LaFIn: Generative Landmark Guided Face Inpainting

Yang Yang[1], Xiaojie Guo[1], Jiayi Ma[2], Lin Ma[3], and Haibin Ling[4]
[1]Tianjin University [2]Wuhan University [3]Tencent AI Lab [4]Stony Brook University
yangyangcic@tju.edu.cn, {xj.max.guo, jyma2010, haibin.ling}@gmail.com, forestlma@tencent.com

Abstract

It is challenging to inpaint face images in the wild, due to the large variation of appearance, such as different poses, expressions and occlusions. A good inpainting algorithm should guarantee the realism of output, including the topological structure among eyes, nose and mouth, as well as the attribute consistency on pose, gender, ethnicity, expression, etc. This paper studies an effective deep learning based strategy to deal with these issues, which comprises of a facial landmark predicting subnet and an image inpainting subnet. Concretely, given partial observation, the landmark predictor aims to provide the structural information (e.g. topological relationship and expression) of incomplete faces, while the inpaintor is to generate plausible appearance (e.g. gender and ethnicity) conditioned on the predicted landmarks. Experiments on the CelebA-HQ and CelebA datasets are conducted to reveal the efficacy of our design and, to demonstrate its superiority over state-of-the-art alternatives both qualitatively and quantitatively. In addition, we assume that high-quality completed faces together with their landmarks can be utilized as augmented data to further improve the performance of (any) landmark predictor, which is corroborated by experimental results on the 300W and WFLW datasets. The code is available on https://github.com/YaN9-Y/lafin

1. Introduction

Image inpainting (*a.k.a.* image completion) refers to the process of reconstructing lost or deteriorated regions of images, which can be applied to, as a fundamental component, various tasks such as image restoration and editing [1, 30]. Undoubtedly, one expects the completed result to be realistic, so that the reconstructed regions can be hardly perceived. Compared with natural scenes like oceans and lawns, manipulating faces, the focus of this work, is more challenging. Because the faces have much stronger topological structure and attribute consistency to preserve. Figure 1 shows three such examples. Very often, given the observed clues, human beings can easily infer what the lost parts possibly, although inexactly, look like. As a consequence, a slight violation on the topological structure and/or the attribute consistency in the reconstructed face highly likely leads to a significant perceptual flaw. The following gives the definition of the problem:

Figure 1: Three face completion results by our method. From left to right: corrupted inputs, plus landmarks predicted from the inputs, and our final results, respectively.

Definition 1. *Face Inpainting. Given a face image I with corrupted regions masked by M. Let $\overline{M}$ designate the complement of M, and $\circ$ the Hadamard product. The goal is to fill the target part with semantically meaningful and visually continuous information to the observed part. In other words, the completed result $\hat{I} := M \circ \hat{I} + \overline{M} \circ I$ should preserve the topological structure among face components such as eyes, nose and mouth, and the attribute consistency on like pose, gender, ethnicity and expression.*

1.1. Previous Arts

Various image inpainting methods have been developed over the last decades. In what follows, we briefly review classic and contemporary works closely related to ours.

Traditional Inpainting Methods. In this category, diffusion-based and patch-based approaches are two representative branches. Diffusion-based approaches [2, 5, 32] iteratively propagate low-level features around the occluded areas. However, these methods are limited to reconstructing structureless and small-size regions. While patch-based methods [1, 4, 8] attempt to copy similar blocks from either the same image or a set of images to the target regions. On the one hand, their computational cost of calculating the similarity between blocks is expensive, even though some works like [1] have been proposed towards accelerating the procedure. On the other hand, as a common limitation, they all hypothesize that the missing part can be found elsewhere, which does not always hold in practice.

Deep Learning-based Methods. Recently, deep learning based methods have become the mainstream for image inpainting. The context encoder [22], as a pioneer deep-learning method for image completion, introduces an encoder-decoder network trained with an adversarial loss [6]. After that, plenty of follow-ups have been proposed to improve the performance from various aspects. For instance, the scheme by [9] employs both the global and local discriminators to accomplish the task. Another attempt proposed in [34] designs a coarse-to-fine network structure and applies a self-attention layer to connect related features at distant spatial locations. Besides, Yu *et al.* and Liu *et al.* [33, 16] upgrade the convolutional layers for making networks adaptive to the masked input. However, most of the above-mentioned methods can barely keep the structure of the original image, and the inpainted result frequently tends to be blurry, especially on large occluded areas. For the sake of maintaining the structure of corrupted images, a number of methods, such as [20, 31], try to first predict the edge information for corrupted images, then apply it as a condition to guide the inpainting. These methods work well on small corrupted regions though, when the corruption becomes larger, the performance significantly degrades as it is not easy to predict reasonable edges inside the masked regions, leading to unsatisfactory results.

Deep Face Inpainting Methods. Specific to face completion, the authors of [15] construct a loss taking care of the gap in semantic segmentation (face parsing) between generated face images and ground truth, expecting to better preserve the structure. But this work often suffers from the color inconsistency, and lacks of ability in handling faces with large poses. Besides, [11, 33] directly ask users to manually label edges for generating corresponding results. Although providing a flexible way to editing faces, sometimes it is difficult/inconvenient for users to input precise edge information. To relive the requirement from users, Nazeri *et al.* applied a network to predict the edges [20], which however suffers from inaccurate/unreasonable prediction on large holes. Moreover, we argue that, for face completion, both face parsing and edge information are relatively redundant, which may even degenerate the performance when feeding slightly inaccurate information into the inpainting module. Facial landmarks are better to act as the indicator, which are neat, sufficient, and robust to reflect the structure of face, please see Fig. 2 for an example. Many works have successfully applied landmarks to the task of face generation, such as [36], [26] and [35]. It is worth noting that, different from the generation task [36] and [35], in our problem, the landmarks need to be obtained from the corrupted images.

Figure 2: An illustration of different facial features. From left to right: the input, Canny edges, landmarks, edges by connecting the landmarks, and parsing regions.

1.2. Challenges and Considerations

As stated previously, completing face images in the wild is challenging. A qualified face inpainting algorithm should carefully take into account the following two concerns to guarantee the realism of output:

- Faces are of strong structure. The topological relationship among facial features including eyebrows, eyes, nose and mouth is always well-organized. The completed faces must satisfy this topology structure primarily;
- The attributes of face, such as pose, gender, ethnicity, and expression, should be consistent across the inpainted regions and the observed part.

Otherwise, a slight violation on these two factors will result in a significant perceptual flaw.

Why adopt landmarks? This work employs facial landmarks as structural guidance, because of their compactness, sufficiency, and robustness. One may ask whether the edge or parsing information provide more powerful guidance than the landmarks? If the information is precise, the answer is yes. But, taking the strategy using edges [20] as an example, it is not easy to generate reasonable edges in challenging situations like large-area corrupted faces with large-poses. Under the circumstances, the redundant and inaccurate information would instead hurt the performance. Alternatively, a set of landmarks (pre-defined fiducial points)

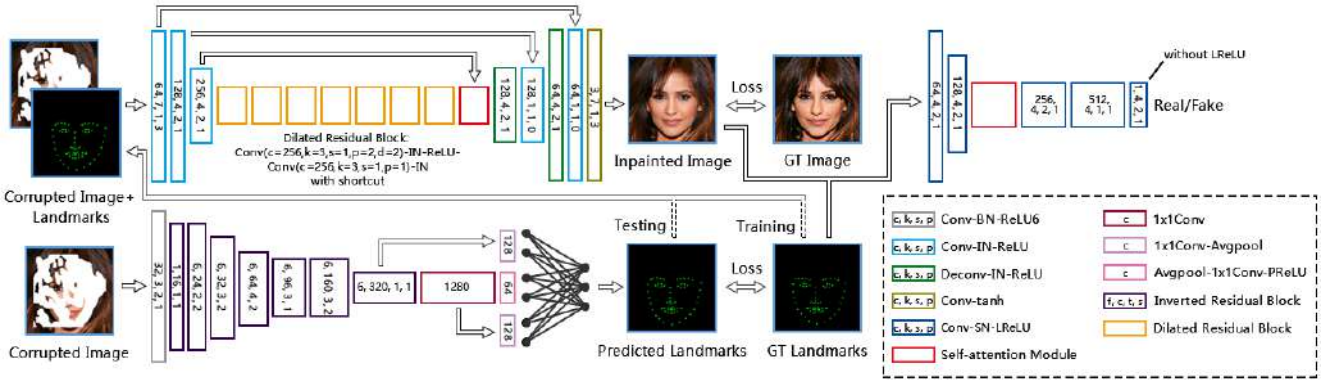


Figure 3: The architecture of the proposed model. For a corrputed image, the landmarks are first estimated by the landmark prediction module. Then the inpaint module applies the landmarks as prior to inpaint the image. The notations c, k, s and p stand for the channel number, kernel size, stride, and padding, respectively. In addition, for each inverted residual block, it contains a sequence of identical layers repeating t times, and the expansion factor is f.

always exists, no matter what situation the face is in. Further, the landmarks can be viewed as the discrete/ordered points sampled on the **key** edges/regions of face, which are sufficient to conversely reform the **key** edges/facial regions (face parsing) with redundant information removed. Compared with the edge [20] and parsing [15] information, for one thing, the landmarks are much neater and more robust, please see Fig. 2 for illustration. For another thing, once the landmarks for a face are obtained, the topology structure, pose and expression can be subsequently determined. Moreover, the landmarks are more convenient to control from the editing perspective. These properties support that using landmarks is a better choice for face completion.

How to guarantee the attribute consistency? Except for the pose and expression attributes determined by the landmarks, there are several other attributes, such as gender, ethnicity, and makeup style, need to be concerned. Notice that the consistency is to bridge the observed and the inpainted regions. This is to say, for these finer-grained attributes, the inpainting algorithm should take the observed (real) information as reference for the reconstruction. Harnessing distant spatial context (large receptive field) and connecting temporal feature maps (long-short term) can effectively fulfill the requirement.

1.3. Contributions

This paper presents a deep network, namely Generative **La**ndmark Guided **F**ace **In**paintor (LaFIn for short), which comprises of a facial landmark predicting subnet and an image inpainting subnet, for solving the face inpainting problem. The main contributions can be summarized in the following aspects.

- As analyzed, facial landmarks are neat, sufficient, and robust to act as the indicator for face inpainting. We construct a module for predicting landmarks on incomplete faces, which reflect the topological structure, pose and expression of the target face.
- To complete faces, we design an inpainting subnet that employs the predicted landmarks as guidance. For the attribute consistency, the subnet harnesses distant spatial context and connects temporal feature maps.
- Extensive experiments are conducted to reveal the efficacy of our design and, show its advances over state-of-the-art alternatives both qualitatively and quantitatively.

In addition, we can further use the completion results to help boosting the performance of data-driven landmark detectors. Since, in real situations, the training data are often insufficient, and manually labeling landmarks is time-consuming, a simple yet reliable data augmentation manner is definitely desired. Our another contribution is as follows:

- The completion can generate various plausible new faces **conditioned** on the landmarks. Thus, the generated face and the corresponding (ground-truth) landmarks can be employed as the augmented data to relieve the pressure from manual annotation. The effectiveness of this manner is confirmed by experimental results on the WFLW and 300W datasets.

2. Methodology

A desired face inpaintor should generate natural-looking results with logical structures and attributes. We build a deep network, denoted as LaFIn, to achieve the goal. As schematically illustrated in Fig. 3, the network is composed of a subnet for predicting landmarks, and another one for generating new pixels conditioned on the predicted landmarks. In the next subsections, we shall detail the network.

2.1. Landmark Prediction Module

The landmark prediction module $\mathcal{G}_L$ aims to retrieve a set of ($n = 68$ in this work) landmarks from a corrupted face image $I^M := I \circ M$, *i.e.* $\hat{L} \in \mathbb{R}^{2\times n} := \mathcal{G}_L\left(I^M; \theta_L\right)$, with θ_L the trainable parameters. Technically, $\mathcal{G}_L$ can be accomplished by any landmark detector like [29, 14, 28]. Please notice that, for the target inpainting task, what we expect from the landmarks is more about the underlying topology structure and some attributes (pose and expression) than the precise location of each individual landmark. The following may explain the reason: considering the landmarks on face contour for an example, with the corresponding region fixed, shifting them along the contour will not affect the final result much. Consequently, we build a simple yet sufficiently effective $\mathcal{G}_L$. Our $\mathcal{G}_L$ is built upon the MobileNet-V2 model proposed in [24], which focuses on feature extraction. The final landmark prediction is achieved by fully connecting the fused feature maps at different rear stages, as illustrated in Fig. 3. The training loss for $\mathcal{G}_L$ is simply as follows:

$$\mathcal{L}_{lmk} := \|\hat{L} - L_{gt}\|_2^2, \tag{1}$$

where L_{gt} denotes the ground-truth landmarks. In addition, $\|\cdot\|_2$ stands for the ℓ_2 norm.

2.2. Image Inpainting Module

The inpainting network $\mathcal{G}_P$ desires to complete faces by taking corrupted images I^M and their (predicted or ground-truth) landmarks L ($\hat{L}$ or L_{gt}) as input, *i.e.* $\hat{I} := \mathcal{G}_P\left(I^M, L; \theta_P\right)$, with θ_P the network parameters. This subnet comprises of a generator and a discriminator.

Generator. Overall, the generator is based on the U-Net structure. More specifically, the network consists of three gradually down-sampled encoding blocks, followed by 7 residual blocks with dilated convolutions and a long-short term attention block. Then, the decoder processes the feature maps gradually up-sampled to the same size as input. The long-short attention layer [37] is harnessed to connect temporal feature maps, and the stacked dilated blocks are to enlarge the receptive filed so that features in a wider range can be taken into account. Besides, shortcuts are added between corresponding encoder and decoder layers. Moreover, the 1×1 convolution operation is executed before each decoding layer as the channel attention to adjust weights of features from the shortcut and last layer. In such a way, the network can better make use of distant features both spatially and temporally. The structure of the generator can be found in Fig. 3, and more in Appendix.

Discriminator. Based on the concept of two-player game, the generator tries to produce completed faces conditioned on the landmarks to fool the discriminator, while the discriminator aims to determine whether the generated result satisfies the data distribution. The convergence is reached when the generated results are not distinguishable from the real ones. In this work, our discriminator is built upon the 70×70 Patch-GAN architecture [10]. To stabilize the training process, we introduce the spectral normalization (SN) [19] into the blocks of the discriminator. Besides, an attention layer is inserted to adaptively treat the features. It is worth to notice that the works like [9] employ two discriminators, *i.e.* a global discriminator focuses on the entire image to assess if it is coherent as a whole, and a local one looks only at the completed region to ensure the local consistency. Differently, our discriminator adopts only one judger to accomplish the job, which takes an image and its landmarks as input, *i.e.* $\mathcal{D}(I, L; \theta_D)$ with θ_D the parameters. The reasons are: 1) the generated results are conditioned on the landmarks, already ensuring the global structure; and 2) the attention layer concentrates more on the attribute consistency. The configuration of our discriminator can be found in Fig. 3, and more details in Appendix.

Loss. We use a combination of a per-pixel loss, a perceptual loss , a style loss, a total variation loss and an adversarial loss, for training the inpaintor.

(I) The per-pixel loss is defined as follows:

$$\mathcal{L}_{pixel} := \frac{1}{N_m}\|\hat{I} - I\|_1, \tag{2}$$

where $\|\cdot\|_1$ stands for the ℓ_1 norm. Notice that we use the mask size N_m as the denominator to adjust the penalty. It means that if a face is interfered by a small occlusion, the inpainted result should be very close to the ground-truth, while if the corruption is large, the restriction can be relaxed as long as the structure and consistency are rational.

(II) The perceptual loss measures the difference of feature maps extracted from a pre-trained network, which is calculated in the following manner:

$$\mathcal{L}_{perc} := \sum_p \frac{\|\phi_p(\hat{I}) - \phi_p\left(I\right)\|_1}{N_p \times H_p \times W_p}, \tag{3}$$

where $\phi_p(\cdot)$ denotes the N_p feature maps with size $H_p \times W_p$ of the p-th layer from the pre-trained network. $relu1\_1$, $relu2\_1$, $relu3\_1$, $relu4\_1$ and $relu5\_1$ of the VGG-19 pre-trained on the ImageNet [23] are utilized to calculated the perceptual loss, as well as the style loss described below.

(III) The style loss computes the style distance between two images as follows:

$$\mathcal{L}_{style} := \sum_p \frac{1}{N_p \times N_p}\|\frac{G_p(\hat{I} \circ M) - G_p(I \circ M)}{N_p \times H_p \times W_p}\|_1, \tag{4}$$

where $G_p(x) = \phi_p(x)^T\phi_p(x)$ stands for the Gram Matrix corresponding to $\phi_p(x)$.

(IV) The total variation loss is utilized to suppress the

checkerboard artifact, which is defined as:

$$\mathcal{L}_{tv} := \frac{1}{N_I}\|\nabla \hat{I}\|_1, \tag{5}$$

where N_I is the pixel number of I, and ∇ is the first order derivative, containing ∇_h (horizontal) and ∇_v (vertical).
(V) The adversarial loss adopts the LSGAN proposed in [18], due to its stability during the training process and the advance in visual quality, which is as follows:

$$\begin{aligned}\mathcal{L}_{adv_G} &:= \mathbb{E}[(\mathcal{D}(\mathcal{G}_P(I^M, L), L_{gt}) - 1)^2], \\ \mathcal{L}_{adv_D} &:= \mathbb{E}[\mathcal{D}(\hat{I}, L_{gt})^2] + \mathbb{E}[(\mathcal{D}(I, L_{gt}) - 1)^2].\end{aligned} \tag{6}$$

The total loss with respect to the generator yields:

$$\begin{aligned}\mathcal{L}_{inp} := \mathcal{L}_{pixel} + \lambda_{perc}\mathcal{L}_{perc} + \lambda_{sty}\mathcal{L}_{style} \\ + \lambda_{tv}\mathcal{L}_{tv} + \lambda_{adv}\mathcal{L}_{adv_G}.\end{aligned} \tag{7}$$

We use $\lambda_{perc} = 0.1$, $\lambda_{style} = 250$, $\lambda_{tv} = 0.1$ and $\lambda_{adv} = 0.01$ in our experiments. The whole training procedure alternatively minimizes $\mathcal{L}_{inp}$ for the generator $\mathcal{G}_P$ and $\mathcal{L}_{adv_D}$ for the discriminator $\mathcal{D}$ until converged.

2.3. Training Strategy

The generator is desired to complete image via $\hat{I} := \mathcal{G}(I^M)$. For face images, their strong regularity, like the landmarks considered by our design $\hat{I} := \mathcal{G}_P(\mathcal{G}_L(I^M), I^M)$, could benefit model reduction and training procedure, as the space is considerably restricted by the regularity. Intuitively, the training for $\mathcal{G}_P$ and $\mathcal{G}_L$ can be finished jointly. Technically, it is feasible. But, in practice, it is not a good choice. The reasons are as follows: 1) the loss for $\mathcal{G}_L$, say $\mathcal{L}_{lmk}$, computes over a small number of (only 68 in this work) **locations**, which is incompatible with $\mathcal{L}_{inp}$. In other words, the parameter tuning is extremely hard; and 2) even with the well-tuned parameters, the performance of both $\mathcal{G}_L$ and $\mathcal{G}_P$ may be too inaccurate especially at the beginning of training, which consequently leads to low-quality landmark prediction and inpainting results. These two coupled factors very likely drag the training into dilemmas, like bad points of convergence and/or high prices of training. Thus, we decouple the joint model into the landmark prediction and inpainting modules, and train them separately. It is worth to note that we actually have trained the model in a joint way with different carefully-tuned settings, the best shot is still inferior to our separate training. In experiments shown in this work, the landmark prediction model and the inpainting model are trained using 256×256 images and optimized by the Adam optimizer [13] with $\beta_1 = 0$ and $\beta_2 = 0.9$, and the learning rate $= 10^{-4}$. The learning rate of the discriminator is 10^{-5}. We use batch size $= 16$ for the landmark prediction module and batch size $= 4$ for the inpainting model.

	Mask	CA	PIC	EC	Ours
PSNR	10-20%	27.51	30.33	30.73	**31.48**
	20-30%	24.42	27.05	27.56	**28.31**
	30-40%	22.14	24.68	25.34	**26.14**
	40-50%	20.29	22.58	23.44	**24.22**
	50%+	18.10	19.54	20.71	**21.61**
	Center	24.13	24.22	24.82	**25.92**
SSIM	10-20%	0.942	0.968	0.971	**0.975**
	20-30%	0.892	0.936	0.942	**0.951**
	30-40%	0.832	0.894	0.907	**0.922**
	40-50%	0.761	0.838	0.859	**0.883**
	50%+	0.646	0.715	0.754	**0.805**
	Center	0.864	0.870	0.874	**0.905**
FID	10-20%	7.29	2.72	2.33	**2.05**
	20-30%	14.62	4.49	3.89	**3.36**
	30-40%	24.41	6.56	5.94	**5.11**
	40-50%	38.83	9.24	8.78	**7.12**
	50%+	51.21	13.32	13.92	**10.84**
	Center	7.39	**4.98**	8.26	6.63

Table 1: Quantitative comparison on the CelebA-HQ dataset in terms of PSNR, SSIM and FIC on random and center masks.

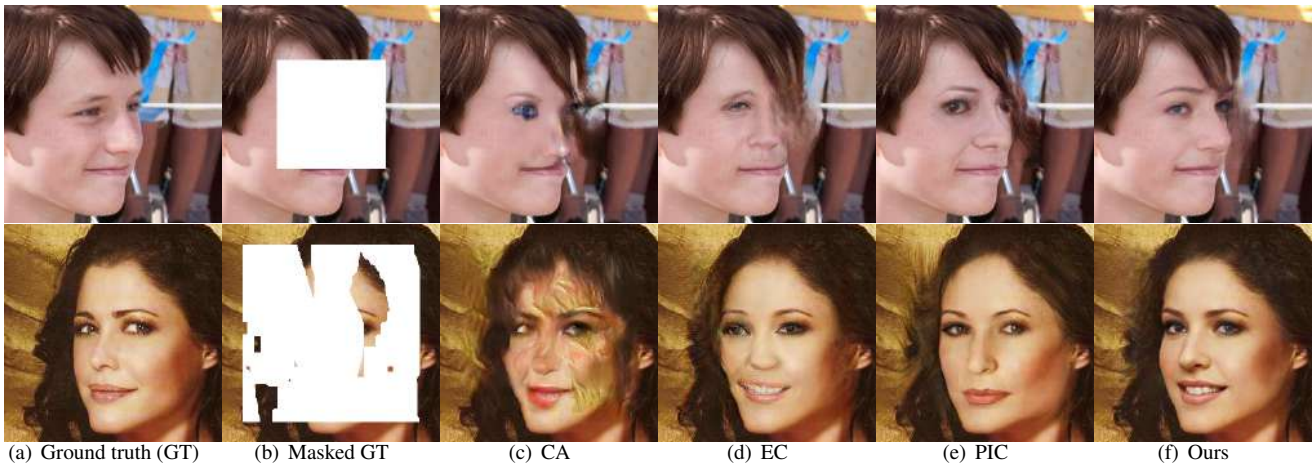
Metric	CE	GFC	EC	Ours
PSNR	25.46	21.04	25.83	**26.25**
SSIM	0.909	0.766	0.899	**0.912**
FID	**1.731**	14.958	3.519	3.512

Table 2: Quantitative comparison on the CelebA dataset in PSNR, SSIM and FID on center masks.

3. Experimental Validation on Face Inpainting

In this part, we evaluate the face inpainting performance of our LaFIn on the CelebA-HQ face dataset [17, 12]. The masks used for training come from the random mask dataset [16] and additional block masks randomly generated. The competitors involved in the comparison include Context Encoder (CE) [21], Generative Face Completion (GFC) [15], Contextual Attention (CA) [34], Geometry Aware Face Completion (GAFC) [25], Pluralistic Image Completion (PIC) [37], and EdgeConnect (EC) [20]. For quantitatively measuring the performance difference among the competitors, we employ PSNR, SSIM [27] and FID [7], as metrics. For PSNR and SSIM, higher values indicate better performance, while for FID, the lower the better. As the ground-truth landmarks are unavailable for the CelebA-HQ dataset, we apply the results by FAN [3] to perform as the reference information for training our landmark predictor.

Result comparison. Table 1 reports the performance of CA, EC, PIC and our LaFIn with different types and sizes of mask. Notice that for CA and PIC, the pre-trained mod-



(a) Ground truth (GT) (b) Masked GT (c) CA (d) EC (e) PIC (f) Ours

Figure 4: Qualitative comparison with other state-of-the-art techniques on the CelebA-HQ dataset. (a) shows the ground-truth images. (b) depicts the masked versions of (a). (c)-(f) are the results obtained by CA, EC, PIC, and our LaFIn, respectively.

(a) Ground truth (GT) (b) Masked GT (c) CE (d) GFC (e) EC (f) Ours

Figure 5: Visual comparison between the competitors on the CelebA dataset. (a) shows the ground-truth images. (b) depicts the masked versions of (a). (c)-(f) are the results obtained by CE, GFC, EC, and our LaFIn, respectively.

	Mask	w/o LSTA	w/o LMK	Ours
PSNR	10-20%	31.02	31.10	**31.48**
	40-50%	24.00	23.75	**24.22**
	Center	25.75	24.90	**25.92**
SSIM	10-20%	0.973	0.972	**0.9754**
	40-50%	0.879	0.869	**0.883**
	Center	0.902	0.879	**0.905**
FID	10-20%	2.18	2.26	**2.05**
	40-50%	8.25	7.72	**7.12**
	Center	8.56	7.21	**6.63**

Table 3: Ablation study on different configurations of LaFIn.

els on the CelebA-HQ are given*†. While the authors of EC do not offer the pre-trained model on the CelebA-HQ dataset, we try our best to retrain it using the training code‡. As can be seen from the numbers in Table 1, EC is superior over PIC and CA in most cases, as it employs the edge information to help inpainting. Overall, our LaFIn outperforms the others by large margins in terms of all PSNR, SSIM and FID, except for the case of center falling behind PIC in terms of FID 6.63 *vs.* 4.98, the explanation is in Appendix. This comparison verifies that the landmarks are stronger and more robust guidance than the edges for the task of face inpainting. Further quantitative comparisons with CE, EC and GFC under center masks on CelebA are shown in Table 2. Figure 4 depicts two visual comparisons

*https://github.com/JiahuiYu/generative_inpainting

†https://github.com/lyndonzheng/Pluralistic-Inpainting

‡https://github.com/knazeri/edge-connect

(a) Ground truth (GT) (b) Masked GT (c) CE (d) GFC (e) GAFC (f) Ours

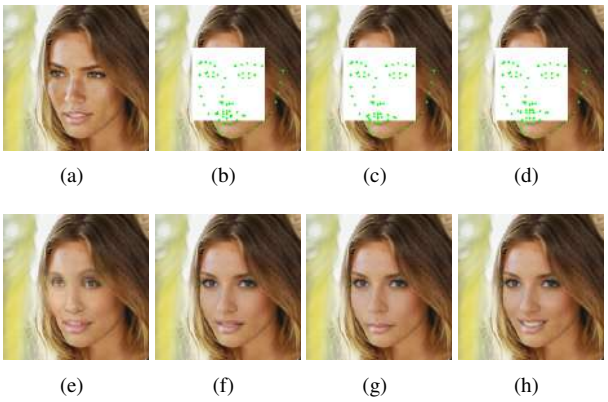
Figure 6: Visual comparison between the competitors on the CelebA dataset. (a) shows the ground-truth images. (b) depicts the masked versions of (a). (c)-(f) are the results obtained by CE, GFC, GAFC, and our LaFIn, respectively.

(a) (b) (c) (d)

Figure 7: More results by LaFIn. (a) Ground truth images. (b) Masked images. (c) Predicted landmarks. (d) Results.

among CA, EC, PIC, and our LaFIn, from which, we can see that LaFIn can generate more natural-looking and visually striking results even on the cases with large poses and extreme occlusions. Figure 5 and 6 further provide visual comparisons of CE, GFC, GAFC, EC and LaFIn on four samples from the CelebA dataset. Notice that GFC utilizes the face parsing information and GAFC uses both the landmark and parsing to guide the inpainting. As observed from the results, those by GFC suffer from the face component shifting problem. GAFC[§] seems to somewhat

[§]Since neither the code nor implementation details of GAFC is available, when this paper is prepared, we only compare the cases cropped from the GAFC paper.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 8: (a) Ground truth image. (b),(c) and (d) present three versions of landmark on the masked image. (f), (g) and (h) give the results conditioned on (b), (c), and (d), respectively. (e) the result without landmark information.

mitigate the problem due to the introduction of landmarks, but still inferior to our LaFIn. This comparison tells that the redundancy of the face parsing prior may alternatively hurt the performance. It is worth to emphasize that GAFC considers the symmetry property of faces and low rankness of mask in the loss, which are not so reasonable because large poses of faces and random corruptions can easily violate these properties. Also, the editing on parsed regions (+landmarks) is much more difficult than on sparse landmarks. Figure 7 gives several more results by LaFIn. Due to page limit, please find more comparisons in Appendix.

Ablation study. Table 3 reports the difference of LaFIn with the long short term attention (LSTA) disabled (denoted as w/o LSTA), LaFIn with the landmark guidance canceled (w/o LMK) , and the complete LaFIn. From the numbers reported in Table 3, both the LSTA and LMK help the task

Figure 9: From left to right: the original image, masked image with ground-truth landmarks, augmented result, and augmented image with ground-truth landmarks.

of face inpainting. Specifically, the LSTA influences more than the landmark indication on the cases with relatively small corruptions. This phenomenon is reasonable because the completed part should pay more attention on the attribute consistency to make the results visually coincident to the observed (large) area. While for the cases with relatively large masks, the attribute consistency is barely violated in the generated result as there is few information given to match. Alternatively the landmark information is more important to ensure the structure well-preserved. The above corroborates the principle of our design, say the LSTA is for the attribute consistency and the landmarks for the main structure. To view the effect of landmark, Figure 8 shows the inpainting results based on different landmark templates. By varying the templates (mouth), the completed faces accordingly change with much better visual quality than the one without adopting any landmark information. This experiment also informs us that editing faces is viable by manipulating the landmark template. Affirmatively, operating sparse landmarks is more convenient than modifying parsed regions (together with landmarks [25]).

4. Further Finding on Data Augmentation

Most of data-driven approaches, if not all, require well-labeled data, which is time consuming and labor intensive. Like the original motivation of GANs, it attempts to produce more samples for training networks. Specifically for facial landmark detectors/predictors, one wants to generate diverse plausible faces given the ground-truth landmarks. Intrinsically, this is how our work stands. For an image I, we are able to obtain the augmented data I_{aug} through $I_{aug} := \mathcal{G}_P(L_{gt}, M \circ I)$, where L_{gt} is the landmark of I, and M stands for any mask. By doing so, for the image I, the augmented faces vary with different masks. The discriminator will make sure that the inpainted results match L_{gt}. An example is shown in Fig. 9, from which we can see that the features of I_{aug} are significantly different from those of I with the same landmarks. Consequently, the pair of (I_{aug}, L_{gt}) can be used for training.

To validate the effectiveness of such a data-augmentation manner, we feed the augmented data into both our $\mathcal{G}_L$ and LAB [28] on the WFLW dataset [28]. We notice that LAB[¶] is carefully built for the task of facial landmark detection, while the landmark module in LaFIn is much simpler and smaller because as previously explained, in our task, the landmarks can be not that accurate as long as they can provide the main structure of faces. Therefore, our performance in NME (normalized mean error by inter-ocular factor) is inferior to LAB. Nevertheless, as can be viewed from Table 4, the augmentation improves both the performance of LAB and our LaFIn. In addition, we also test LaFIn on the 300W dataset, the numerical results consistently reveal the effectiveness of the augmentation. Notice that no obvious difference in inpainted results is observed using the landmark predictors without and with augmentation, which again verifies that our inpainting module is robust against variation in landmarks, and can produce striking results as long as the structure is reasonably offered.

LAB	LAB_{aug}	LaFIn	LaFIn_{aug}
5.66	5.43	6.79	5.92

Table 4: Comparison in NME on the WFLW dataset.

Dataset	Common	Challenging	Full
LaFIn	4.69	8.95	5.42
LaFIn_{aug}	4.45	8.91	5.21

Table 5: Comparison in NME on the 300W dataset.

5. Conclusion

In this study, we have developed a generative network, namely LaFIn, for completing face images. The proposed LaFIn first predicts the landmarks then accomplishes the inpainting conditioned on the predicted landmarks. Our principle is that the landmarks are neat, sufficient, and robust to perform as guidance for providing the structural information to the face inpainting module. For ensuring the attribute consistency, we designed to harness distant spatial context and connect temporal feature maps. Extensive experiments have been conducted to verify our claims, reveal the efficacy of our design and, demonstrate its advances over state-of-the-art alternatives both qualitatively and quantitatively. Furthermore, we proposed to use our LaFIn to augment face-landmark data for relieving manual annotation in the task of landmark detection. The effectiveness of this manner has been experimentally confirmed.

[¶]We use the PyTorch version of LAB downloaded from https://github.com/FunkyKoki/Look_At_Boundary_PyTorch

Appendices

A. Network Architecture

A.1. The Landmark Predictor

Our landmark predictor is based on the MobileNet-V2. A series of bottlenecks are employed to extract the features and speed up the network. Feature maps at different stages of fusion layers are fully connected to achieve the final landmark prediction. The detailed architecture is shown in Table 6.

Input	Operator	t	c	n	s
$256^2 \times 3$	conv2d	-	32	1	2
$128^2 \times 32$	bottleneck	1	16	1	1
$128^2 \times 16$	bottleneck	6	24	2	2
$64^2 \times 24$	bottleneck	6	32	3	2
$32^2 \times 32$	bottleneck	6	64	4	2
$16^2 \times 64$	bottleneck	6	96	3	1
$16^2 \times 96$	bottleneck	6	160	3	2
$8^2 \times 160$	bottleneck	6	320	1	1
(C1) $8^2 \times 320$	conv2d 1x1	-	1280	1	1
(C2) $8^2 \times 1280$	avgpool 8x8	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	64	-	-
(S3) $1 \times 1 \times 64$	-	-	64	-	-
C1	conv2d 1x1	-	128	1	1
$8^2 \times 128$	avgpool 8x8	-	128	1	-
(S1) $1 \times 1 \times 128$	-	-	128	1	-
C2	conv2d 1x1	-	128	1	1
$8^2 \times 128$	avgpool 8x8	-	128	1	-
(S2) $1 \times 1 \times 128$	-	-	128	1	-
S1,S2,S3	Full Connection	-	136	1	-

Table 6: The network architecture of our landmark predictor. Each line represents a sequence of identical layers, repeating n times. For layers in the same sequence, they have the same number c of output channels. The first convolution layer of each sequence has a stride s. The expansion factor is applied to the input size in bottleneck layers.

A.2. Discriminator of the Inpaintor

The discriminator is built upon the Patch-GAN architecture. Spectral normalization is applied on the convolution layers to stabilize the training process. The attention block is placed in the discriminator to adaptively treat the features. The detailed architecture is given in Table 7.

A.3. Generator of the Inpaintor

The generator is based on the U-Net structure. Three encoding blocks are applied for down-sampling, followed by 7 residual blocks with dilated convolutions to enlarge receptive fields. The long-short term attention block connects the features from the last residual block and the last down-sampling block so that the features in a wider range can be better used. The shortcuts are added between corresponding encoder and decoders. The 1x1 convolutions are employed as channel attention to adjust the weights of features from shortcut and last layer. The detailed architecture is shown in Table 8.

Input	Operator	k	c	s	p
$256^2 \times 4$	Conv-SN-LReLU	4	64	2	1
$128^2 \times 64$	Conv-SN-LReLU	4	128	2	1
$128^2 \times 128$	Attention	-	128	-	-
$64^2 \times 128$	Conv-SN-LReLU	4	256	2	1
$32^2 \times 256$	Conv-SN-LReLU	4	512	1	1
$31^2 \times 512$	Conv-SN-Sigmoid	4	1	1	1
$30^2 \times 1$	-	-	1	-	-

Table 7: The discriminator network architecture. Each line represents a sequence of listed layers or a hole block. The k, c, s, p represent kernel size, output channels, stride and padding of convolution or deconvolution layers, respectively. SN refers to spectral normalization and LReLU means leaky relu with the slope set to 0.2.

B. Implementation Details on Data Augmentation

In the last but one section of the main paper, we validated the effectiveness of the proposed data-augmentation manner. The implementation details are as follows. In our experiment, for each pair of training sample(I_{gt}, L_{gt}) in a single epoch, a pair of augmented data (I_{aug}, L_{gt}) will be generated by the inpaintor and be applied as the additional training data. Moreover, in different epochs, the masked region will change so that various augmented images can be produced. The training settings of LaFIn is same as above mentioned except the batch size shrinks to 4. The settings of LAB[28] follow its original implementation.

C. Further Analysis on Experimental Results

In Table 1 of the main paper, our LaFIn falls behind PIC in terms of FID 6.63 *vs.* 4.98 in the case of center mask. First we give the definition of FID as follows:

$$\mathrm{FID}(x, g) = \|\mu_x - \mu_g\|_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x \Sigma_g)^{\frac{1}{2}}). \quad (8)$$

Assuming that the extracted features x and g follow multidimensional Gaussian distributions $(\mu_x, \mathbf{\Sigma}_x)$ and $(\mu_g, \mathbf{\Sigma}_g)$ respectively, the FID calculates the Frechet distance between

Input	Operator	k	c	s	p	Out
$256^2 \times 4$	Conv-IN-ReLU	7	64	1	3	E1
$256^2 \times 64$	Conv-IN-ReLU	4	128	2	1	E2
$128^2 \times 128$	Conv-IN-ReLU	4	256	2	1	E3
$64^2 \times 256$	Dilated Residual Block	-	256	-	-	-
$64^2 \times 256$	Dilated Residual Block	-	256	-	-	-
$64^2 \times 256$	Dilated Residual Block	-	256	-	-	-
$64^2 \times 256$	Dilated Residual Block	-	256	-	-	-
$64^2 \times 256$	Dilated Residual Block	-	256	-	-	-
$64^2 \times 256$	Dilated Residual Block	-	256	-	-	-
$64^2 \times 256$	Dilated Residual Block	-	256	-	-	R7
E3,R7 $64^2 \times 512$	Short+Long Term Attention	-	256	-	-	
$64^2 \times 256$	Deconv-IN-ReLU	4	128	2	1	D1
E2,D1 $128^2 \times 256$	Conv-IN-ReLU	1	256	1	0	-
$128^2 \times 256$	Deconv-IN-ReLU	4	64	2	1	D2
E1,D2 $256^2 \times 128$	Conv-IN-ReLU	1	128	1	0	-
$256^2 \times 128$	Conv-IN-tanh	7	3	1	3	-
$256^2 \times 3$	-	-	3	-	-	-

Table 8: The generator network architecture. Each line represents a sequence of listed layers or a hole block. The k, c, s, p represent kernel size, output channels, stride and padding of convolution or deconvolution layers, respectively. Reflection paddings are applied in the first convolution layer and last deconvolution layer while others apply zero-padding. The layers with skip connections are showed at first column. Other layers directly take output of previous layers as input. IN represents instance normalization.

the two distributions. And Tr stands for the trace of matrix. From Eq.(8) we can see that the FID takes both the mean and the variance of features into consideration. As Figures 11 and 12 show, in the situation of center mask, the available information in images for inpainting is limited and our LaFIn tends to generate common but reasonable results, which decreases the performance in terms of FID, especially in the variance term. While PIC is designed to generate pluralistic features, but some of the results are not visually satisfactory. More results comparing with CA [34], EC[20], PIC[37] on CelebA-HQ and CE[21], GFC [15], GAFC [25] on CelebA are shown in Figure 10 to Figure 14.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28(3):24:1–24:11, 2009. 1, 2

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 417–424, 2000. 2

[3] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *ICCV*, pages 1021–1030, 2017. 5

[4] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038, 1999. 2

[5] S. Esedoglu and J. Shen. Digital inpainting based on the mumford–shah–euler image model. *European Journal of Applied Mathematics*, 13(4):353–370, 2002. 2

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 2

[7] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, pages 6626–6637, 2017. 5

[8] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf. Image completion using planar structure guidance. *ACM Transactions on graphics (TOG)*, 33(4):129:1–129:10, 2014. 2

[9] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):107, 2017. 2, 4

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 4

[11] Y. Jo and J. Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. *arXiv preprint arXiv:1902.06838*, 2019. 2

[12] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 5

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[14] A. Kumar and R. Chellappa. Disentangling 3d pose in a dendritic cnn for unconstrained 2d face alignment. In *CVPR*, pages 430–439, 2018. 4

[15] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. In *CVPR*, pages 3911–3919, 2017. 2, 3, 5, 10

[16] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, pages 85–100, 2018. 2, 5

[17] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 5

[18] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pages 2794–2802, 2017. 5

[19] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 4

[20] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019. 2, 3, 5, 10

[21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 5, 10

[22] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 2

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 4

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 4

[25] L. Song, J. Cao, L. Song, Y. Hu, and R. He. Geometry-aware face completion and editing. In *AAAI*, pages 2506–2513, 2019. 5, 8, 10

[26] Q. Sun, L. Ma, S. Joon Oh, L. Van Gool, B. Schiele, and M. Fritz. Natural and effective obfuscation by head inpainting. In *CVPR*, pages 5050–5059, 2018. 2

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5

[28] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, pages 2129–2138, 2018. 4, 8, 9

[29] S. Xiao, J. Feng, L. Liu, X. Nie, W. Wang, S. Yang, and A. Kassim. Recurrent 3d-2d dual learning for large-pose facial landmark detection. In *CVPR*, pages 1633–1642, 2017. 4

[30] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *NeurIPS*, pages 341–349, 2012. 1

[31] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo. Foreground-aware image inpainting. In *CVPR*, pages 5840–5848, 2019. 2

[32] H. Yamauchi, J. Haber, and H.-P. Seidel. Image restoration using multiresolution texture synthesis and image inpainting. In *Computer Graphics International*, pages 120–125. IEEE, 2003. 2

[33] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2

[34] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *CVPR*, pages 5505–5514, 2018. 2, 5, 10

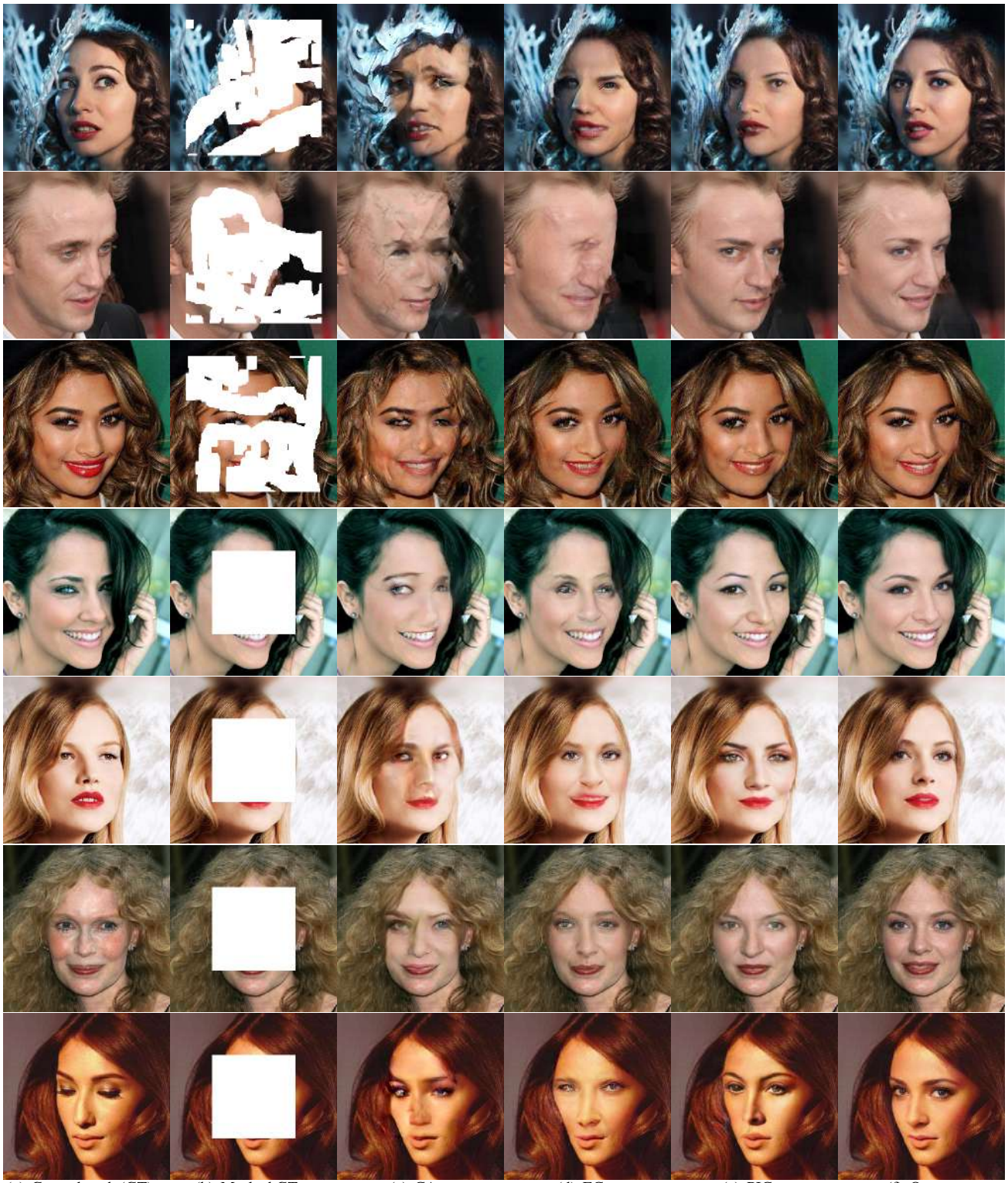
[35] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky. Few-shot adversarial learning of realistic neural talking head models. *arXiv preprint arXiv:1905.08233*, 2019. 2

[36] J. Zhang, X. Zeng, Y. Pan, Y. Liu, Y. Ding, and C. Fan. Faceswapnet: Landmark guided many-to-many face reenactment. *arXiv preprint arXiv:1905.11805*, 2019. 2

[37] C. Zheng, T.-J. Cham, and J. Cai. Pluralistic image completion. In *CVPR*, pages 1438–1447, 2019. 4, 5, 10

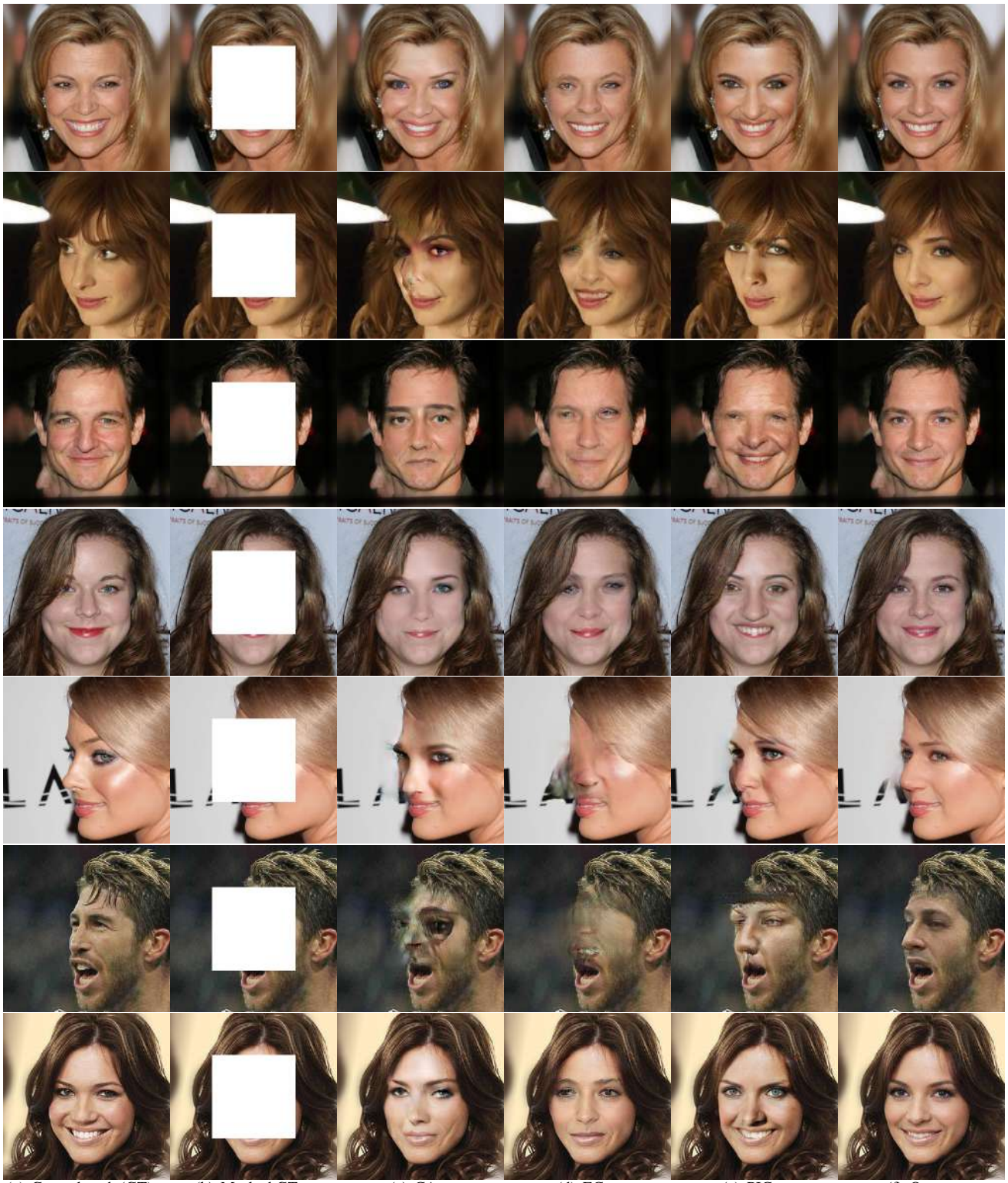
(a) Ground truth (GT) (b) Masked GT (c) CA (d) EC (e) PIC (f) Ours

Figure 10: More results with other state-of-the-art techniques on the CelebA-HQ dataset. (a) shows the ground-truth images. (b) depicts the masked versions of (a). (c)-(f) are the results obtained by CA, EC, PIC, and our LaFIn, respectively.



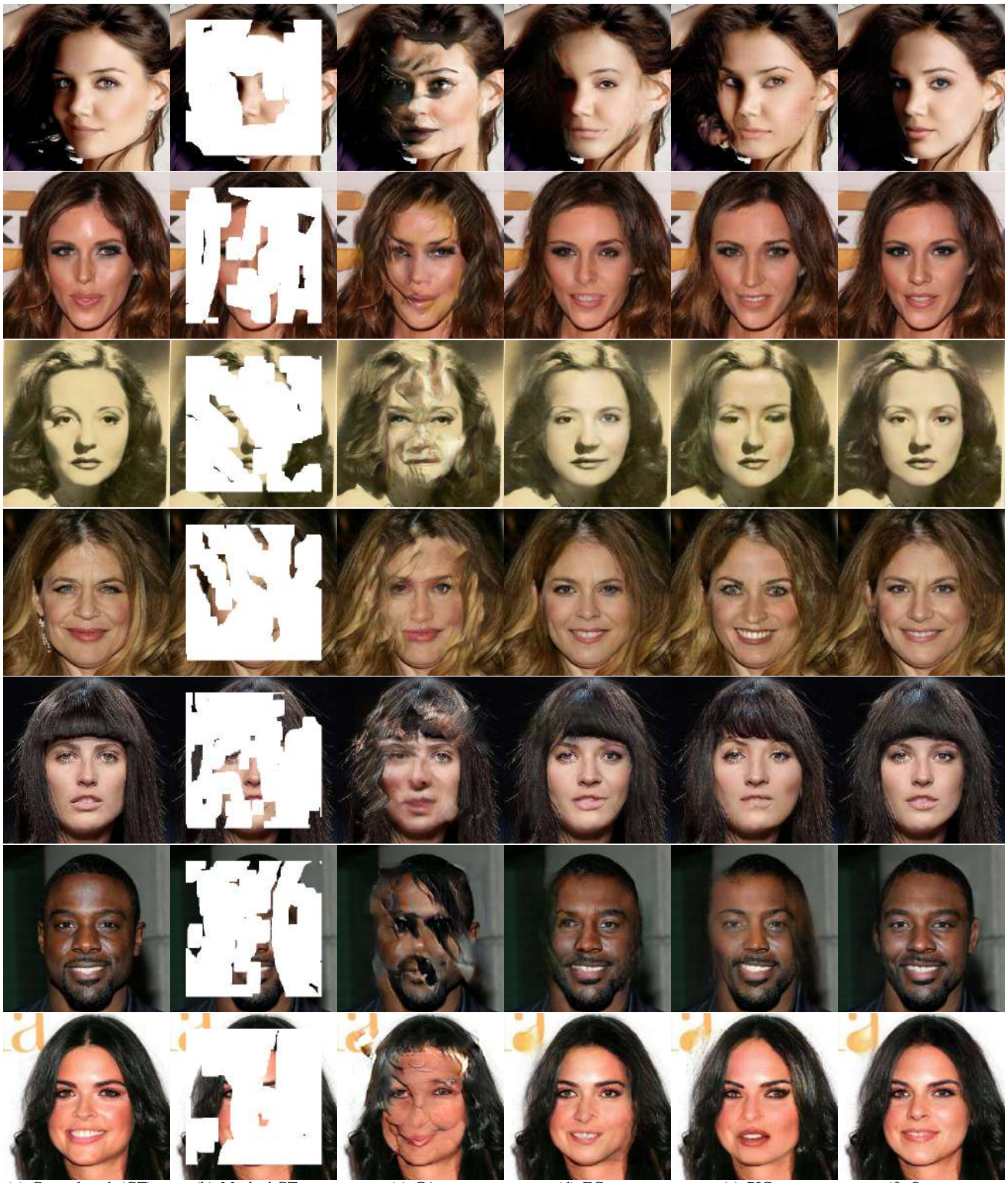
(a) Ground truth (GT) (b) Masked GT (c) CA (d) EC (e) PIC (f) Ours

Figure 11: More results with other state-of-the-art techniques on the CelebA-HQ dataset. (a) shows the ground-truth images. (b) depicts the masked versions of (a). (c)-(f) are the results obtained by CA, EC, PIC, and our LaFIn, respectively.



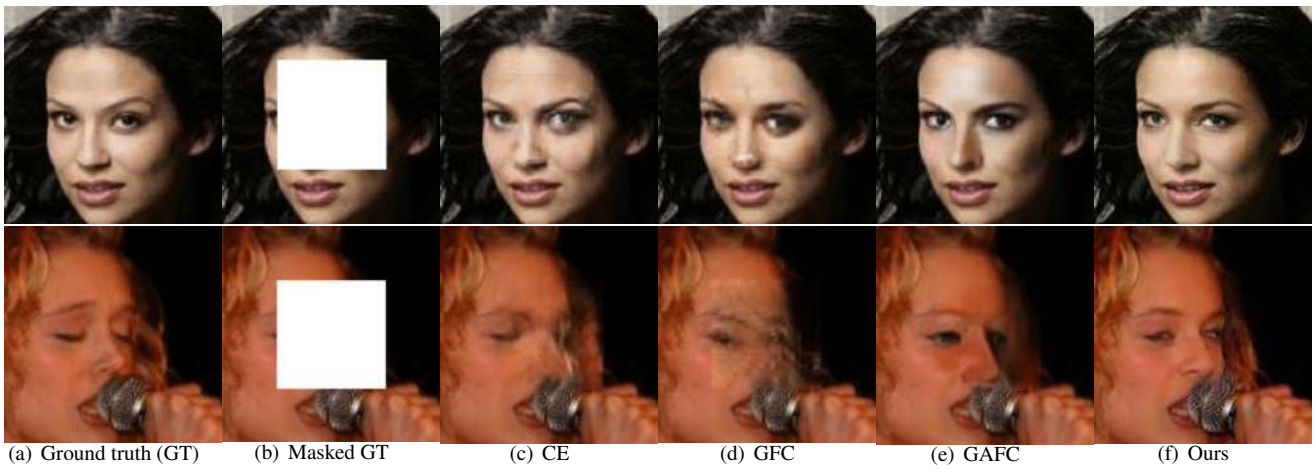
(a) Ground truth (GT) (b) Masked GT (c) CA (d) EC (e) PIC (f) Ours

Figure 12: More results with other state-of-the-art techniques on the CelebA-HQ dataset. (a) shows the ground-truth images. (b) depicts the masked versions of (a). (c)-(f) are the results obtained by CA, EC, PIC, and our LaFIn, respectively.



(a) Ground truth (GT) (b) Masked GT (c) CA (d) EC (e) PIC (f) Ours

Figure 13: More results with other state-of-the-art techniques on the CelebA-HQ dataset. (a) shows the ground-truth images. (b) depicts the masked versions of (a). (c)-(f) are the results obtained by CA, EC, PIC, and our LaFIn, respectively.



(a) Ground truth (GT) (b) Masked GT (c) CE (d) GFC (e) GAFC (f) Ours

Figure 14: Visual comparison between the competitors on the CelebA dataset. (a) shows the ground-truth images. (b) depicts the masked versions of (a). (c)-(f) are the results obtained by CE, GFC, GAFC, and our LaFIn, respectively.