Computational Visual Media
https://doi.org/10.1007/s41095-0xx-xxxx-x

Research Article

GRIG: Data-efficient generative residual image inpainting

Wanglong Lu[1,2], **Xianta Jiang**[2], **Xiaogang Jin**[3], **Yong-Liang Yang**[4], **Minglun Gong**[5], **Kaijie Shi**[1,2], **Tao Wang**[6] **and Hanli Zhao**[1](✉)

Abstract Image inpainting is the task of filling in missing or masked regions of an image with semantically meaningful content. Recent methods have shown significant improvement in dealing with large missing regions. However, these methods usually require large training datasets to achieve satisfactory results, and there has been limited research into training such models on a small number of samples. To address this, we present a novel data-efficient generative residual image inpainting method that produces high-quality inpainting results. The core idea is to use an iterative residual reasoning method that incorporates convolutional neural networks (CNNs) for feature extraction and transformers for global reasoning within generative adversarial networks, along with image-level and patch-level discriminators. We also propose a novel forged-patch adversarial training strategy to create faithful textures and detailed appearances. Extensive evaluation shows that our method outperforms previous methods on the data-efficient image inpainting task, both quantitatively and qualitatively.

Keywords Image inpainting, iterative reasoning, residual learning, generative adversarial networks.

1 Introduction

Image inpainting is a fundamental task in computer graphics and computer vision [1–4]. It has been employed in many downstream applications, such as image restoration [5], and image manipulation [6]. Recently proposed image inpainting methods have achieved impressive results [7, 8] on both realistic and facial images [9]. However, these methods have an overlooked limitation: they require a large amount of data to train their convolutional neural network (CNN) or transformer models [10]. When these models are trained on small image datasets, there is a high possibility of overfitting and model collapse [11]. In practice, it is much easier for users if only a small number of training images is required. Furthermore, in certain image domains (e.g., medical, art, and historical relics), large image datasets are either too expensive or infeasible to collect. This has severely restricted the use of image inpainting in real-world scenarios. Moreover, real-world data often come with their own set of challenges, including privacy concerns, data security, and data quality. These issues can significantly limit the amount of usable data. Improving the data efficiency of model training can be a crucial factor in expediting the adoption and application of image inpainting, thereby increasing its applicability to various fields that face limitations of data availability. Moreover, small-scale training samples demand less processing power and memory, which enhances the feasibility of training models in resource-limited environments.

Achieving high-quality image inpainting on small-scale datasets is still a challenging and open problem. Most existing methods [8, 12, 13] rely on single-pass inferencing which may generate ambiguous results in inpainted sub-regions. Some methods [14, 15] perform inpainting in a progressive fashion by reusing parts of previously inpainted features

1 Key Laboratory of Intelligent Informatics of Safety & Emergency of Zhejiang Province, Wenzhou University, Wenzhou 325035, China. E-mail: W. Lu, wanglongl@mun.ca; K. Shi, kaijies@mun.ca; H. Zhao, hanlizhao@wzu.edu.cn(✉).

2 Department of Computer Science, Memorial University of Newfoundland, St. John's, NL A1B 3X5, Canada. E-mail: X. Jiang, xiantaj@mun.ca.

3 State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058, China. E-mail: jin@cad.zju.edu.cn.

4 Department of Computer Science, University of Bath, Bath, BA2 7AY, United Kingdom. E-mail: y.yang@cs.bath.ac.uk.

5 School of Computer Science University of Guelph Guelph, ON, N1G 2W1, Canada. E-mail: minglun@uoguelph.ca.

6 Department of Computer Science and Technology, Nanjing University, China. E-mail: taowangzj@gmail.com.

Manuscript received: 2023-10-11; accepted: 2024-02-03

from early refinement stages. However, these methods do not fully utilize the inpainted pixels as useful information for the next iteration. Moreover, existing inpainting methods are not designed specifically for data-efficient learning and may not work well with a limited number of training samples. Domain-related prior knowledge [16] or lightweight generative models [17, 18] may be employed in image inpainting to mitigate overfitting. However, such approaches do not perform well when there is a large domain gap between two tasks or the reduced network capacity affects the inpainting quality.

Thus, we propose a novel data-efficient *generative residual image inpainting* framework (GRIG), which enables high-quality image inpainting on small-scale datasets. To effectively optimize inpainting results, we use iterative reasoning to more accurately and generalizably solve algorithmic reasoning tasks [19], based on residual learning [20] to incrementally refine previous estimates. By continually updating residual offsets and utilizing inpainted information from previous iterations, our model dynamically refines the input image. This approach reduces direct memorization of input-to-ground-truth mappings, effectively diminishing overfitting and enhancing visual quality. We also investigate whether combining iterative reasoning and residual learning with CNNs and transformers [10], as well as image-level and patch-level discriminators, can lead to a more robust and data-efficient method to tackle the data-efficient image inpainting task.

We have implemented our framework using three components: a generator, a projected discriminator, and a forged-patch discriminator. The generator uses CNNs to extract shallow features of edges and textures and transformer blocks to capture global interactions between feature contexts at each iterative step. To accelerate network convergence and reduce overfitting, we decouple image distribution learning by using image-level and patch-level discriminators. We first build the projected discriminator to capture the whole image-level distribution. We then use a forged-patch discriminator to enhance the patch-level details of generated images, as the projected discriminator has difficulty in capturing fine details in inpainted images. The inpainting process is carried out in several forward passes by feeding the generator with the output of the previous iteration and the corresponding mask. Experimental results on ten small-scale and four large-scale datasets show that our method is superior to state-of-the-art methods in terms of data-efficient and high-quality image inpainting.

In summary, this paper proposes a novel data-efficient generative residual image inpainting framework with the following contributions:

- A novel algorithm for data-efficient image inpainting, which integrates CNNs and transformers, as well as employing image-level and patch-level discriminators for iterative residual reasoning.
- A forged-patch discriminator that assists the generative network to improve the fine details of generated images and prevent overfitting for data-efficient image inpainting.
- State-of-the-art performance on ten small-scale benchmark datasets with varying contents and characteristics, including facial, photorealistic, animal, medical, cartoon, and artistic images.

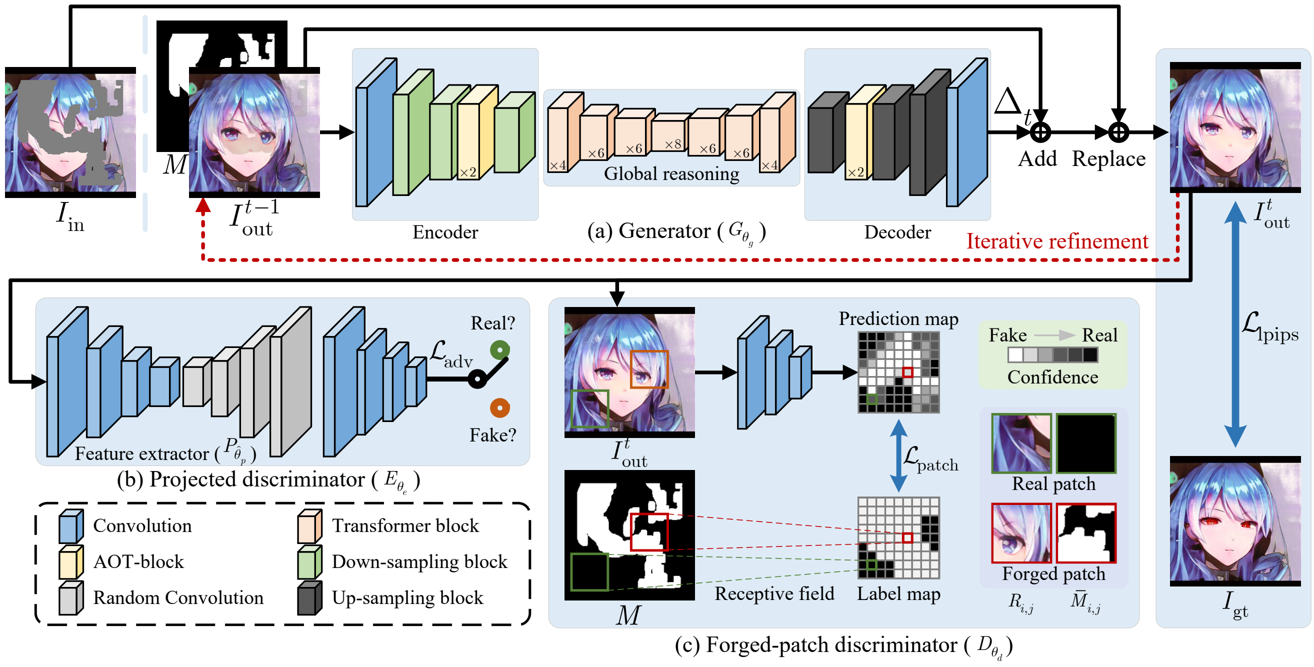
The subsequent sections are organized as follows. Section 2 presents an overview of related work, covering both traditional and deep-learning-based image inpainting methods. In Section 3, we detail our proposed data-efficient generative residual image inpainting method. Section 4 presents benchmarking results and a corresponding analysis of existing image inpainting approaches on ten small-scale and four large-scale datasets. Finally, Section 5 summarizes the findings and conclusions of this study.

2 Related work

Image inpainting can be grouped into traditional image inpainting methods and deep-learning-based inpainting approaches. The former mainly rely on low-level features, while the latter leverage deep neural networks to extract semantic features [21], resulting in better visual quality. However, there has been limited research into training such deep-learning-based models on a small number of samples.

Early image inpainting techniques rely heavily on low-level features from pixels and image patches. Methods based on diffusion [22–24] propagate undamaged information along the boundary to the hole's center. Patch-based methods [25–27] iteratively search for and copy similar appearances from image datasets or known backgrounds. Some variants include GPU-based parallel methods [28], summarizing non-stationary patterns [29], and inpainting with nonlocal texture similarity [30]. Because of the lack of semantic understanding of the image, these methods perform well for small-scale and narrow missing regions but fail to recover meaningful contents for large holes.

Deep-learning-based inpainting methods have achieved great success in semantic completion. Deep neural networks have been used extensively to improve the visual quality of inpainting [31]. These works include an auto-encoder-based architecture [32] and its variants [33–35]. Various



(a) Generator (G_{θ_g})

(b) Projected discriminator (E_{θ_e})

(c) Forged-patch discriminator (D_{θ_d})

Fig. 1 Pipeline of our data-efficient generative residual image inpainting framework (GRIG). In each t-th residual reasoning step, the generator (a) utilizes the $(t-1)$-th inpainted image I_{out}^{t-1} to generate the residual image Δ_t. In the first residual reasoning step ($t = 1$), we set $I_{\text{out}}^{0} = I_{\text{in}}$. With the inpainted image I_{out}^{t-1} and the initial input image I_{in}, add and replace operations (refer to Eq. 1) are performed to obtain I_{out}^{t} for the next iterative refinement. During adversarial training, the inpainted image is fed into the projected discriminator (b) and forged-patch discriminator (c), respectively. At each iterative reasoning step, the loss functions and corresponding back-propagation are re-computed. During the test phase, a similar multi-step prediction is performed without the loss functions and back-propagation. For simplicity, down- and up-sampling operations are omitted.

sophisticated modules or learning strategies have been developed to enhance the effectiveness of image inpainting, including global and local discriminators [36], contextual attention [37–41] to improve semantic understanding, methods for dealing with irregular holes [12, 13, 42], and utilization of auxiliary information (such as sketches [43, 44], foreground contours [45], structures [44, 46], exemplars [47], and prior features [1]). Recent research has addressed issues related to high-resolution [7, 8, 14, 39, 48, 49], pluralistic generation [8, 12, 50–52], and large hole filling [8, 12, 15, 49, 51–56]. The methods discussed above aim for semantically high-quality completion, but they may overfit when trained on data with a small number of samples.

Progressive-based image inpainting methods [14, 15, 53–55] are closely related to our work. These methods primarily inpaint pixels from the hole boundary to the center in a progressive manner [14, 53–55] or employ multi-stage refinement schemes [14, 15]. For example, Zeng et al. [14] improved high-resolution inpainting by iteratively predicting a confidence map and corresponding intermediate results. Such methods reuse only a portion of the predicted information and do not change pixels with high confidence for the next iterative inpainting. Recurrent Feature Reasoning (RFR) [15] runs embedded feature maps through their feature reasoning module multiple times to generate multiple features for adaptive feature merging. RFR's final inpainted results, on the other hand, are produced from their decoder with a single forward pass, indicating that the model cannot readjust its results at the pixel level for better fine details. In this paper, we make a first attempt at image inpainting training on a small number of samples. We show how our framework can refine the results of inpainting through iterative residual reasoning, which combines CNNs and transformers, as well as image-level and patch-level discriminators. Our approach allows for the efficient reuse of all previously predicted pixels and network parameters, leading to an effective model for the data-efficient image inpainting task.

3 Methodology

Iterative reasoning, which involves applying underlying computations to the outputs of previous reasoning steps repeatedly, has the potential to more accurately and generalizably solve algorithmic reasoning tasks [19], whereas residual learning [20, 57] facilitates the progressive optimization of previously predicted results. By iteratively predicting residual offsets and reusing previously predicted information within

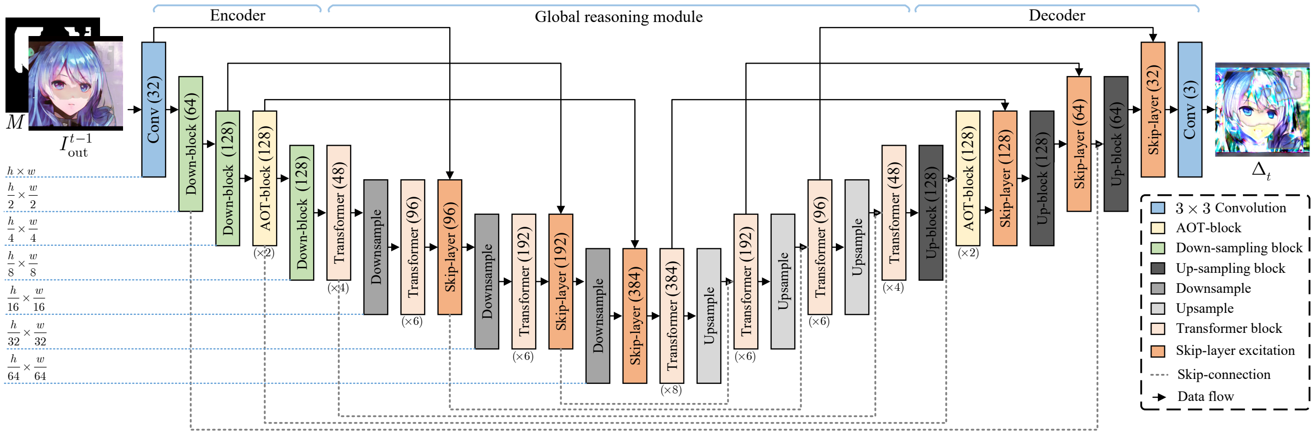


Fig. 2 Network architecture of our generator. We show the output number of channels and dimensions for each layer/block at each scale.

the same generator during each reasoning step, our GRIG can dynamically learn to refine the input image at each step. This method avoids memorizing the mapping between the inputs and their ground-truth images, thereby preventing overfitting and improving visual quality for data-efficient inpainting. In addition, to efficiently learn the global distribution of images, we utilize prior knowledge from pre-trained representations to build a compact classifier as an image-level discriminator for improving data efficiency. While this classifier excels in robust feature-based classification, it may not always capture intricate details. Recognizing this potential problem, we introduce a forged-patch discriminator that is trained to recognize real and inpainted patches based on the receptive field of the discriminator. The synergy of two discriminators mitigates the risk of overlooking fine details while also avoiding overfitting, a common challenge in data-efficient training.

As Fig. 1 shows, our framework consists of three main parts: a generator, a projected discriminator, and a forged-patch discriminator. Given a ground-truth image $I_{\text{gt}} \in \mathbb{R}^{h \times w \times 3}$ and a binary mask $M \in \mathbb{R}^{h \times w \times 1}$ (with 1 for unknown and 0 for known pixels), the masked image $I_{\text{in}} \in \mathbb{R}^{h \times w \times 3}$ is obtained as $I_{\text{in}} = I_{\text{gt}} \odot (1 - M)$, where $\odot$ denotes the Hadamard product. The goal of GRIG is to automatically inpaint a realistic image $I_{\text{out}}^{T} \in \mathbb{R}^{h \times w \times 3}$ with T steps of iterative reasoning, where $T > 1$ denotes the iterative reasoning steps during training. For each t-th iterative residual reasoning step, a previously inpainted image I_{out}^{t-1} is fed into the generator to obtain a residual prediction. At the first residual reasoning step ($t = 1$), we set $I_{\text{out}}^{0} = I_{\text{in}}$. Then, addition and replacement operations are performed to produce a new image completion I_{out}^{t}. Adversarial training is conducted at each iterative step with the network weights updated accordingly via back-propagation.

3.1 Network architectures

3.1.1 *Generator*

Taking the previous iteration's inpainted results as input, the generator is designed to combine CNNs and transformers [10, 58, 59] for efficient iterative residual reasoning in data-efficient image inpainting. The generator G_{θ_g} consists of an encoder, a global reasoning module with a stack of Restormer's Transformer blocks [60], and a decoder (see Fig. 1a). The CNN-based encoder and decoder excel at feature extraction, whereas the transformer blocks excel at dynamic attention, global context integration, and generalization. This combination helps the generator generalize effectively on small-scale training samples.

Details of our generator network are shown in Fig. 2. To extract features and enlarge the receptive field for capturing both informative distant image contexts and rich patterns of interest, we first stack a convolution layer, several residual down-sampling blocks [17], and AOT-blocks [48]. The extracted features are then fed into a Restormer's Transformer block stack for global context reasoning. Meanwhile, skip-layer excitation modules (SLE) [17] are utilized for a shortcut gradient flow, and skip connections are employed for collecting the multi-resolution feature maps in the decoder. The decoder is then built using up-sampling blocks [17], AOT-blocks [48], and a convolution layer. The decoder generates the intermediate prediction Δ_t by utilizing the multi-resolution feature maps output by the encoder and global reasoning module. For stable adversarial training, we apply spectral normalization [61] to all convolution layers of the networks.

At each t-th iterative reasoning step, the inpainted image from the previous iteration I_{out}^{t-1} and its corresponding mask M (see Fig. 1a) are fed into a generative network G_{θ_g} with the learnable network parameters θ_g. G_{θ_g} generates the intermediate residual inpainting $\Delta_t = G_{\theta_g}(I_{\text{out}}^{t-1}, M) \in \mathbb{R}^{h\times w\times 3}$. Then, the t-th inpainted image I_{out}^{t} is calculated as:

$$I_{\text{out}}^{t} = (I_{\text{out}}^{t-1} + \Delta_t) \odot M + I_{\text{in}} \odot (1 - M). \tag{1}$$

3.1.2 Projected discriminator

To stabilize GAN training and improve data efficiency, we use prior knowledge from pre-trained representations to train a compact classifier for learning the global distribution of small-scale images. The projected discriminator (see Fig. 1b) learns to assign high confidence scores to feature maps extracted from real images while assigning low scores to synthetic ones. Initially, feature maps are extracted from the input image I (i.e., I_{out}^{t} or I_{gt}) using a U-net-like projector $P_{\hat{\theta}_p}$ with the pre-trained network parameters $\hat{\theta}_p$. $P_{\hat{\theta}_p}$ is implemented by a pre-trained EfficientNet-Lite1 [62] with cross-channel mixing and cross-scale mixing mechanisms [18]. Subsequently, the projected discriminator E_{θ_e} with learnable network parameters θ_e maps the extracted feature maps to a scalar. Here we selected the discriminator with the largest scale of feature projections (i.e., removing the other three small-scale discriminators) from Projected GAN [18].

3.1.3 Forged-patch discriminator

Because the projected discriminator is primarily focused on extracting global image features for robust classification, it is possible that some fine detail features may be overlooked in these projected features. To help the generator produce faithful fine-grained textures and avoid overfitting in data-efficient training, we propose a forged-patch discriminator that learns to identify real and inpainted patches based on the receptive field [63] of the discriminator.

As shown in Fig. 1c, the forged-patch discrimination network D_{θ_d} with learnable network parameters θ_d learns to recognize real or forged image patches from a given image I (i.e., I_{out}^{t} or I_{gt}). The discriminator D_{θ_d} maps I to a prediction map, where each unit indicates a confidence score for each image patch based on the receptive field. In this work, we adopted the network architecture for D_{θ_d} from PatchGAN [64]. The patch-level receptive field in neural networks has also been studied as a means of overfitting avoidance in interactive video stylization [65] and improving diversity and generalizability in image generation [66].

3.2 Objective functions

GRIG is trained to optimize the learnable network parameters θ_g, θ_e, and θ_d using the objective functions explained below.

3.2.1 LPIPS loss

At each iterative reasoning step, we use the Learned Perceptual Image Patch Similarity (LPIPS) metric [67] to constrain the perceptual similarity between the inpainted image I_{out}^{t} and the ground-truth image I_{gt}:

$$\begin{aligned}&\mathcal{L}_{\text{lpips}}(\theta_g) = \\ &\sum_{l} \frac{1}{H_l W_l} \sum_{u,v} \left\| w_l \odot \left(F_l(I_{\text{out}}^{t})_{u,v} - F_l(I_{\text{gt}})_{u,v}\right)\right\|_2^2,\end{aligned} \tag{2}$$

where H_l and W_l represent the height and width of the feature map for layer l, respectively, u and v are spatial indices in the feature maps, w_l is the weight assigned to the feature map for layer l, $F(\cdot)$ is the pre-trained perceptual feature extractor, $F_l(\cdot)$ is the feature map for layer l; we use VGG-16 in our work [68]. This can assist our generative network in learning to maintain higher visual quality.

3.2.2 Projected adversarial loss

For fast convergence, the projected adversarial loss utilizes pre-trained classification models to extract prior knowledge (see Fig. 1b). We employ the hinge loss [18] to optimize the projected discriminator E_{θ_e} and generative network G_{θ_g}, respectively. The objective function can be formulated as:

$$\begin{aligned}\mathcal{L}_{\text{adv}}^{E}(\theta_e) &= \mathbb{E}_{I_{\text{gt}}}[\text{ReLu}(1 - E_{\theta_e}(P_{\hat{\theta}_p}(I_{\text{gt}})))] \\ &\quad + \mathbb{E}_{I_{\text{out}}^{t}}[\text{ReLu}(1 + E_{\theta_e}(P_{\hat{\theta}_p}(I_{\text{out}}^{t})))], \\ \mathcal{L}_{\text{adv}}^{G}(\theta_g) &= -\mathbb{E}_{I_{\text{out}}^{t}}[E_{\theta_e}(P_{\hat{\theta}_p}(I_{\text{out}}^{t}))].\end{aligned} \tag{3}$$

The projected discriminator is constrained to assign low scores to inpainted images and high scores to real images, while the generator G_{θ_g} is supervised by the projected discriminator to inpaint the masked input based on the distribution of real images.

3.2.3 Adversarial forged-patch loss

As shown in Fig. 1c, we use the forged-patch discriminator to distinguish forged patches from real patches in a given image. We achieve this by constructing the corresponding label map $X \in \mathbb{R}^{h'\times w'}$ to supervise the discriminator. Specifically, we partition I and M into $h' \times w'$ pairs of partially overlapping patches ($R_{i,j}$ and $\overline{M}_{i,j}$) based on the receptive field of forged-patch discriminator D_{θ_d}. Here, $1 \leqslant i \leqslant h'$ and $1 \leqslant j \leqslant w'$ are horizontal and vertical indices, and the sizes of $R_{i,j}$ and $\overline{M}_{i,j}$ are equal to the receptive field $N \times N$. The label map is expressed as follows:

$$X_{i,j} = \begin{cases} 0 & \text{if } \|\overline{M}_{i,j}\|_0 = 0; \\ 1 & \text{otherwise,} \end{cases} \tag{4}$$

where $\|\overline{M}_{i,j}\|_0$ is defined as the L0 norm of the sub-region mask $\overline{M}_{i,j}$. If $\|\overline{M}_{i,j}\|_0$ is not zero, it indicates that there are some masked pixels in this sub-region mask, and the

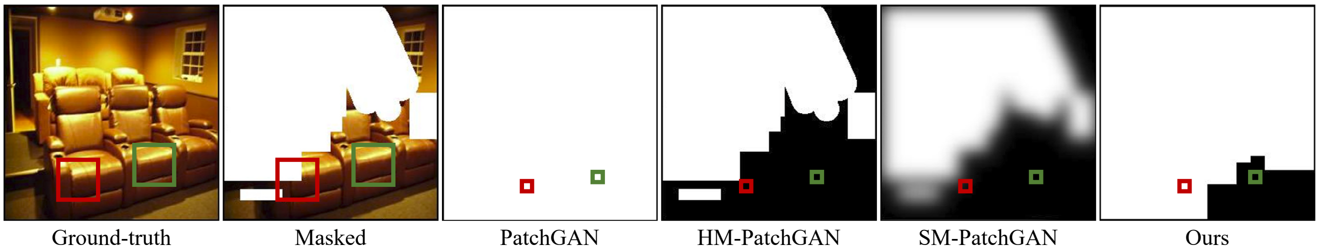

Ground-truth Masked PatchGAN HM-PatchGAN SM-PatchGAN Ours

Fig. 3 Differences between the discriminators of PatchGAN, HM-PatchGAN, SM-PatchGAN, and our algorithm. The boxes represent patches with the size of the discriminator's receptive field (left two images) and corresponding projected positions (right four images) in the resultant label maps over the (red) masked and (green) unmasked regions; Pixel values in the label maps indicate labels for fake (white) and real (black) patches.

image patch $R_{i,j}$ contains inpainted pixels. Thus, we set $X_{i,j} = 1$, which means that the sub-region $R_{i,j}$ is a forged patch. Otherwise, it is a real patch. The hinge version of adversarial forged-patch loss is expressed as:

$$\begin{aligned}\mathcal{L}^{D}_{\text{patch}}(\theta_d) &= \mathbb{E}_{I_{\text{gt}}}[\text{ReLu}(1 - D_{\theta_d}(I_{\text{gt}}))]\\ &\quad + \mathbb{E}_{I^t_{\text{out}}}[\text{ReLu}(1 - D_{\theta_d}(I^t_{\text{out}})) \odot (1 - X)]\\ &\quad + \mathbb{E}_{I^t_{\text{out}}}[\text{ReLu}(1 + D_{\theta_d}(I^t_{\text{out}})) \odot X],\\ \mathcal{L}^{G}_{\text{patch}}(\theta_g) &= -\mathbb{E}_{I^t_{\text{out}}}[D_{\theta_d}(I^t_{\text{out}}) \odot X].\end{aligned} \tag{5}$$

Fig. 3 illustrates the differences between the proposed forged-patch discriminator and other closely related discriminators. PatchGAN's discriminator [64] directly assigns all patches in inpainted images as fake patches, which can confuse the discriminator when extracted patches do not have any generated pixel. HM-PatchGAN and SM-PatchGAN [48] aim to segment synthesized patches of missing regions according to inpainting masks. Since the inpainting masks have to be downsampled first to agree with the spatial size of the discriminator's output, the constraints around the mask boundaries may be unclear. For example, downsampling inpainting masks results in information loss of the precise location of inpainted pixels. SM-PatchGAN tries to identify the generated and real patches, whereas our discriminator goes one step further to consider whether generated pixels are consistent with surrounding real pixels in a given patch. Our discriminator constructs the label map based on the receptive field and treats all patches with any inpainted pixels as fake patches, which gives more constraints than PatchGAN and SM-PatchGAN.

3.2.4 *Total objective*

The total training objective of the generator is expressed as:

$$\mathcal{L}^{G}_{\text{total}} = \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}} + \lambda_{\text{adv}}\mathcal{L}^{G}_{\text{adv}} + \lambda_{\text{patch}}\mathcal{L}^{G}_{\text{patch}}, \tag{6}$$

where λ_{lpips}, λ_{adv}, and λ_{patch} weight corresponding losses. During training, we alternately optimize parameters θ_g, θ_e, and θ_d.

Algorithm 1 GRIG training procedure

```
while $G_{\theta_g}$, $E_{\theta_e}$, and $D_{\theta_d}$ have not converged do
    Sample batch images $\mathcal{I}_{\text{gt}}$ from the training set
    Create random masks $\mathcal{M}$ for $\mathcal{I}_{\text{in}}$
    Get inputs $\mathcal{I}_{\text{in}} \leftarrow \mathcal{I}_{\text{gt}} \odot (1 - \mathcal{M})$
    Set inputs $\mathcal{I}^0_{\text{out}} \leftarrow \mathcal{I}_{\text{in}}$
    for iterative residual reasoning step $t = 1$ to $T$ do
        Get $\Delta_t \leftarrow G_{\theta_g}(\mathcal{I}^{t-1}_{\text{out}}, \mathcal{M})$
        Get $\mathcal{I}^t_{\text{out}} \leftarrow (I^{t-1}_{\text{out}} + \Delta_t) \odot \mathcal{M} + \mathcal{I}_{\text{in}} \odot (1 - \mathcal{M})$
        Update $G_{\theta_g}$ with $\mathcal{L}^G_{\text{total}}$
        Update $E_{\theta_e}$ with $\mathcal{L}^E_{\text{adv}}$
        Update $D_{\theta_d}$ with $\mathcal{L}^D_{\text{patch}}$
    end for
end while
```

3.3 Iterative residual reasoning

The iterative residual reasoning for image inpainting can be formulated as an optimization process over adversarial generative networks. This enables the generator to implicitly learn to leverage previously predicted results and focus on residual information in order to achieve high quality and better generality.

We introduce a generative network $G_{\theta_g}(I^{t-1}_{\text{out}})$ as an explicit function to predict residual information (see Eq. 1). At each iterative reasoning step t, the generator G_{θ_g} is trained to maximize the confidence values of $E_{\theta_e}(I^t_{\text{out}})$ and values in the prediction map of $D_{\theta_d}(I^t_{\text{out}})$ while minimizing the perceptual similarity difference between I^t_{out} and I_{gt}. Thus, θ_g is solved by $\theta_g = \arg\min_{\theta_g} \mathcal{L}^G_{\text{total}}$. Simultaneously, E_{θ_e} and D_{θ_d} are trained to distinguish images (fake or real) and patches, respectively, where $\theta_e = \arg\min_{\theta_e} \mathcal{L}^E_{\text{adv}}$ and $\theta_d = \arg\min_{\theta_d} \mathcal{L}^D_{\text{patch}}$. The generative network G_{θ_g} directly predicts the residual information while its parameters θ_g are supervised by the discriminators as well as I_{gt}. Pseudocode for the GRIG training procedure is given in Algorithm 1.

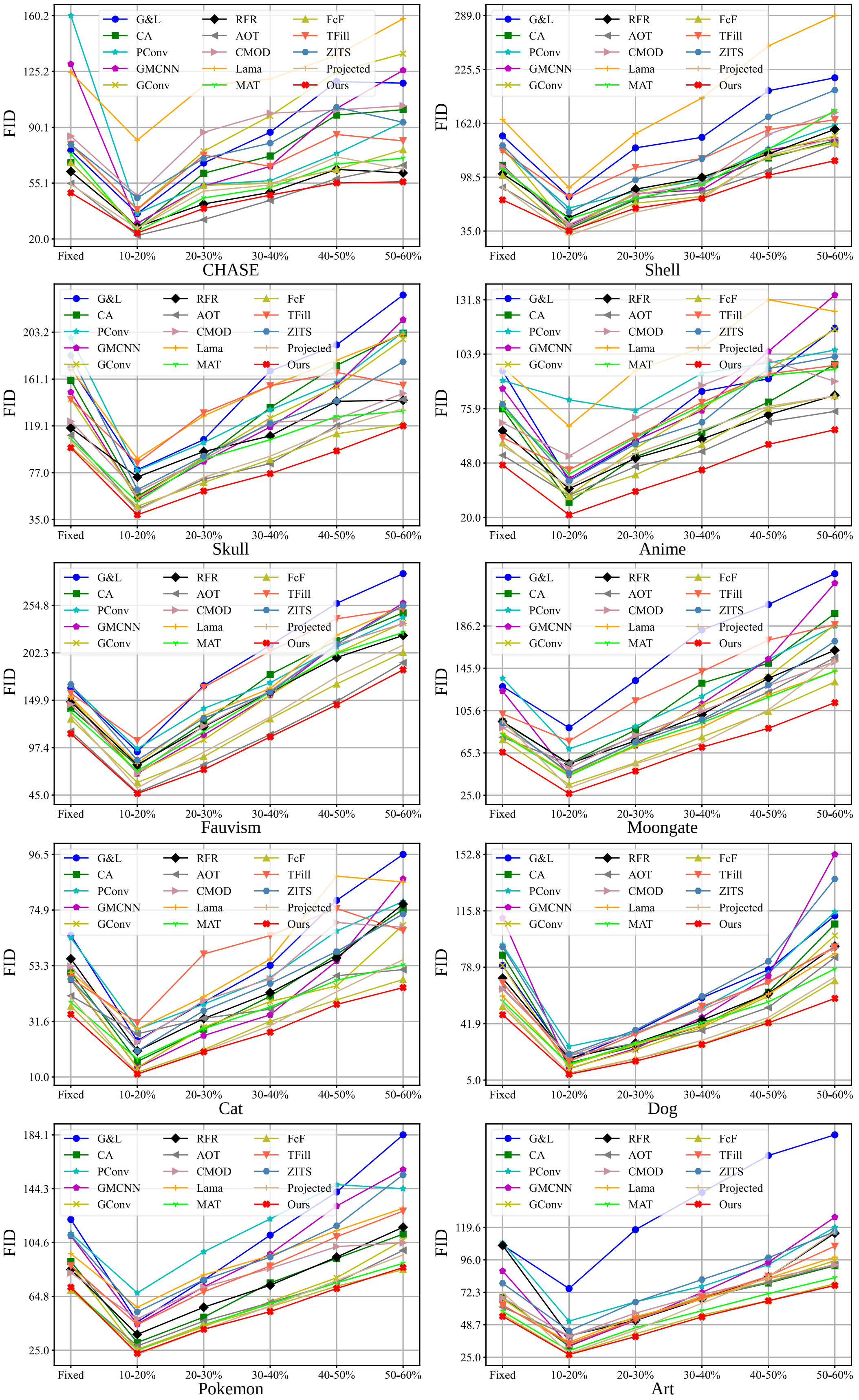


Fig. 4 Quantitative comparisons of GRIG with various state of the art image inpainting methods on ten small-scale datasets, using FID evaluation metric. In each graph, the horizontal axis indicates mask ratios; ‘Fixed’ denotes the fixed center 25% rectangular mask.

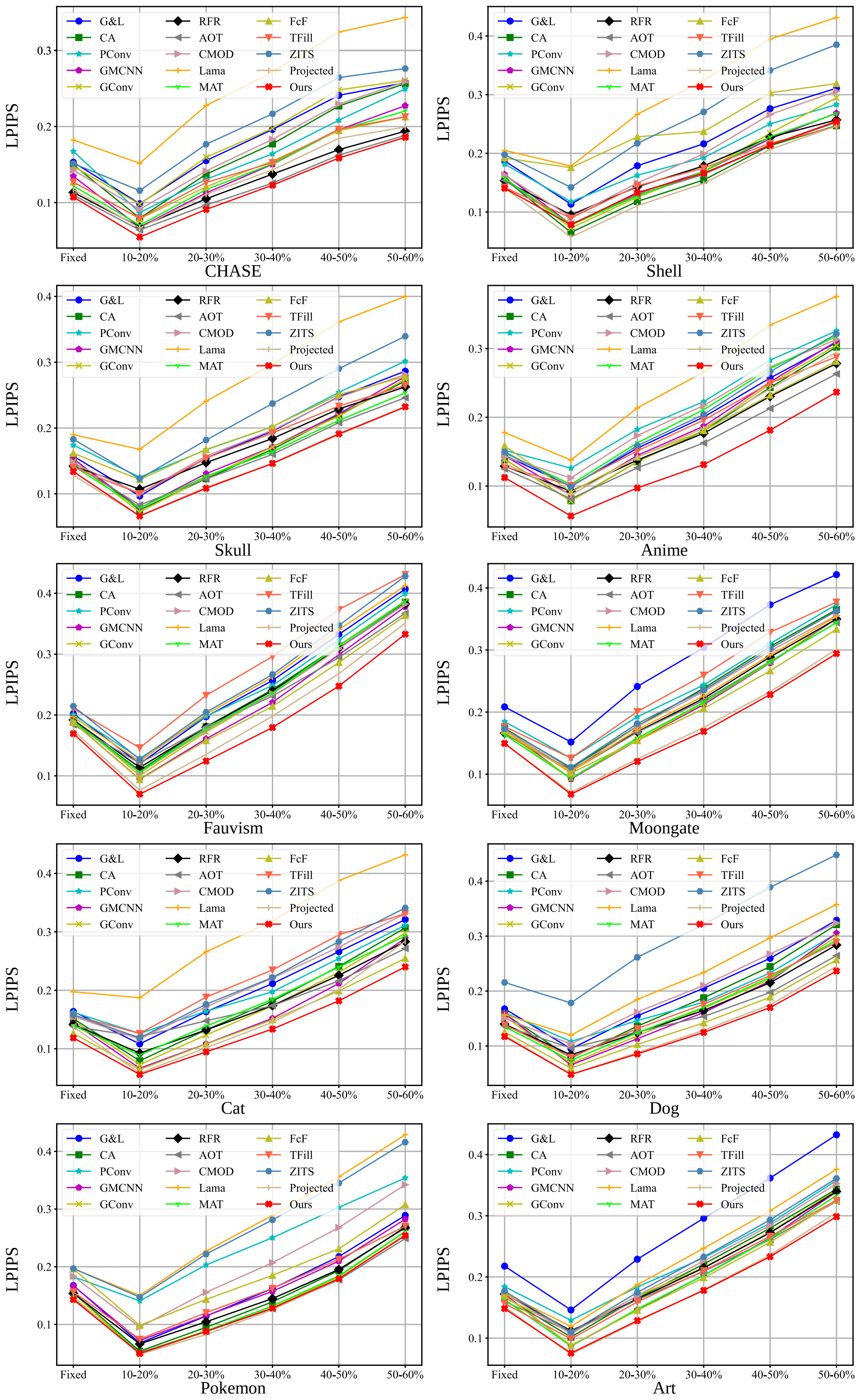


Fig. 5 Quantitative comparisons of GRIG with various state of the art image inpainting methods on ten small-scale datasets, using LPIPS evaluation metric. In each graph, the horizontal axis indicates mask ratios; ‘Fixed’ denotes the fixed center 25% rectangular mask.

Table 1 Details of the ten small-scale and four large-scale image datasets.

Type	Dataset	# Training set	# Test set
Small-scale	CHASE	18	10
	Shell	48	16
	Skull	72	25
	Anime	90	30
	Fauvism	94	30
	Moongate	106	30
	Cat	120	40
	Dog	309	80
	Pokemon	633	200
	Art	750	250
Large-scale	CelebA-HQ	28K	2K
	FFHQ	60K	10K
	PSV	14.9K	100
	Places365	1.8M	36.5K

4 Experiments

4.1 Experimental setting

Python and PyTorch were used to build the proposed framework. We set $\lambda_{\text{lpips}} = 1.5$, $\lambda_{\text{adv}} = 1$, $\lambda_{\text{patch}} = 1$, and $T = 3$ for all experiments in both training and testing phases, unless otherwise specified. We used the Adam optimizer with first momentum coefficient $\beta_1 = 0.5$, second momentum coefficient $\beta_2 = 0.999$, and learning rate 0.0002. Our masks were created with the CMOD mask generation algorithm [12]. Our generator contains 31.76M parameters and achieves around 21 FPS for each residual reasoning step on an NVIDIA GeForce RTX 2080 GPU with 8 GB memory.

We compared GRIG to various state of the art image inpainting methods: Globally&Locally (G&L) [36], Contextual Attention (CA) [40], Partial Convolutions (PConv) [42], GMCNN [33], Gated Convolution (GConv) [13], Recurrent Feature Reasoning (RFR) [15], AOT-GAN (AOT) [48], Comod-GAN (CMOD) [12], Lama [7], MAT [8], FcF [56], TFill [49], and ZITS [44]. We also compared GRIG to an inpainting model (Projected) based on the light-weight Projected GAN [18] to further demonstrate the superiority of GRIG for data-efficient image inpainting. The publicly available MMEditing framework [69], an open-source image and video editing toolbox based on PyTorch, implements the models of G&L, CA, PConv, and GConv. We used the authors' codes for GMCNN, RFR, AOT, Lama, MAT, FcF, TFill, and ZITS. We used the authors' TensorFlow-based code to create a PyTorch-based version of CMOD. To implement the Projected model, we added a mirrored encoder of Projected GAN with skip connections and perceptual similarity $\mathcal{L}_{\text{lpips}}$. This Projected model was created using PyTorch with the same hyper-parameters as GRIG with $\lambda_{\text{lpips}} = 1.5$.

Experiments were conducted on ten small-scale datasets (CHASE [70], Shell [17], Skull [17], Anime [17], Fauvism [17], Moongate [17], Cat [71], Dog [71], Pokemon (pokemon.com), and Art (wikiart.org)) and four large-scale image datasets (including CelebA-HQ [72], FFHQ [73], Paris Street View (PSV) [74], and Places365 [75]). Details of the sizes of the datasets are given in Table 1. We used the original training and testing splits from the PSV and Places365 datasets, while other datasets were split using random sampling. To ensure fairness, we used the same training/testing splits for all experiments.

All images were resized to a resolution of 256×256. All compared models were retrained on the datasets mentioned in the paper, using a batch size of 8, unless otherwise noted. During testing, various irregular masks with different mask ratios [42] and a fixed center 25% (128×128) rectangular mask were used to simulate different situations for all experiments. All methods in our evaluation replaced the unmasked known regions with the original image. All models were trained and tested on NVIDIA V100 GPUs (with 32 GB memory).

Since L1 distance, PSNR, and SSIM all heavily prefer blurry results [12], we used Fréchet inception distance (FID) [76] and LPIPS metrics for quantitative evaluation following established practice in recent literature [7].

4.2 Comparison on small-scale datasets

To evaluate the performance on ten small-scale datasets (see Table 1), all models were trained with $400,000$ image batches. We implemented the early-stopping technique for each method, ensuring that each model is adequately trained without overfitting and achieved optimal performance for evaluation. Fig. 4 and Fig. 5 quantitatively compare GRIG to the other state of the art inpainting methods on the ten small-scale datasets. To compare the performance of image inpainting, various irregular masks with different mask ratios [42], as well as a fixed center 25% (128×128) rectangular mask, were used to simulate various scenarios. In the small-scale setting, differentiable data augmentation [77] was applied for all compared methods when sampling images in the training phase. As Fig. 4 and Fig. 5 show, GRIG outperforms all baselines in terms of FID and LPIPS metrics by large margins on most benchmarks for various kinds of masks. For most datasets, significant gains were obtained by our method. Notably, for a 50-60% mask ratio, GRIG achieves a relative improvement of FID to the second-best methods of 9.12% (CHASE), 13.98% (Shell), 1.26% (Skull), 12.65% (Anime), 4.06% (Fauvism), 14.86% (Moongate), 6.64% (Cat), 16.40% (Dog), and 1.60% (Art).

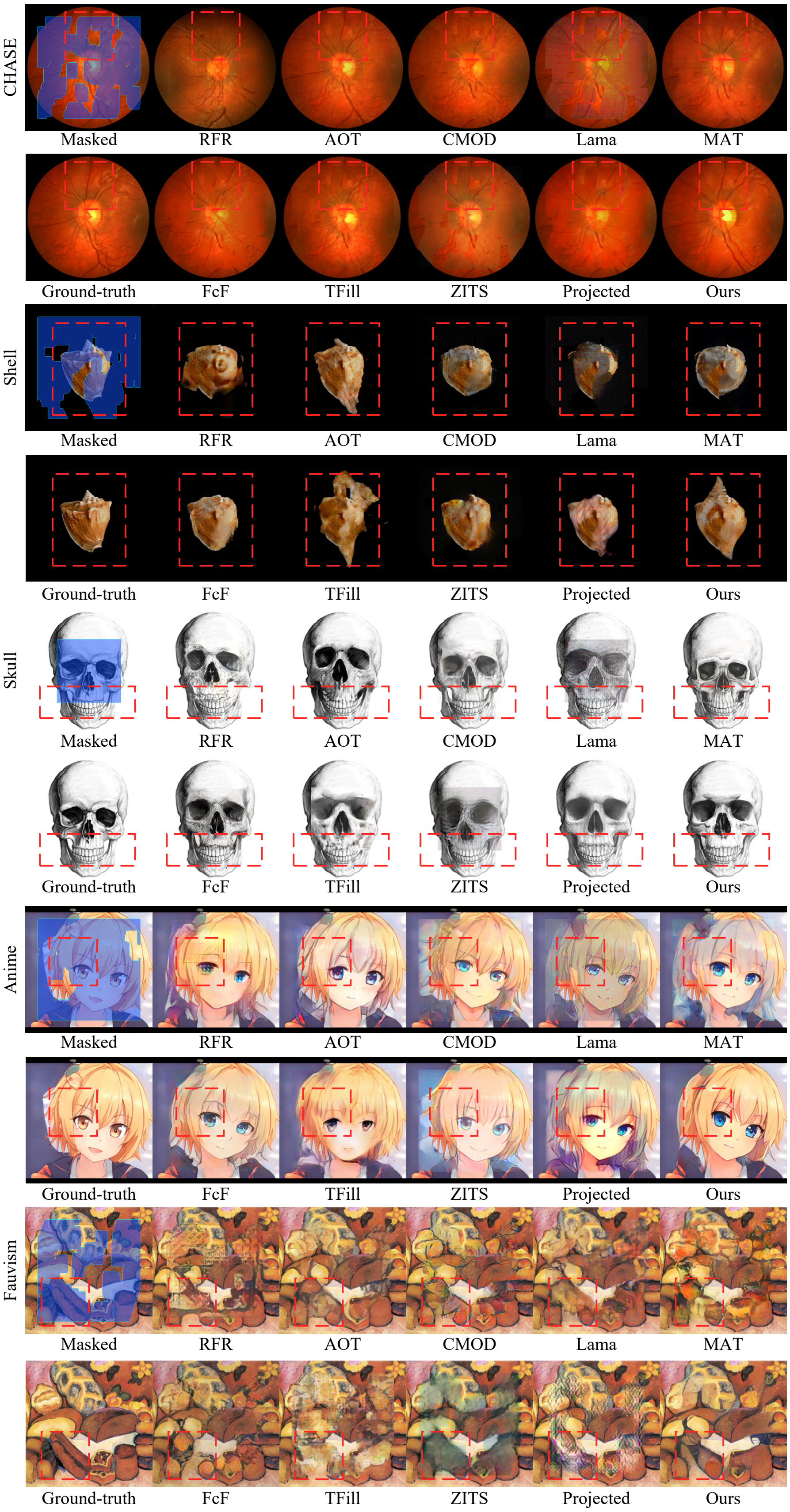


Fig. 6 Results of GRIG and other state of the art image inpainting methods on small-scale datasets (CHASE, Shell, Skull, Anime, Fauvism).

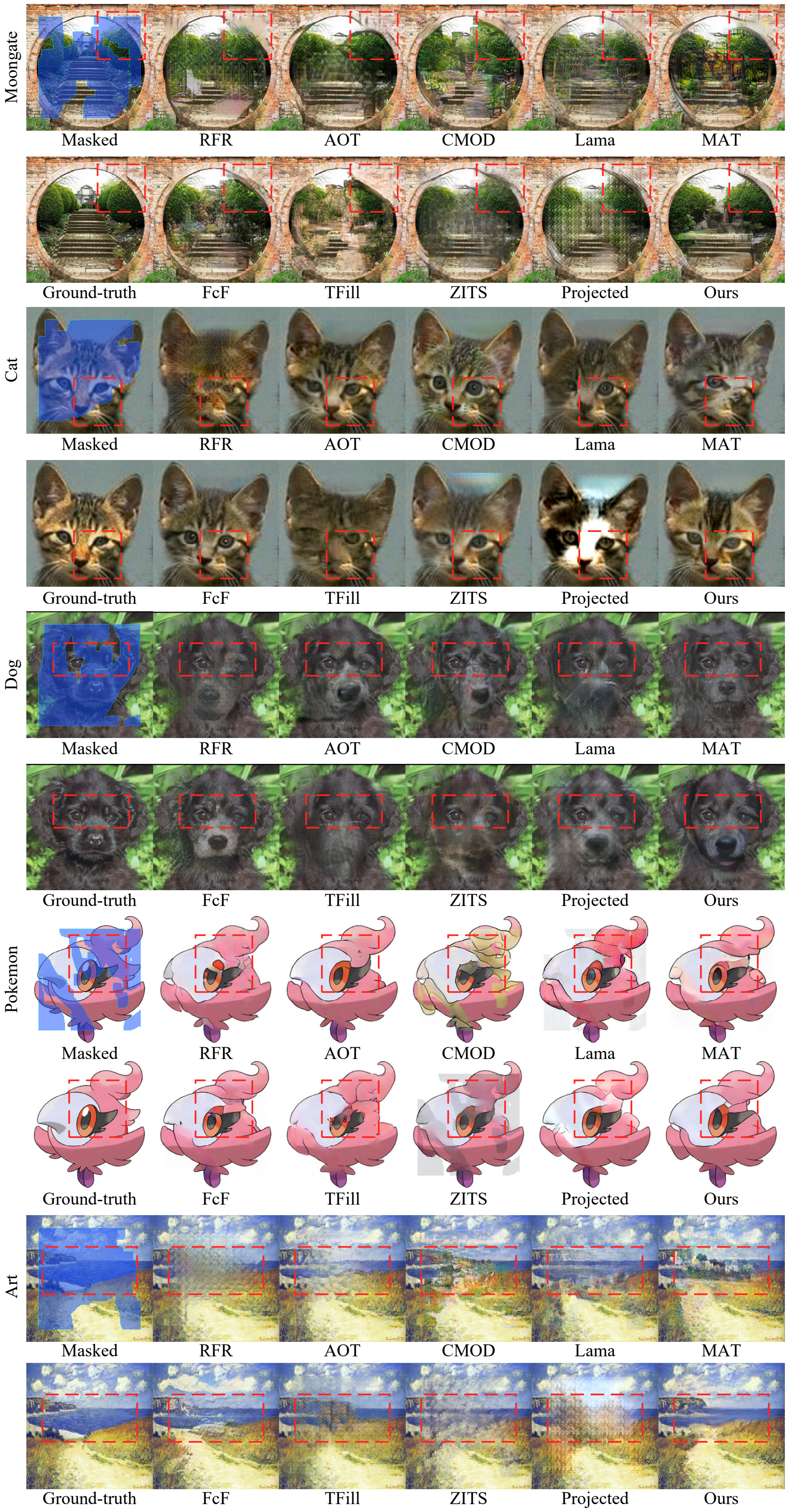


Fig. 7 Results of GRIG and other state of the art image inpainting methods on small-scale datasets (Moongate, Cat, Dog, Pokemon, Art).

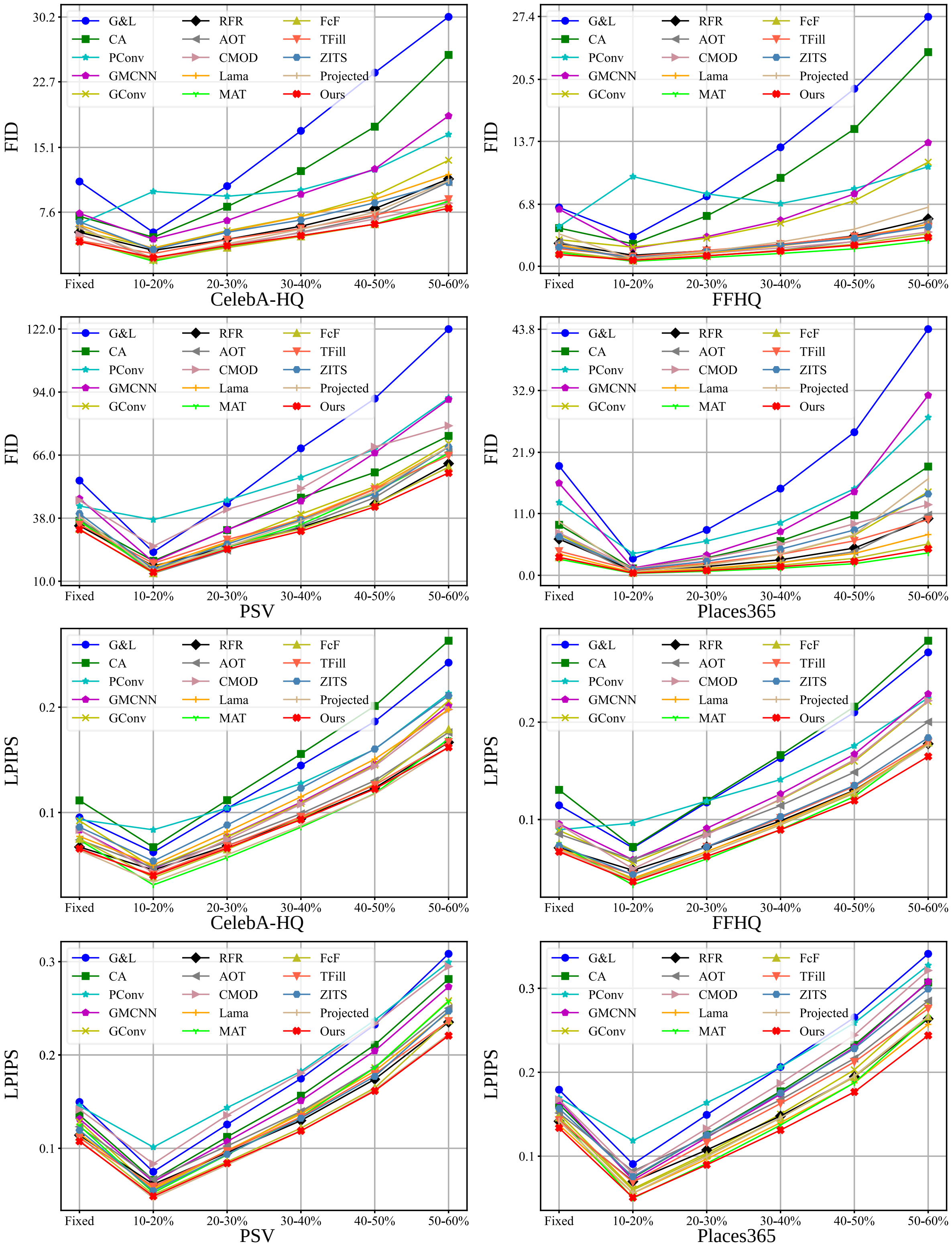


Fig. 8 Quantitative comparisons of GRIG with various state of the art image inpainting methods on four large-scale datasets, for FID and LPIPS evaluation metrics. In each graph, the horizontal axis indicates mask ratios; ‘Fixed’ denotes the fixed center 25% rectangular mask.

Fig. 6 and Fig. 7 present inpainted results for the compared methods. It reveals that most methods fail to produce plausible contents for datasets with fewer than 100 training samples (for example, CHASE, Shell, Skull, and Anime) due to overfitting to features and patterns from a small number of samples. When trained on datasets with more than 500 samples (such as Pokemon and Art), some methods may be able to fill more semantic content within masked areas. However, artifacts can still be seen under close inspection. When the masked area is large, RFR is prone to producing repetitive image patches in inpainted regions, and while AOT, Lama, and ZITS can inpaint structures in missing regions, they leave artifacts in fine detail. We also noticed that Lama and ZITS have similar blurring phenomena in the inpainted regions when trained on small-scale data, which may be because the Fast Fourier Convolution [78] (FFC) overfits the limited global repeating patterns [56], harming subsequent feature extraction. CMOD, MAT, and TFill tend to overfit the training data due to their large numbers of learnable parameters. Projected GANs can handle the semantic structure, but they may introduce color inconsistency around mask boundaries. By combining the benefits of FFC and stochasticity, FcF shows robust performance on both textural and structural image inpainting. Fig. 4, Fig. 5, Fig. 6, and Fig. 7 demonstrate that GRIG can achieve better performance on quantitative metrics and visual quality, even though our method has more learnable parameters (31.76M) than those of GConv (4.0M) and is trained on limited samples. GRIG demonstrates strong generalization capabilities in various small-scale datasets with differing numbers of training samples, and produces images with higher visual quality.

We believe that there are three reasons for the better generalization performance and inpainting quality achieved on data-efficient image inpainting. Firstly, our iterative residual reasoning strategy enables the generator to use information learned in previous iterations while also increasing the diversity of inputs to improve performance. Secondly, the self-attention mechanism in Transformers [10] has advantages in leveraging existing information for further context reasoning. In our generator, the encoder and decoder are used to extract local features, while the Restormer Transformer blocks [60] are used for global context reasoning. Thirdly, the projected discriminator and forged-patch discriminator, with 2.829M and 2.765M learnable parameters, respectively, help improve the generality of our method. The projected discriminator focuses on images at the semantic level based on the generality of pre-trained features. The forged-patch discriminator focuses on learning patch-level consistency to capture patch statistics and distinguishing between real and inpainted patches to prevent overfitting by avoiding the need to memorize the entire image.

4.3 Comparison on large-scale datasets

We also compared our method to the same inpainting methods on four large-scale datasets. All methods were trained with their default settings to ensure fair comparisons. Our model was trained with $1,000,000$ image batches on CelebA-HQ, FFHQ, and PSV, respectively, and $2,000,000$ image batches on Places365.

The quantitative results in Fig. 8 show that GRIG outperforms the majority of the SOTA inpainting methods in terms of FID and LPIPS metrics on large-scale datasets. In particular, GRIG achieves the best FID scores on PSV, and the best LPIPS scores on PSV and Places365. MAT has the best FID scores on FFHQ and Places365. Overall, GRIG performs comparably to MAT on the other large-scale datasets while containing many fewer learnable weights (31.76M) than MAT (62.0M). Our iterative residual learning effectively assists the networks in decomposing the inpainting process into multiple reasoning steps with the progressive refinement of inpainting results. Moreover, the decoupling of image distribution learning into image-level and patch-level constraints with our projected discriminator and forged-patch discriminator helps our GRIG model achieve excellent performance in both data-efficient scenarios and large datasets.

Fig. 9 shows a corresponding qualitative performance evaluation. It demonstrates that semantic inpainting on large masks remains difficult for most inpainting methods. RFR produces repetitive image patches in inpainted regions because iterative refinement in feature space may overlook fine details in image space. AOT and CMOD perform well on these datasets. However, with complex backgrounds, they struggle with larger masked areas in some cases. MAT and FcF handle texture and structure inpainting well and generalize well to different types of datasets. Because one-time inferencing cannot re-adjust inpainted results, complex backgrounds are likely to have negative impacts on MAT and FcF inpainting quality. With their multi-stage inpainting processes, TFill and ZITS utilize Transformer architectures to notably enhance the visual quality of inpainted pixels. However, their performance may be influenced when previous networks in the process do not perform optimally. Because fine details are easily overlooked in projected features, projection-based models [18] tend to produce blurred results. Our GRIG can inpaint plausible contents in complex structures with high mask ratios.

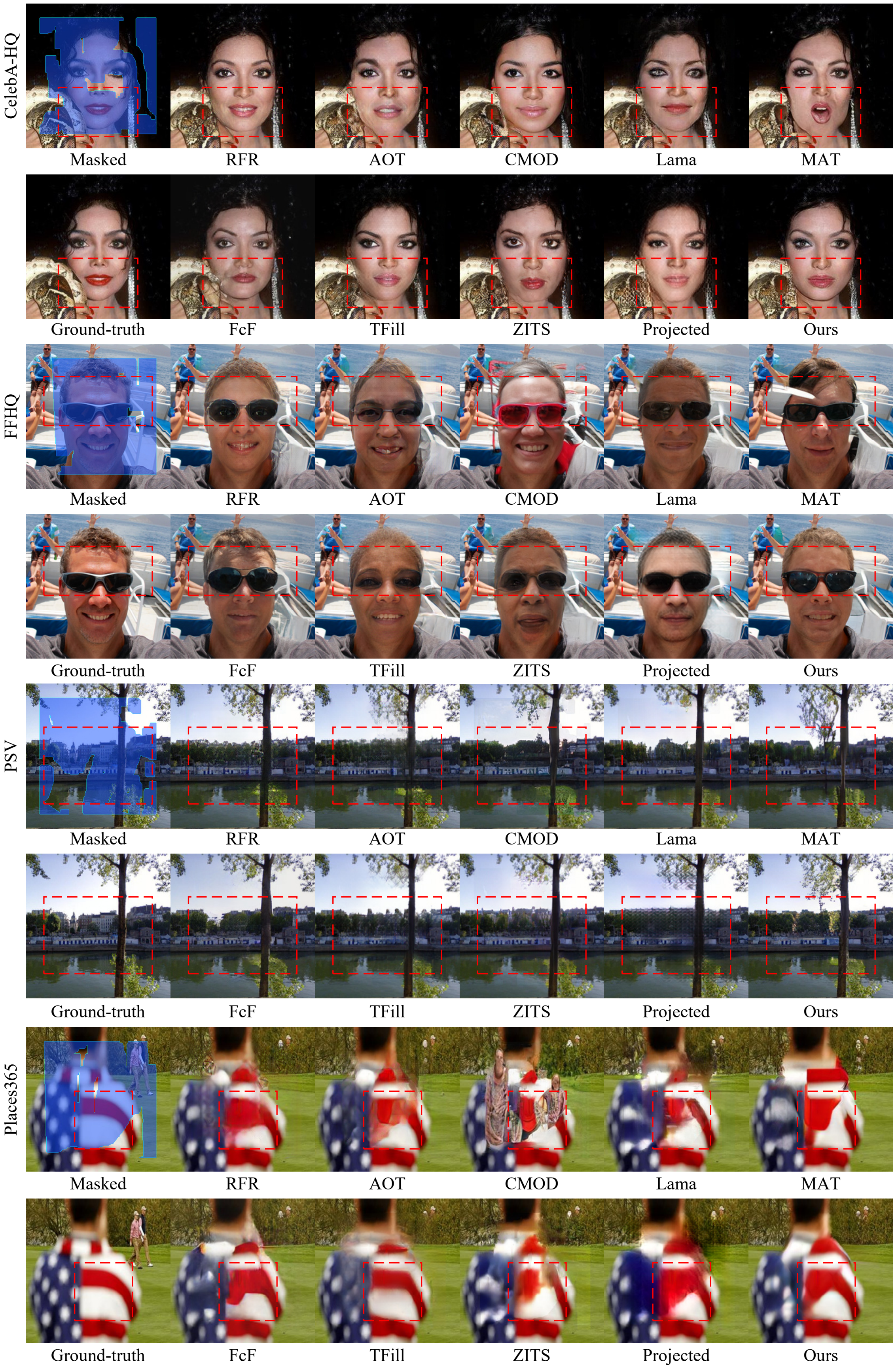


Fig. 9 Visual comparison of GRIG and state of the art image inpainting methods on large-scale datasets.

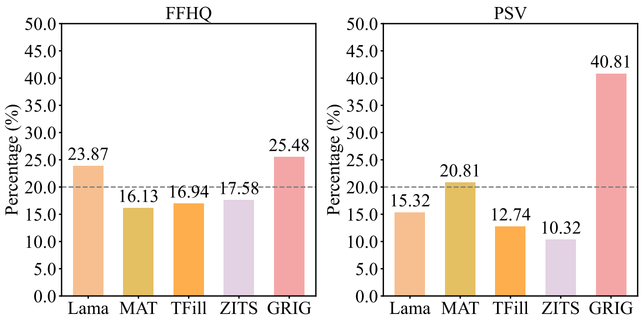


Fig. 10 User study results on the FFHQ and PSV datasets using state of the art methods (Lama, MAT, TFill, and ZITS). We give percentages of cases in which each method is ranked first over others.

Table 2 User study results: average rankings of compared methods on the FFHQ and PSV datasets. **Bold** indicates best results.

Dataset	Lama	MAT	TFill	ZITS	GRIG
FFHQ	2.88	3.16	3.10	2.99	**2.87**
PSV	2.97	3.13	3.25	3.39	**2.26**

We conducted a user study using various state of the art methods (Lama, MAT, TFill, and ZITS) on the FFHQ and PSV datasets to demonstrate GRIG's inpainting performance on large-scale datasets. For each dataset, we randomly sampled 100 images from the testing set, then randomly selected and assigned 20 of those images to each participant. Each question contained a masked image, a ground-truth image, and shuffled inpainted images from the five compared methods. The users were asked to rank the compared methods based on visual quality and realism. We recruited 31 participants, totaling 620 votes for each method on each dataset.

Fig. 10 shows the percentage of time each method achieved the top rank on the FFHQ and PSV datasets. Our GRIG had the highest percentages at 25.48% on FFHQ and 40.81% on PSV. Table 2 displays the average rankings for each compared method. All average rankings are within the range of [2.0, 3.5], indicating comparable performance for these methods. Notably, our GRIG had the best average rankings on FFHQ and PSV, of 2.87 and 2.26 respectively. The user study results show that our GRIG produces high-quality image inpainting results.

4.4 Comparison on various few-shot settings

We conducted comparisons on various few-shot settings on small-scale and large-scale datasets. The term "n-shot" means that n images in each training set in Table 1 were selected for training and the test sets were kept unchanged.

The quantitative results of FID scores are shown in Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16, Fig. 17, and Fig. 18. FID and LPIPS scores decrease as the number of training samples increases (e.g., 50-shot images), implying that more training samples could improve inpainting quality. We further calculated the mean scores of FID and LPIPS across all masked ratios and few-shot settings for each dataset, as shown in Table 3. It highlights our GRIG's superiority in few-shot settings. For example, when trained on the Dog dataset, our GRIG achieved a mean FID score of 68.02, indicating a 17.18% relative improvement over the second-best method FcF (with 82.13). The results demonstrate that our method can improve the performance on few-shot scenarios.

Fig. 19 presents visual comparisons on various few-shot settings. The results reveal that GRIG achieves greater visual fidelity compared to the SOTA methods. For instance, when trained on 30 and 50 samples, GRIG produces sharp structural and clear texture contents, while compared methods show more blurry results. Fig. 20 presents more inpainted examples of our GRIG. The quality of inpainted images drops quickly when models were trained on fewer samples. For example, models trained on 5-shot images are unable to inpaint semantic structures within masked areas; while models trained on 10-shot and 30-shot images can inpaint more plausible contents, some output results still show obvious color inconsistency around mask boundaries. A similar phenomenon is also shown on CMOD and MAT in Fig. 6. In contrast, models trained on 50-shot settings produce sharper results with more complex textures and rich colors. We can find that the more training samples the models trained on, the better their performance on both quantitative and qualitative evaluations. Our method produces more plausible contents when trained on few-shot settings.

4.5 Network complexity of recent SOTA methods

In Table 4, we show the model complexity of our GRIG and the state of the art methods in terms of number of parameters, and FLOPs needed for a resolution of 256 × 256. Our GRIG model achieves optimal efficiency with the lowest number of FLOPs at an iterative reasoning step of $T = 1$, and is ranked as the fourth most efficient at $T = 3$. While our model does not have the fewest parameters, its strong performance on small-scale datasets highlights a different strength. This success is not due to a reduced risk of overfitting from fewer parameters; instead, it is attributable to the effectiveness of our proposed framework for data-efficient image inpainting. The robust capacity of our network also plays a pivotal role in securing competitive results on larger-scale datasets. Additionally, our superior image inpainting performance on small-scale datasets, large-scale datasets, and various few-shot settings

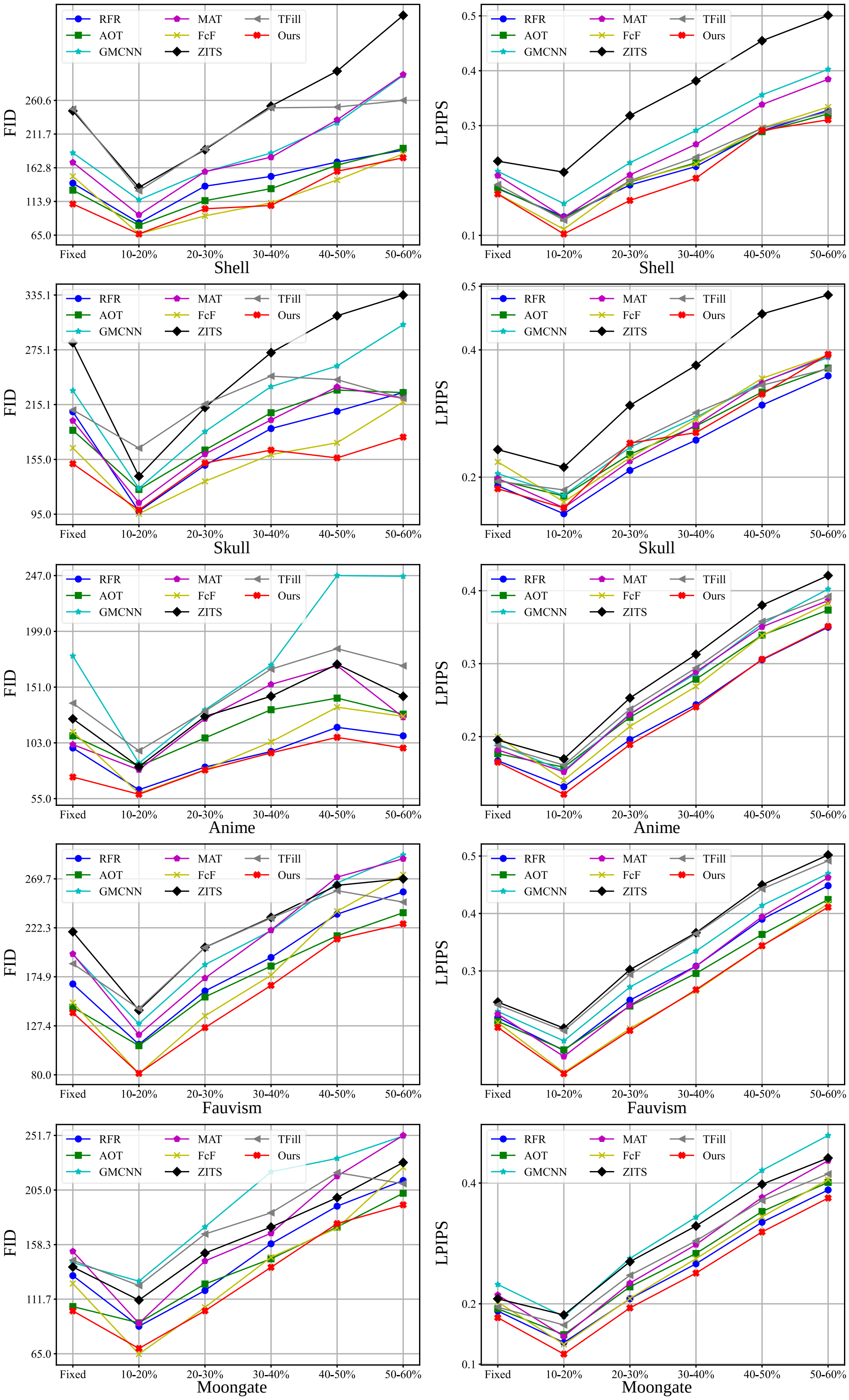


Fig. 11 Quantitative comparisons between models (RFR, AOT, GMCNN, MAT, FcF, TFill, ZITS, and ours) trained on the 5-shot setting of Shell, Skull, Anime, Fauvism, and Moongate datasets. In each graph, the horizontal axis indicates mask ratios; 'Fixed' denotes the fixed center 25% rectangular mask.

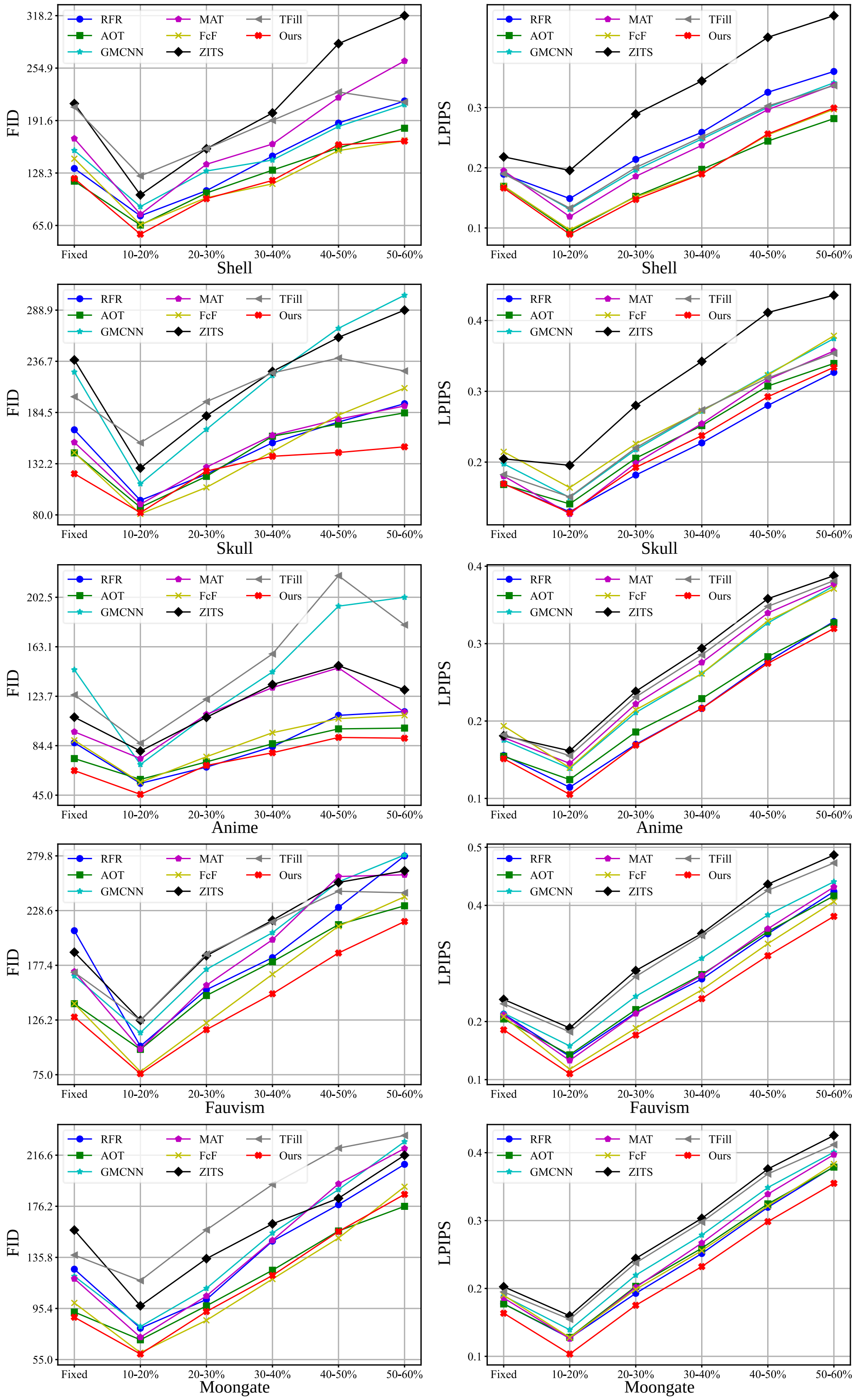


Fig. 12 Quantitative comparisons between models (RFR, AOT, GMCNN, MAT, FcF, TFill, ZITS, and ours) trained on the 10-shot setting of Shell, Skull, Anime, Fauvism, and Moongate datasets. In each graph, the horizontal axis indicates mask ratios; ‘Fixed’ denotes the fixed center 25% rectangular mask.

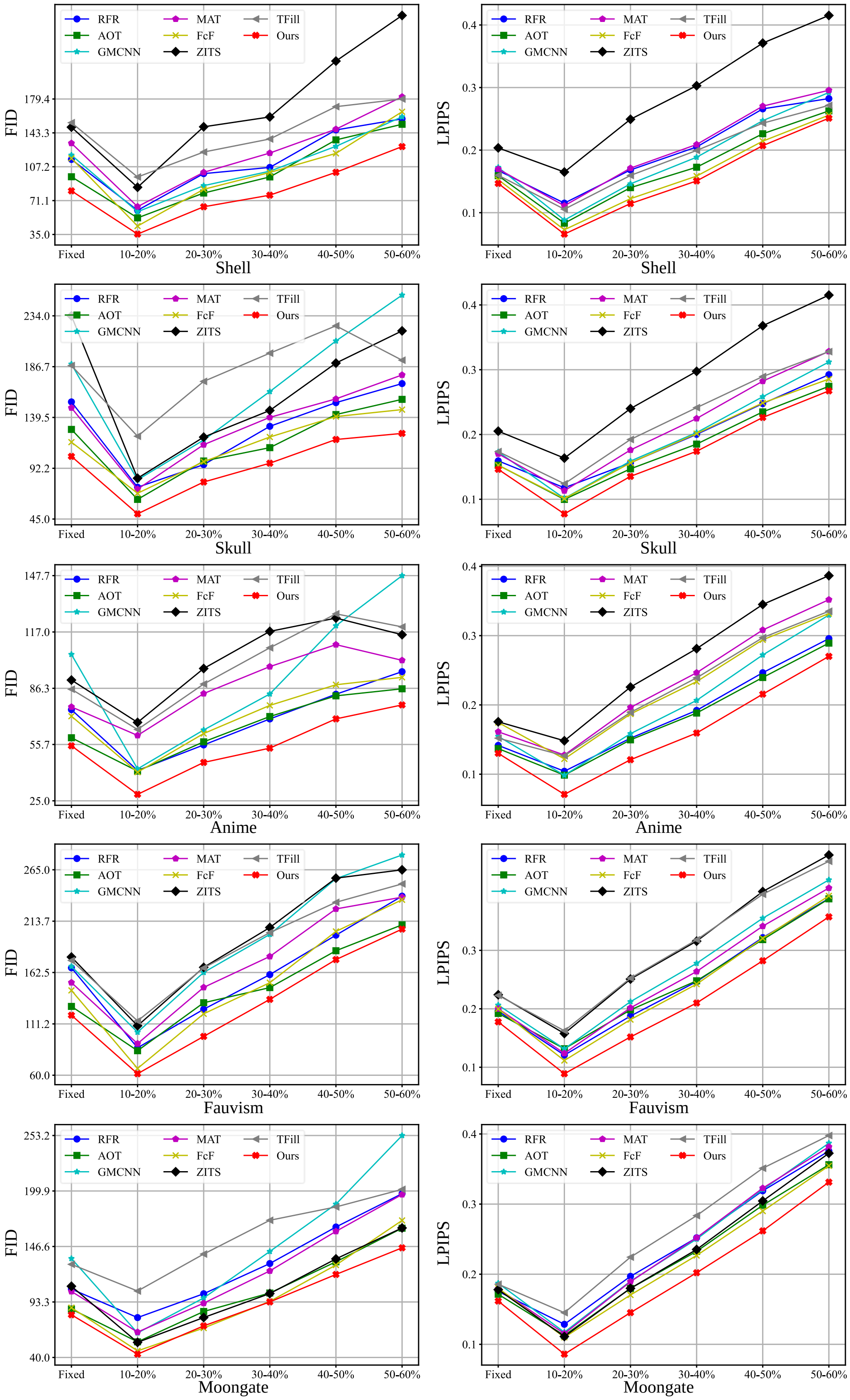


Fig. 13 Quantitative comparisons between models (RFR, AOT, GMCNN, MAT, FcF, TFill, ZITS, and ours) trained on the 30-shot setting of Shell, Skull, Anime, Fauvism, and Moongate datasets. In each graph, the horizontal axis indicates mask ratios; ‘Fixed’ denotes the fixed center 25% rectangular mask.

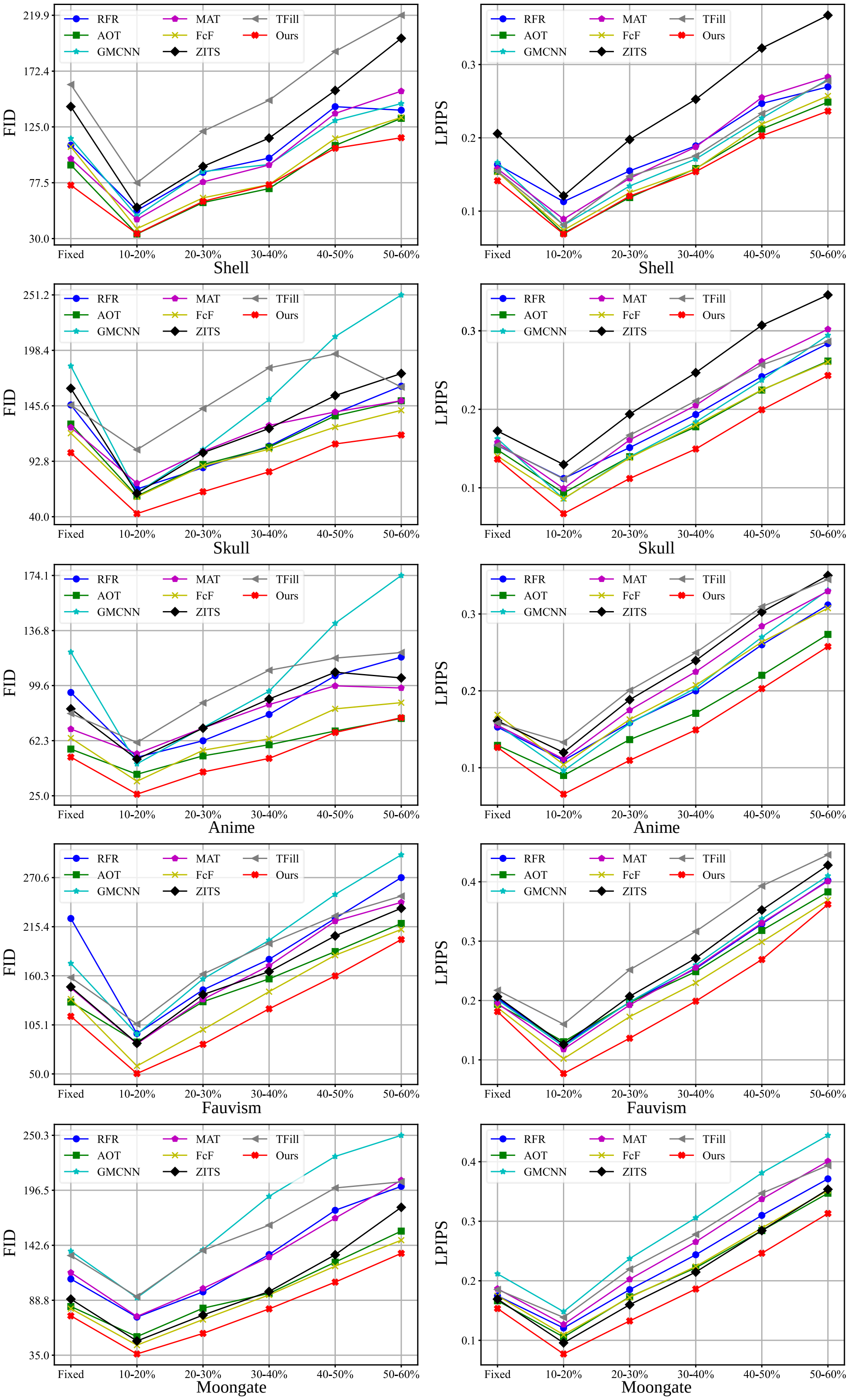


Fig. 14 Quantitative comparisons between models (RFR, AOT, GMCNN, MAT, FcF, TFill, ZITS, and ours) trained on the 50-shot setting of Shell, Skull, Anime, Fauvism, and Moongate datasets. In each graph, the horizontal axis indicates mask ratios; ‘Fixed’ denotes the fixed center 25% rectangular mask.

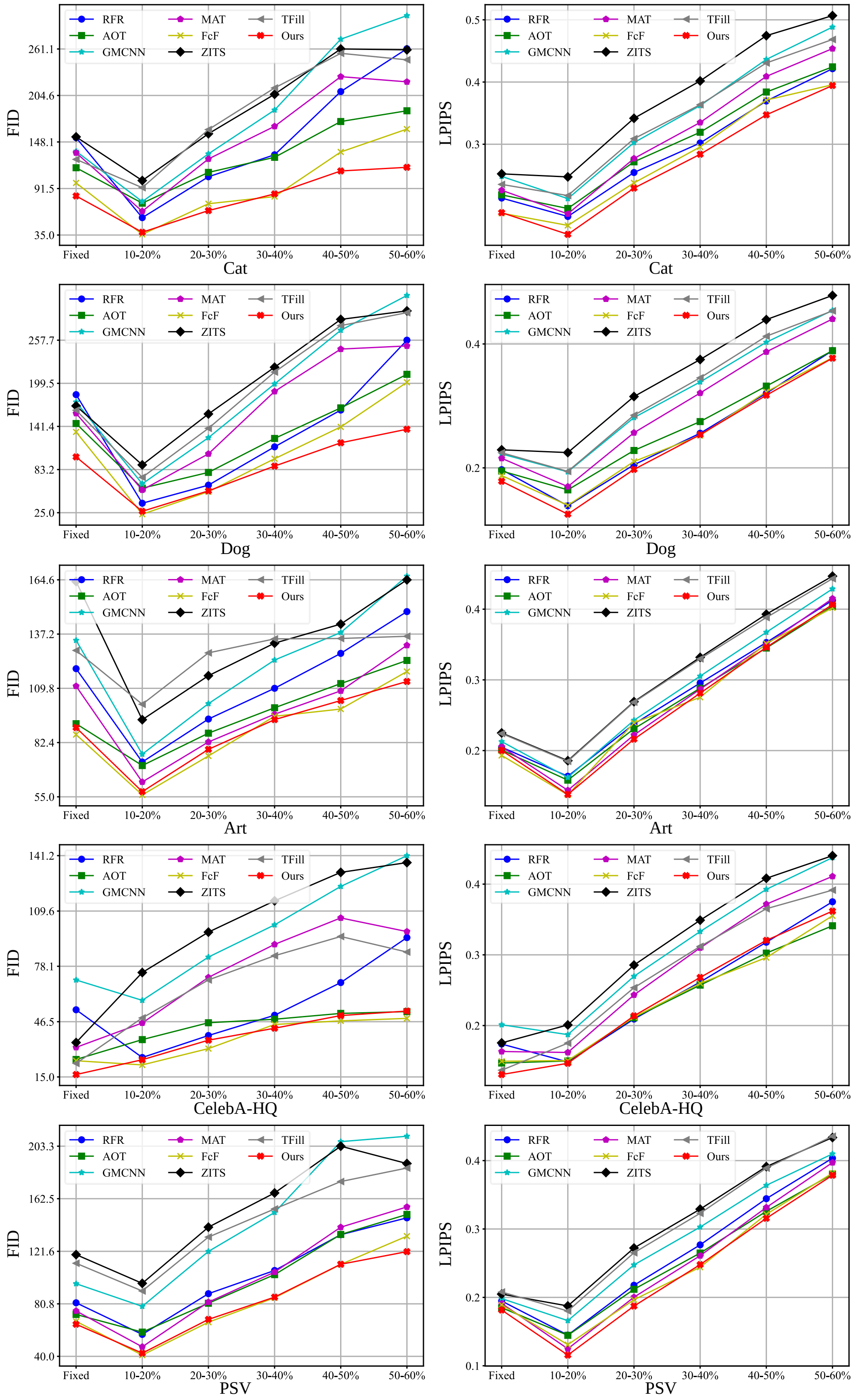


Fig. 15 Quantitative comparisons between models (RFR, AOT, GMCNN, MAT, FcF, TFill, ZITS, and ours) trained on the 5-shot setting for Cat, Dog, Art, CelebA-HQ, and PSV datasets. In each graph, the horizontal axis indicates mask ratios; ‘Fixed’ denotes the fixed center 25% rectangular mask.

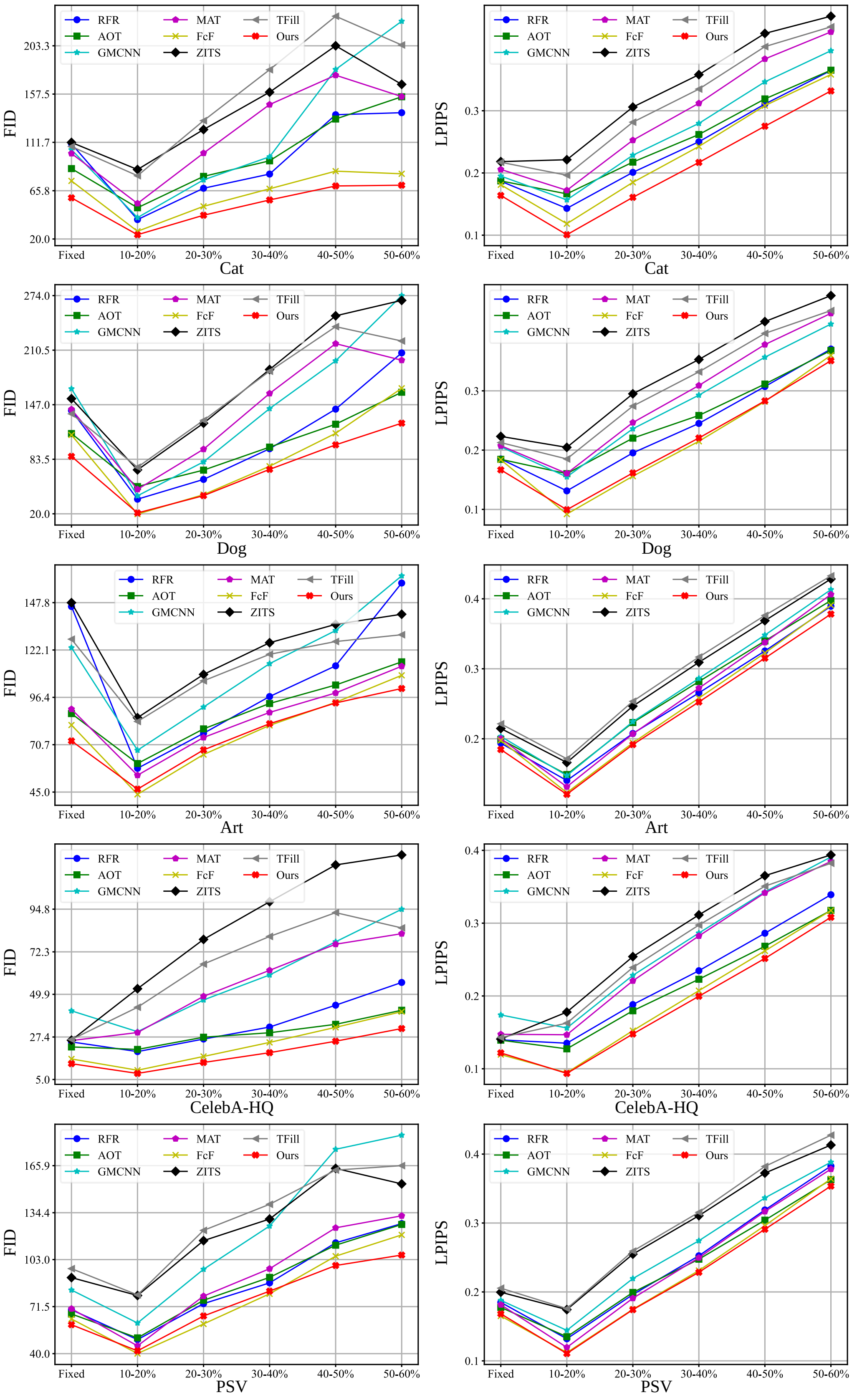


Fig. 16 Quantitative comparisons between models (RFR, AOT, GMCNN, MAT, FcF, TFill, ZITS, and ours) trained on the 10-shot setting for Cat, Dog, Art, CelebA-HQ, and PSV datasets. In each graph, the horizontal axis indicates mask ratios; 'Fixed' denotes the fixed center 25% rectangular mask.

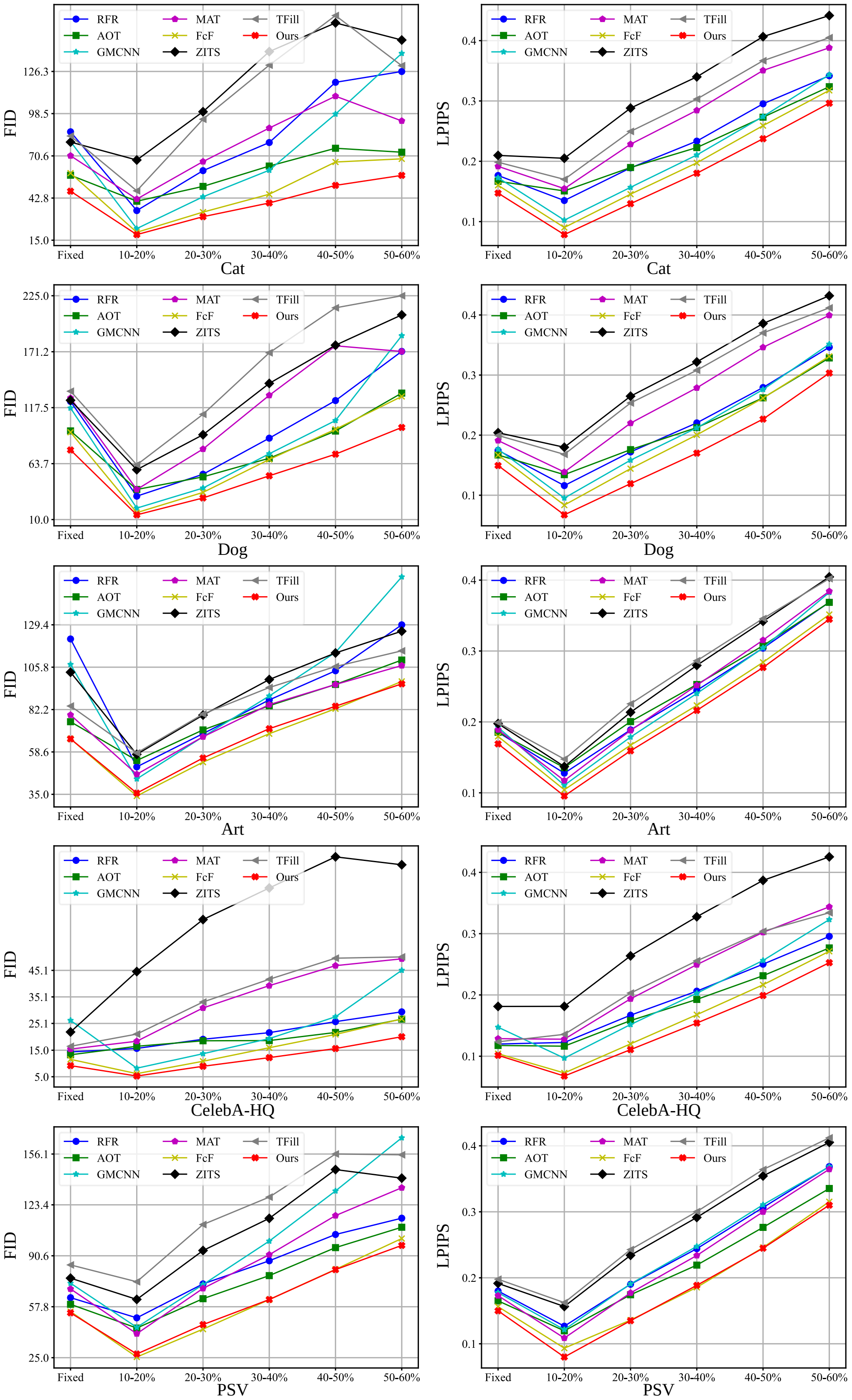


Fig. 17 Quantitative comparisons between models (RFR, AOT, GMCNN, MAT, FcF, TFill, ZITS, and ours) trained on the 30-shot setting for Cat, Dog, Art, CelebA-HQ, and PSV datasets. In each graph, the horizontal axis indicates mask ratios; ‘Fixed’ denotes the fixed center 25% rectangular mask.

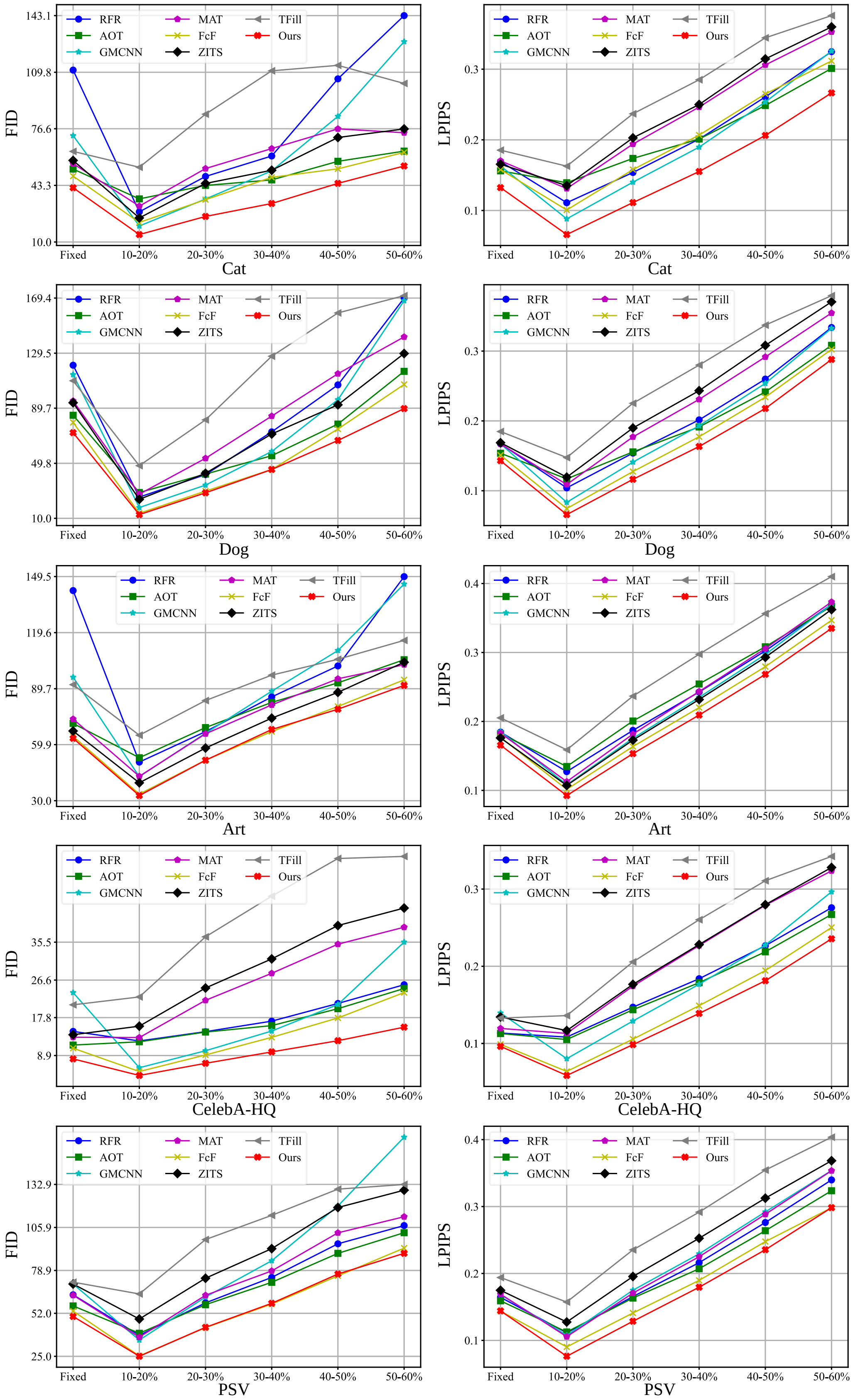


Fig. 18 Quantitative comparisons between models (RFR, AOT, GMCNN, MAT, FcF, TFill, ZITS, and ours) trained on the 50-shot setting for Cat, Dog, Art, CelebA-HQ, and PSV datasets. In each graph, the horizontal axis indicates mask ratios; 'Fixed' denotes the fixed center 25% rectangular mask.

Table 3 Comparisons of mean FID and LPIPS scores across all mask ratios and few-shot settings for each dataset. **Bold** indicates best results.

Metrics	Methods	Shell	Skull	Anime	Fauvism	Moongate	Cat	Dog	Art	CelebA-HQ	PSV
FID	RFR	127.31	144.71	83.72	183.50	138.11	104.10	109.63	103.29	31.92	86.17
	AOT	112.06	140.73	80.66	160.93	115.87	85.38	95.46	87.04	26.99	82.93
	GMCNN	139.87	192.34	130.63	201.42	164.46	111.20	128.53	106.64	49.19	113.25
	MAT	146.67	147.96	101.18	185.71	142.61	104.36	129.27	85.53	46.63	88.87
	FcF	111.09	130.80	81.87	157.42	110.28	63.21	82.13	76.01	23.65	73.67
	TFill	176.16	190.69	122.77	196.17	149.54	134.01	156.56	106.38	52.08	122.49
	ZITS	190.74	193.72	108.92	197.22	134.08	125.43	147.07	106.96	68.22	117.37
	Ours	**100.12**	**114.77**	**66.49**	**140.03**	**105.21**	**53.45**	**68.02**	**74.83**	**19.27**	**69.17**
LPIPS	RFR	0.2163	0.2122	0.2074	0.2652	0.2426	0.2413	0.2279	0.2508	0.2093	0.2391
	AOT	0.1875	0.2141	0.2074	0.2610	0.2385	0.2449	0.2305	0.2547	0.1992	0.2274
	GMCNN	0.2191	0.2292	0.2307	0.2824	0.2785	0.2530	0.2496	0.2550	0.2344	0.2492
	MAT	0.2190	0.2296	0.2416	0.2683	0.2576	0.2766	0.2677	0.2509	0.2402	0.2338
	FcF	0.2842	0.2442	0.2337	0.2448	0.2374	0.2188	0.2061	0.2334	0.1990	0.2172
	TFill	0.2085	0.2034	0.2478	0.3147	0.2688	0.2989	0.2923	0.2867	0.2480	0.2867
	ZITS	0.2998	0.2296	0.2614	0.3108	0.2582	0.3135	0.3001	0.2709	0.2722	0.2753
	Ours	**0.1773**	**0.2028**	**0.1869**	**0.2266**	**0.2106**	**0.2023**	**0.1983**	**0.2298**	**0.1774**	**0.2051**

Table 4 Network complexity of various image inpainting methods, including GRIG. **Bold** indicates best.

Method	RFR	AOT	CMOD	Lama	MAT	FcF	TFill	ZITS	GRIG$_{T=1}$	GRIG$_{T=3}$
# Parameters (M)	30.59	**15.20**	79.17	27.05	62.0	70.33	109.45	68.16	31.76	31.76
FLOPs (G)	206.17	72.88	90.25	42.87	139.11	40.26	45.45	270.08	**20.47**	61.41

Table 5 Ablation study providing FID scores for (A) GRIG without the forged-patch discriminator, (B) GRIG without the projected discriminator, (C) GRIG with forged-patch discriminator replaced by PatchGAN's discriminator, (D) GRIG with forged-patch discriminator replaced by SM-PatchGAN's discriminator, (E) GRIG with Transformer blocks replaced by down-sampling and up-sampling blocks, and full GRIG . Results were evaluated on 50–60% mask ratios. **Bold** indicates best results.

Dataset	(A)	(B)	(C)	(D)	(E)	GRIG
CHASE	73.96	57.00	56.89	64.17	59.16	**55.84**
Anime	77.49	68.56	66.03	71.12	69.96	**65.05**
Dog	65.33	62.92	61.17	59.83	61.87	**58.49**
Art	96.84	79.19	79.16	78.83	77.35	**77.32**
CelebA-HQ	10.14	8.92	8.41	8.96	8.62	**8.06**
PSV	60.53	61.76	59.62	61.07	61.45	**58.08**

demonstrate that GRIG shows a good trade-off between image inpainting quality and computational resources.

4.6 Ablation study

4.6.1 Ablation study on components

We further analyzed the effects of the components of GRIG. To analyze the effects of discriminators in GRIG, we individually removed each discriminator and replaced our forged-patch discriminator with PatchGAN [64] and SM-PatchGAN [48], in turn. All compared discriminators used the same network architecture of 70×70-sized PatchGAN. To demonstrate that incorporating Transformer blocks can further improve the inpainting quality, we tested replacing Transformer blocks [60] with down-sampling and up-sampling blocks [17]. We evaluated inpainting performance to show the impact of these changes. Table 5 shows quantitative results of the compared variants. GRIG outperforms all variants in terms of FID score on various small-scale and large-scale datasets. The FID scores increase dramatically when removing either the forged-patch discriminator (model A) or the projected discriminator (model B). Replacing our forged-patch discriminator with other discriminators (models C and D) also leads to higher FID scores. These results indicate that removing our discriminators or replacing the proposed forged-patch discriminator causes a significant degradation in inpainting performance. The best FID scores of our GRIG on various datasets validate the effectiveness of our forged-patch discriminator for performance boosting and mitigating overfitting on small-scale image inpainting. Additionally, without the global context integration of Transformers (model E), the model performs worse. Our generator leverages both advantages of shallow feature extraction and global context reasoning to enhance the visual quality of inpainted images.

Fig. 21 shows corresponding visual results. When removing the forged-patch discriminator, the inpainted results show noticeable artifacts around mask boundaries, and the produced textures are blurred, as shown in Fig. 21(A). When the projected discriminator is removed, both quantitative performance and visual quality suffer. It is more difficult to maintain the semantic structure of outputs in this case, e.g., the asymmetrical Anime face, as shown in Fig. 21(B). The

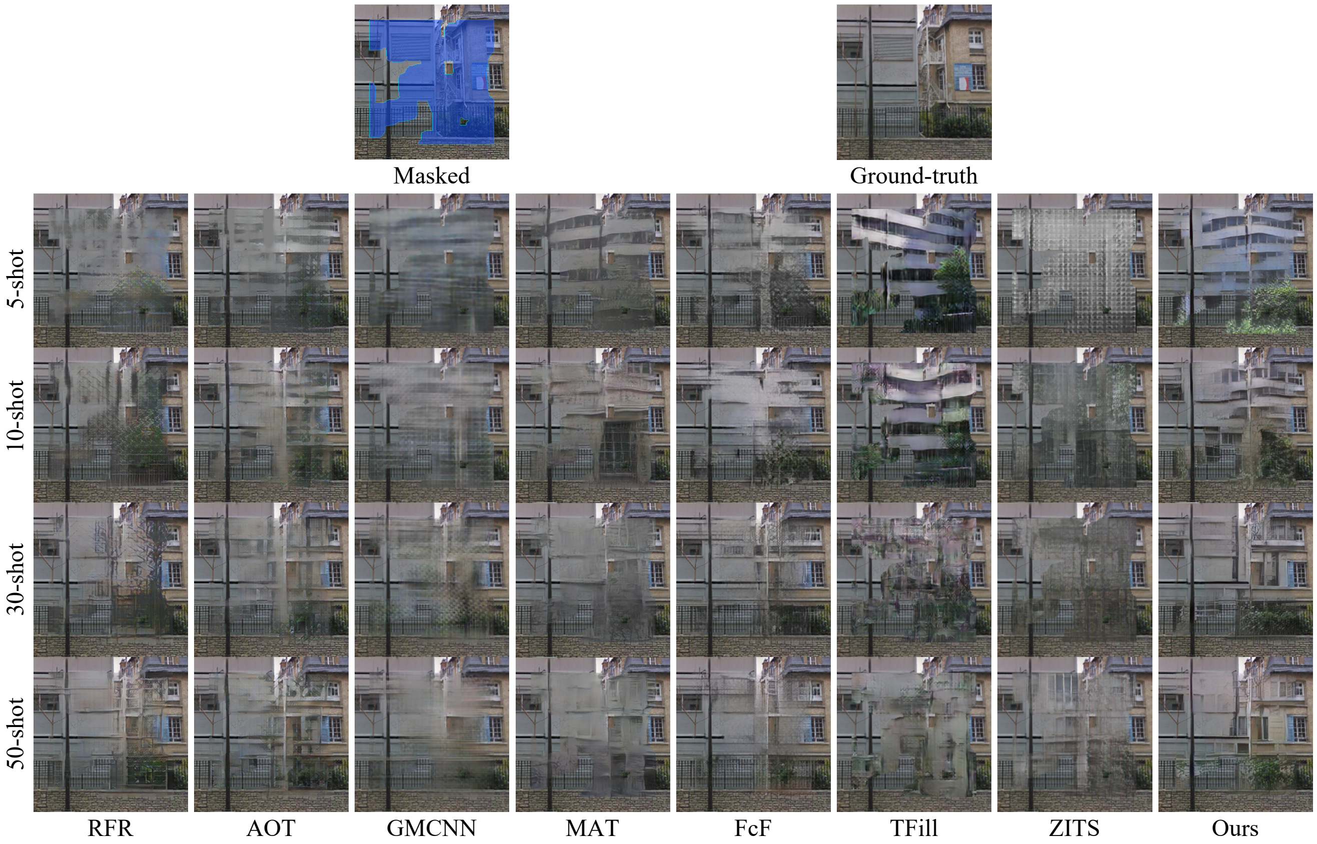


Fig. 19 Visual comparison of results of models (RFR, AOT, GMCNN, MAT, FcF, TFill, ZITS, and ours) trained on various few-shot settings.

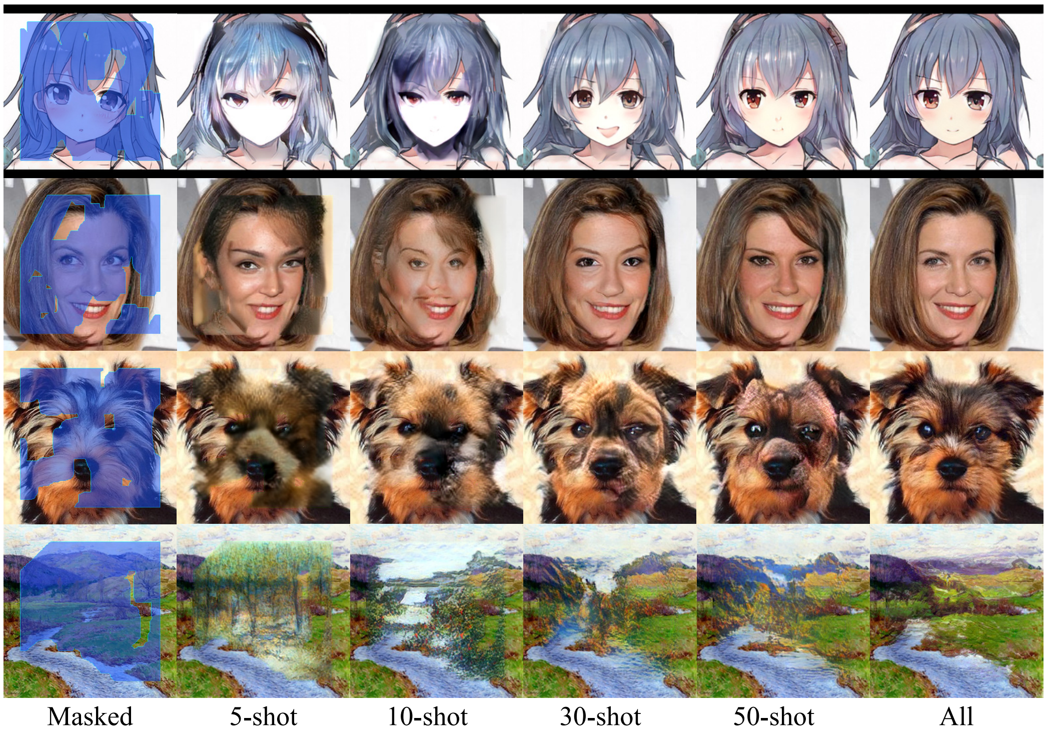


Fig. 20 Visual results of our GRIG models trained on various few-shot settings. “All” means the training sets mentioned in Table 1.

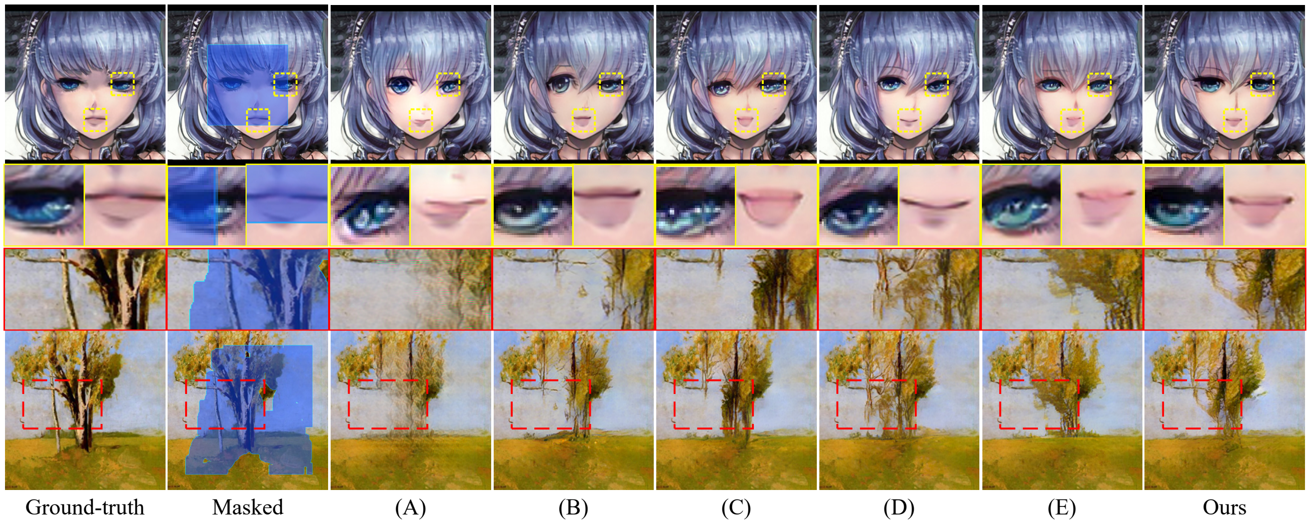


Fig. 21 Examples from the ablation study with (A) GRIG without the forged-patch discriminator, (B) GRIG without the projected discriminator, (C) GRIG's forged-patch discriminator replaced by PatchGAN's discriminator, (D) GRIG's forged-patch discriminator replaced by SM-PatchGAN's discriminator, (E) GRIG with Transformer blocks replaced by down-sampling and up-sampling blocks, and full GRIG (ours).

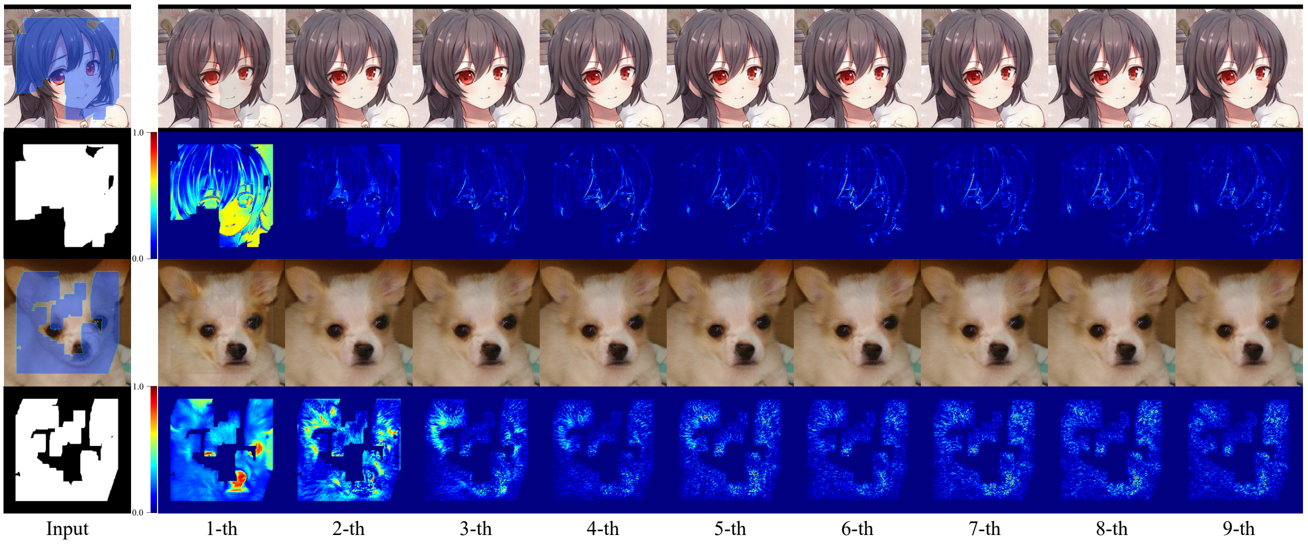


Fig. 22 Inpainting performance at each iterative reasoning step. For each group: Above-left: masked image. Below-left: input binary mask. Above-right: inpainted images. Below–right: heatmaps of residual outputs Δ_t. Colors red–blue in heatmaps represent higher–lower change for the corresponding pixel.

alignment between generated pixels and known pixels may be influenced when we replace our forged-patch discriminator with a Patch-GAN discriminator, as shown in Fig. 21(C). When we replace our forged-patch discriminator with an SM-PatchGAN discriminator, it can create plausible content, but consistency with known areas is poor, as seen in Fig. 21(D). After replacing Transformer blocks with CNN-based blocks, the trained model excels at inpainting texture and detailed contents, but may not be good at capturing structure information, as shown in Fig. 21(E). GRIG shows the best performance on both quantitative and qualitative measures.

Table 6 FID scores for models trained with various numbers of iterative reasoning steps T. Results were evaluated on 50–60% mask ratios. **Bold** indicates best results.

Dataset	$T = 1$	$T = 3$	$T = 5$	$T = 7$	$T = 9$
CHASE	62.76	55.84	56.58	**53.39**	57.84
Anime	69.55	**65.05**	68.05	66.50	69.20
Dog	63.64	**58.49**	59.66	62.09	61.47
Art	78.40	**77.32**	78.04	78.29	78.23

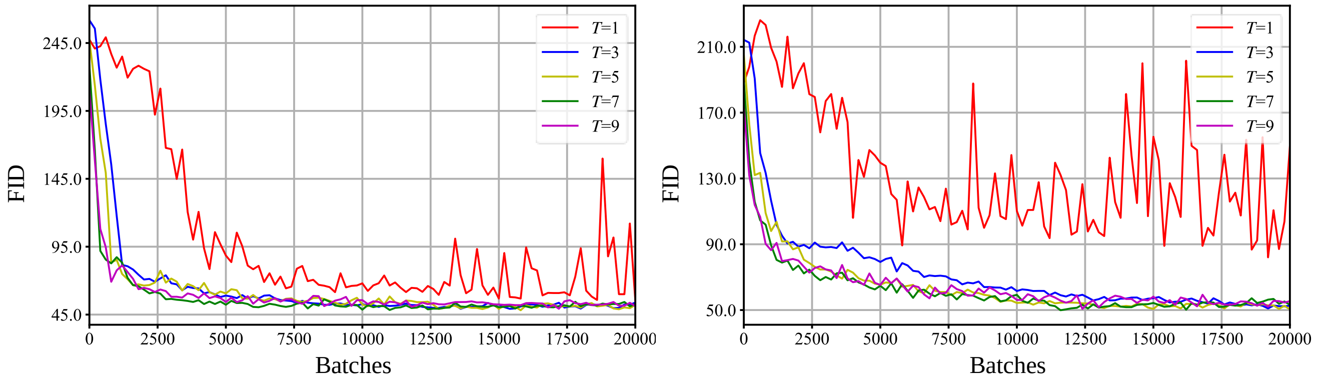


Fig. 23 Comparisons of FID scores for each training iteration on Anime (left) and Dog (right) datasets. Results were evaluated on a fixed center 25% rectangular mask.

4.6.2 Number of iterative reasoning steps

To evaluate the effectiveness of the iterative reasoning in GRIG, we varied the number of iterative reasoning steps T and tested corresponding FID scores on 50–60% mask ratios: see Table 6. Each test used the same number of iterative reasoning steps as the corresponding training phase. Compared to models trained for $T = 1$, models trained for $T > 1$ have large performance gains. For example, on the CHASE dataset, the model trained on $T = 3$ has 6.92 lower FID score than that trained for $T = 1$ (55.84 vs 62.76). When $T > 5$, the performance gains saturate or decrease to some extent, but the inpainting performance is still better than for $T = 1$ in most cases. The results indicate that GRIG can produce satisfactory inpainting outcomes in the early steps, while the residual offsets may fluctuate in subsequent steps, potentially leading to variations in inpainting quality. However, GRIG effectively balances the number of steps and the improvement in inpainting quality, achieving superior performance in the data-efficient image inpainting task. In this paper, we used $T = 3$ to strike a balance between computational cost and visual quality. Fig. 22 visualizes the residual output Δ_t for each step t for the model trained with $T = 3$. The masked images were gradually inpainted. The model prioritizes semantic features in the early steps and fine details in the later steps.

Fig. 23 shows an evaluation of GRIG on a fixed center 25% rectangular mask for models trained with $T = 1, 3, 5, 7, 9$, respectively. The FID scores on Anime and Dog datasets show that models trained with more iterative reasoning steps T converge faster than those with fewer T, and models trained with $T = 1$ do not readily converge. Specifically, models trained with $T > 1$ converge for around $10,000$ image batches, whereas models trained with $T = 1$ are far from convergence and fluctuate drastically even after $10,000$ image batches. This shows that our framework can effectively help networks to converge faster.

5 Conclusions, limitations, and future work

We have taken a first step toward solving data-efficient image inpainting in this paper. By introducing iterative residual reasoning with decoupled image-level and patch-level discriminators, we have presented a novel data-efficient generative residual image inpainting framework. The proposed generator effectively utilizes CNNs for feature extraction and Transformers for global reasoning. To assist the generative network in learning image fine details, a forged-patch discriminator was introduced. Furthermore, we have established new state-of-the-art performance on multiple small-scale datasets, and extensive experiments have demonstrated the efficacy of the proposed method.

Our method has some limitations. The approach can effectively perform high-fidelity image inpainting on small-scale datasets. However, GRIG cannot directly utilize conditional information for guidance-based image inpainting. Introducing a more sophisticated scheme or module to guide the inpainting process would be interesting for controllable small-scale image completion. Moreover, GRIG is not specialized in diverse image inpainting. Using a mapping network to embed random style codes into the generator could be a good solution for diversity of data-efficient image inpainting.

Availability of data and materials

The data involved in this study are all public data, which can be downloaded through public channels.

Authors' contributions

Wanglong Lu: Writing—Original Draft, Methodology, Validation, Software. Xianta Jiang: Writing—Review & Editing, Methodology, Supervision. Xiaogang Jin: Writing—Review

& Editing, Methodology. Yong-Liang Yang: Writing—Review & Editing, Conceptualization, Analysis. Minglun Gong: Writing—Review & Editing, Conceptualization, Analysis. Kaijie Shi: Data Curation, Investigation. Tao Wang: Data Curation, Investigation, Conceptualization. Hanli Zhao: Writing—Review & Editing, Conceptualization, Methodology, Analysis, Supervision.

Acknowledgements

This work was supported by the Zhejiang Provincial Natural Science Foundation of China (Grant No. LZ21F020001) and the Basic Scientific Research Program of Wenzhou (Grant No. S20220018).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Tian H, Zhang L, Li S, Yao M, Pan G. Pyramid-VAE-GAN: Transferring hierarchical latent variables for image inpainting. *Computational Visual Media*, 2023, 9(4): 827–841.

[2] Zeng X, Wu Z, Peng X, Qiao Y. Joint 3D facial shape reconstruction and texture completion from a single image. *Computational Visual Media*, 2022, 8(2): 239–256.

[3] Wu H, Fu K, Zhao Y, Song H, Li J. Joint self-supervised and reference-guided learning for depth inpainting. *Computational Visual Media*, 2022, 8(4): 597–612.

[4] Wu Z, Guo J, Zhuang C, Xiao J, Yan DM, Zhang X. Joint specular highlight detection and removal in single images via Unet-Transformer. *Computational Visual Media*, 2023, 9(1): 141–154.

[5] Wan Z, Zhang B, Chen D, Zhang P, Chen D, Liao J, Wen F. Bringing Old Photos Back to Life. In *CVPR*, 2020, 2747–2757.

[6] Bian X, Wang C, Quan W, Ye J, Zhang X, Yan DM. Scene text removal via cascaded text stroke detection and erasing. *Computational Visual Media*, 2022, 8(2): 273–287.

[7] Suvorov R, Logacheva E, Mashikhin A, Remizova A, Ashukha A, Silvestrov A, Kong N, Goka H, Park K, Lempitsky V. Resolution-robust Large Mask Inpainting with Fourier Convolutions. In *WACV*, 2022, 3172–3182.

[8] Li W, Lin Z, Zhou K, Qi L, Wang Y, Jia J. MAT: Mask-Aware Transformer for Large Hole Image Inpainting. In *CVPR*, 2022, 10748–10758.

[9] Wang T, Zhang K, Chen X, Luo W, Deng J, Lu T, Cao X, Liu W, Li H, Zafeiriou S. A Survey of Deep Face Restoration: Denoise, Super-Resolution, Deblur, Artifact Removal. *arXiv preprint arXiv:2211.02831*, 2022.

[10] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is All you Need. In *NeurIPS*, volume 30, 2017, 15 pages.

[11] Arjovsky M, Bottou L. Towards Principled Methods for Training Generative Adversarial Networks. In *ICLR*, 2017, 17 pages.

[12] Zhao S, Cui J, Sheng Y, Dong Y, Liang X, Chang EI, Xu Y. Large Scale Image Completion via Co-Modulated Generative Adversarial Networks. In *ICLR*, 2021, 25 pages.

[13] Yu J, Lin Z, Yang J, Shen X, Lu X, Huang T. Free-Form Image Inpainting With Gated Convolution. In *ICCV*, 2019, 4470–4479.

[14] Zeng Y, Lin Z, Yang J, Zhang J, Shechtman E, Lu H. High-Resolution Image Inpainting with Iterative Confidence Feedback and Guided Upsampling. In *ECCV*, 2020, 1–17.

[15] Li J, Wang N, Zhang L, Du B, Tao D. Recurrent Feature Reasoning for Image Inpainting. In *CVPR*, 2020, 7757–7765.

[16] Ojha U, Li Y, Lu C, Efros AA, Lee YJ, Shechtman E, Zhang R. Few-shot Image Generation via Cross-domain Correspondence. In *CVPR*, 2021, 10738–10747.

[17] Liu B, Zhu Y, Song K, Elgammal A. Towards Faster and Stabilized GAN Training for High-fidelity Few-shot Image Synthesis. In *ICLR*, 2021, 13 pages.

[18] Sauer A, Chitta K, Müller J, Geiger A. Projected GANs Converge Faster. In *NeurIPS*, volume 34, 2021, 17480–17492.

[19] Du Y, Li S, Tenenbaum J, Mordatch I. Learning Iterative Reasoning through Energy Minimization. In *ICML*, volume 162, 2022, 5570–5582.

[20] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In *CVPR*, 2016, 770–778.

[21] Wang T, Tao G, Lu W, Zhang K, Luo W, Zhang X, Lu T. Restoring vision in hazy weather with hierarchical contrastive learning. *Pattern Recognition*, 2024, 145: 109956.

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

[23] Bertalmio M, Sapiro G, Caselles V, Ballester C. Image Inpainting. In *SIGGRAPH*, 2000, 417–424.

[24] Levin, Zomet, Weiss. Learning how to inpaint from global image statistics. In *ICCV*, volume 1, 2003, 305–312.

[25] Telea A. An Image Inpainting Technique Based on the Fast Marching Method. *Journal of Graphics Tools*, 2004, 9(1): 23–34.

[26] Kwatra V, Essa I, Bobick A, Kwatra N. Texture Optimization for Example-Based Synthesis. *TOG*, 2005, 24(3): 795–802.

[27] Zhao H, Guo H, Jin X, Shen J, Mao X, Liu J. Parallel and efficient approximate nearest patch matching for image editing applications. *Neurocomputing*, 2018, 305: 39–50.

[28] Barnes C, Shechtman E, Finkelstein A, Goldman DB. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *TOG*, 2009, 28(3): Article 24.

[29] Simakov D, Caspi Y, Shechtman E, Irani M. Summarizing visual data using bidirectional similarity. In *CVPR*, 2008, 1–8.

[30] Ding D, Ram S, Rodríguez JJ. Image Inpainting Using Nonlocal Texture Matching and Nonlinear Filtering. *TIP*, 2019, 28(4): 1705–1719.

[31] Ren JS, Xu L, Yan Q, Sun W. Shepard Convolutional Neural Networks. In *NeurIPS*, volume 28, 2015, 901–909.

[32] Pathak D, Krähenbühl P, Donahue J, Darrell T, Efros AA. Context Encoders: Feature Learning by Inpainting. In *CVPR*, 2016, 2536–2544.

[33] Wang Y, Tao X, Qi X, Shen X, Jia J. Image Inpainting via Generative Multi-column Convolutional Neural Networks. In *NeurIPS*, 2018, 331–340.

[34] Yan Z, Li X, Li M, Zuo W, Shan S. Shift-Net: Image Inpainting via Deep Feature Rearrangement. In *ECCV*, 2018, 3–19.

[35] Zeng Y, Fu J, Chao H, Guo B. Learning Pyramid-Context Encoder Network for High-Quality Image Inpainting. In *CVPR*, 2019, 1486–1494.

[36] Iizuka S, Simo-Serra E, Ishikawa H. Globally and Locally Consistent Image Completion. *TOG*, 2017, 36(4): Article 107.

[37] Liu H, Jiang B, Xiao Y, Yang C. Coherent Semantic Attention for Image Inpainting. In *ICCV*, 2019, 4169–4178.

[38] Xie C, Liu S, Li C, Cheng MM, Zuo W, Liu X, Wen S, Ding E. Image Inpainting With Learnable Bidirectional Attention Maps. In *ICCV*, 2019, 8857–8866.

[39] Yi Z, Tang Q, Azizi S, Jang D, Xu Z. Contextual Residual Aggregation for Ultra High-Resolution Image Inpainting. In *CVPR*, 2020, 7508–7517.

[40] Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS. Generative Image Inpainting With Contextual Attention. In *CVPR*, 2018, 5505–5514.

[41] Wang M, Lu W, Lyu J, Shi K, Zhao H. Generative image inpainting with enhanced gated convolution and Transformers. *Displays*, 2022, 75: 102321.

[42] Liu G, Reda FA, Shih KJ, Wang TC, Tao A, Catanzaro B. Image Inpainting for Irregular Holes Using Partial Convolutions. In *ECCV*, 2018, 89–105.

[43] Jo Y, Park J. SC-FEGAN: Face Editing Generative Adversarial Network With User's Sketch and Color. In *ICCV*, 2019, 1745–1753.

[44] Dong Q, Cao C, Fu Y. Incremental Transformer Structure Enhanced Image Inpainting With Masking Positional Encoding. In *CVPR*, 2022, 11358–11368.

[45] Xiong W, Yu J, Lin Z, Yang J, Lu X, Barnes C, Luo J. Foreground-Aware Image Inpainting. In *CVPR*, 2019, 5833–5841.

[46] Ren Y, Yu X, Zhang R, Li TH, Liu S, Li G. StructureFlow: Image Inpainting via Structure-aware Appearance Flow. In *ICCV*, 2019, 181–190.

[47] Lu W, Zhao H, Jiang X, Jin X, Yang Y, Wang M, Lyu J, Shi K. Do Inpainting Yourself: Generative Facial Inpainting Guided by Exemplars. *arXiv preprint arXiv:2202.06358*, 2022.

[48] Zeng Y, Fu J, Chao H, Guo B. Aggregated Contextual Transformations for High-Resolution Image Inpainting. *TVCG*, 2023, 29(7): 3266–3280.

[49] Zheng C, Cham TJ, Cai J, Phung D. Bridging Global Context Interactions for High-Fidelity Image Completion. In *CVPR*, 2022, 11512–11522.

[50] Zhao L, Mo Q, Lin S, Wang Z, Zuo Z, Chen H, Xing W, Lu D. UCTGAN: Diverse Image Inpainting Based on Unsupervised Cross-Space Translation. In *CVPR*, 2020, 5740–5749.

[51] Wan Z, Zhang J, Chen D, Liao J. High-Fidelity Pluralistic Image Completion with Transformers. In *ICCV*, 2021, 4672–4681.

[52] Liu Q, Tan Z, Chen D, Chu Q, Dai X, Chen Y, Liu M, Yuan L, Yu N. Reduce Information Loss in Transformers for Pluralistic Image Inpainting. In *CVPR*, 2022, 11347–11357.

[53] Zhang H, Hu Z, Luo C, Zuo W, Wang M. Semantic Image Inpainting with Progressive Generative Networks. In *ACM MM*, 2018, 1939–1947.

[54] Li J, He F, Zhang L, Du B, Tao D. Progressive Reconstruction of Visual Structure for Image Inpainting. In *ICCV*, 2019, 5961–5970.

[55] Guo Z, Chen Z, Yu T, Chen J, Liu S. Progressive Image Inpainting with Full-Resolution Residual Network. In *ACM MM*, 2019, 2496–2504.

[56] Jain J, Zhou Y, Yu N, Shi H. Keys To Better Image Inpainting: Structure and Texture Go Hand in Hand. In *WACV*, 2023, 208–217.

[57] Hur J, Roth S. Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation. In *CVPR*, 2019, 5747–5756.

[58] Chen X, Wang X, Zhou J, Qiao Y, Dong C. Activating More Pixels in Image Super-Resolution Transformer. In *CVPR*, 2023, 22367–22377.

[59] Zhang D, Huang F, Liu S, Wang X, Jin Z. SwinFIR: Revisiting the SwinIR with Fast Fourier Convolution and Improved Training for Image Super-Resolution. *arXiv preprint arXiv:2208.11247*, 2022.

[60] Zamir SW, Arora A, Khan S, Hayat M, Khan FS, Yang M. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *CVPR*, 2022, 5718–5729.

[61] Miyato T, Kataoka T, Koyama M, Yoshida Y. Spectral Normalization for Generative Adversarial Networks. In *ICLR*, 2018, 26 pages.

[62] Tan M, Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *ICML*, volume 97, 2019, 6105–6114.

[63] Yu F, Koltun V. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*, 2016, 13 pages.

[64] Isola P, Zhu JY, Zhou T, Efros AA. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*, 2017, 5967–5976.

[65] Texler O, Futschik D, Kučera M, Jamriška O, Šárka Sochorová, Chai M, Tulyakov S, Sýkora D. Interactive Video Stylization Using Few-Shot Patch-Based Training. *TOG*, 2020, 39(4): 73.

[66] Wang W, Bao J, Zhou W, Chen D, Chen D, Yuan L, Li H. SinDiffusion: Learning a Diffusion Model from a Single Natural Image. *arXiv preprint arXiv:2211.12445*, 2022.

[67] Zhang R, Isola P, Efros AA, Shechtman E, Wang O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018, 586–595.

[68] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015, 14 pages.

[69] MMEditing Contributors. MMEditing: OpenMMLab Image and Video Editing Toolbox. https://github.com/open-mmlab/mmediting, 2022, [Online; accessed 14-Feb-2022].

[70] Fraz MM, Remagnino P, Hoppe A, Uyyanonvara B, Rudnicka AR, Owen CG, Barman SA. An Ensemble Classification-Based Approach Applied to Retinal Blood Vessel Segmentation. *IEEE Transactions on Biomedical Engineering*, 2012, 59(9): 2538–2548.

[71] Si Z, Zhu SC. Learning Hybrid Image Templates (HIT) by Information Projection. *TPAMI*, 2012, 34(7): 1354–1367.

[72] Karras T, Aila T, Laine S, Lehtinen J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *ICLR*, 2018, 26 pages.

[73] Karras T, Laine S, Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks. *TPAMI*, 2021, 43(12): 4217–4228.

[74] Doersch C, Singh S, Gupta A, Sivic J, Efros AA. What Makes Paris Look like Paris? In *SIGGRAPH*, 2012, 101:1–101:9.

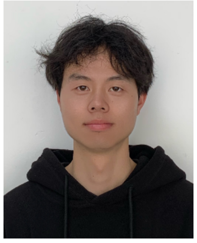
[75] Zhou B, Lapedriza A, Khosla A, Oliva A, Torralba A. Places: A 10 Million Image Database for Scene Recognition. *TPAMI*, 2018, 40(6): 1452–1464.

[76] Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, volume 30, 2017, 6629–6640.

[77] Zhao S, Liu Z, Lin J, Zhu JY, Han S. Differentiable Augmentation for Data-Efficient GAN Training. In *NeurIPS*, volume 33, 2020, 7559–7570.

[78] Chi L, Jiang B, Mu Y. Fast Fourier Convolution. In H Larochelle, M Ranzato, R Hadsell, M Balcan, H Lin, editors, *NeurIPS*, volume 33, 2020, 4479–4488.

Author biography

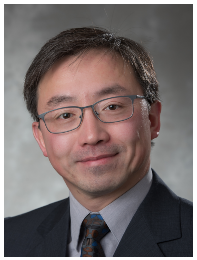


Wanglong Lu is a Ph.D. student in the Department of Computer Science at Memorial University of Newfoundland, Canada. He received his B.Sc. degree in digital media technology from the Communication University of Zhejiang, China, in 2018, and his M.Sc. degree in computer software and theory from Wenzhou University, China, in 2021. His research interests include image inpainting, image editing, and image recognition.

Xianta Jiang is currently an Assistant Professor in the Department of Computer Science at Memorial University of Newfoundland,. He received a dual Ph.D. in computer science from Simon Fraser University and Zhejiang University (2015). He then took a post-doctoral fellowship in Engineering Science, Simon Fraser University and worked as a senior research associate in the Department of Surgery at the University of Alberta. His research interests include machine intelligence, HCI, and wearables.

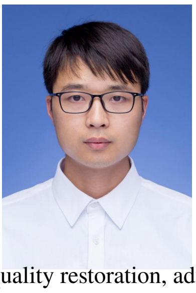
Xiaogang Jin is a Professor in the State Key Laboratory of CAD&CG, Zhejiang University. He received a B.Sc. degree in computer science and M.Sc. and Ph.D degrees in applied mathematics from Zhejiang University, P.R. China, in 1989, 1992, and 1995, respectively. His current research interests include image processing, digital humans, traffic simulation, collective behavior simulation, cloth animation, virtual try-on, digital faces, implicit surface modeling and applications, creative modeling, computer-generated marbling, sketch-based modeling, and virtual reality. He received an ACM Recognition of Service Award in 2015 and two Best Paper Awards from CASA 2017 and CASA 2018. He is a member of the IEEE and ACM.

Yong-Liang Yang is a Senior Lecturer in the Department of Computer Science at the University of Bath, United Kingdom. He received B.Sc. and Ph.D. degrees in computer science from Tsinghua University, China. His research area is broadly in visual computing, with particular interests in shape modeling, computational design, and interactive techniques.



Minglun Gong is a Professor and Director of the School of Computer Science, University of Guelph. Before he moved to Guelph in 2019, he was a Professor and Head of the Department of Computer Science, Memorial University of Newfoundland. He obtained his Ph.D. from the University of Alberta in 2003, his M.Sc. from the Tsinghua University in 1997, and his B.Eng. from Harbin Engineering University in 1994.

Kaijie Shi is a Ph.D. student in the Department of Computer Science at Memorial University of Newfoundland, Canada. He received his B.Sc. degree in geographic information science from Nanjing Xiaozhuang University, China, in 2019, and his M.Sc. degree in computer science and technology from Wenzhou University, China, in 2022. His research interests include computer graphics and computer vision.



Tao Wang is a Ph.D. student in the Department of Computer Science and Technology, Nanjing University, China. He received a B.Sc. degree in information and computing science from Hainan Normal University, China, in 2018. His research interests include several topics in computer vision and machine learning, such as object tracking, image/video quality restoration, adversarial learning, image-to-image translation and reinforcement learning.

Hanli Zhao is a Professor in the Key Laboratory of Intelligent Informatics of Safety & Emergency of Zhejiang Province, Wenzhou University, China. He received his B.Sc. degree in software engineering from Sichuan University, China, in 2004, and his Ph.D. degree in computer science from the State Key Lab of CAD&CG, Zhejiang University in 2009. His current research interests include computer graphics, computer vision, pattern recognition, medical image analysis, and deep learning.