Tien-Tsin Wong. Exploiting aliasing for manga restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[50] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *International Conference on Computer Vision*, pages 1395–1403, 2015.

[51] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[52] Dan Xu, Wanli Ouyang, Xavier Alameda-Pineda, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. 2018.

[53] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[54] Jie Yang, Zhiquan Qi, and Yong Shi. Learning to incorporate structure knowledge for image inpainting. In *he AAAI Conference on Artificial Intelligence*, 2020.

[55] Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. Deep plastic surgery: Robust and controllable image editing with human-drawn sketches. In *European Conference on Computer Vision*, 2020.

[56] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[57] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[58] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *International Conference on Computer Vision*, 2019.

[59] Tao Yu, Zongyu Guo, Xin Jin, Shilin Wu, Zhibo Chen, Weiping Li, Zhizheng Zhang, and Sen Liu. Region normalization for image inpainting. In *he AAAI Conference on Artificial Intelligence*, 2020.

[60] Yu Zeng, Zhe Lin, Huchuan Lu, and Vishal M Patel. Cr-fill: Generative image inpainting with auxiliary contextual reconstruction. In *International Conference on Computer Vision*, pages 14164–14173, 2021.

[61] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *European Conference on Computer Vision*. Springer, 2020.

[62] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5143–5153, 2020.

[63] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019.

[64] Yuqian Zhou, Connelly Barnes, Eli Shechtman, and Sohrab Amirghodsi. Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2266–2276, 2021.

[65] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision*, pages 2223–2232, 2017.