# LIGHTWEIGHT LONG-RANGE GENERATIVE ADVERSARIAL NETWORKS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In this paper, we introduce novel lightweight generative adversarial networks, which can effectively capture long-range dependencies in the image generation process, and produce high-quality results with a much simpler architecture. To achieve this, we first introduce a long-range module, allowing the network to dynamically adjust the number of focused sampling pixels and to also augment sampling locations. Thus, it can break the limitation of the fixed geometric structure of the convolution operator, and capture long-range dependencies in both spatial and channel-wise directions. Also, the proposed long-range module can highlight negative relations between pixels, working as a regularization to stabilize training. Furthermore, we propose a new generation strategy through which we introduce metadata into the image generation process to provide basic information about target images, which can stabilize and speed up the training process. Our novel long-range module only introduces few additional parameters and is easily inserted into existing models to capture long-range dependencies. Extensive experiments demonstrate the competitive performance of our method with a lightweight architecture.

## 1 INTRODUCTION

Generating realistic and diverse samples from high-dimensional data distributions has made much progress with the emergence of autoregressive models (Oord et al., 2016; Van den Oord et al., 2016), variational autoencoders (Kingma & Welling, 2013), and generative adversarial networks (GANs) (Goodfellow et al., 2014), which greatly boost various research areas, including speech synthesis (Donahue et al., 2018; Bińkowski et al., 2019; Engel et al., 2019), image (Karras et al., 2017; Zhang et al., 2019; Karras et al., 2019) and video generation (Clark et al., 2019), text-to-image generation (Zhang et al., 2018; Xu et al., 2018; Li et al., 2019), text-guided image manipulation (Dong et al., 2017; Nam et al., 2018; Li et al., 2020a), and image-to-image translation (Zhu et al., 2017; Park et al., 2019; Li et al., 2020b).

In fact, most of the aforementioned generative approaches depend heavily on the convolution operator to model dependencies across different regions. However, the convolution operator has a fixed geometric structure with a local receptive field, and thus long-range dependencies on non-neighboring locations can only be captured by passing through several convolution layers. Unfortunately, increasing the number of layers is undesirable for devices with limited memory storage (e.g., mobile phones), because (1) it makes the architecture more complex with numerous parameters, (2) it can cause much trouble for optimization algorithms to effectively coordinate multiple layers to capture long-range dependencies, and (3) it may fail to keep a globally semantic consistency (e.g., structure of objects) due to its fixed geometric structure.

To address the above problems, it is desirable to have a module with an adjustable receptive field and augmented sampling locations to effectively capture long-range dependencies. To achieve this, we propose a novel long-range module, which enables free-form deformation of the sampling grid and allows the network to dynamically adjust the number of focused sampling pixels, shown in Fig. 1. By adopting our module, the network is able to capture long-range dependencies between non-neighboring pixels without greatly increasing the network layers, enabling a possibility to install the model into memory-limited devices. The proposed module only introduces few additional parameters, and can be readily inserted into existing models to capture long-range dependencies.

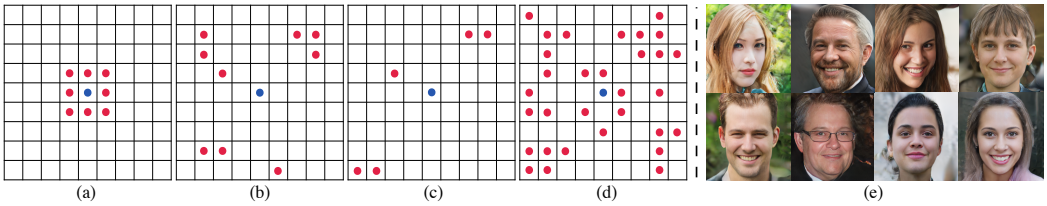



Figure 1: (a): local dependencies of the blue point captured by a $3 \times 3$ regular convolution operator; (b): long-range dependencies of the blue point highlighted in distant locations and captured by our long-range module; (c) and (d) show that our long-range module can flexibly adjust the number of focused sampling locations, related to the blue point, to capture long-range dependencies; and (e) presents eight sample results generated by our lightweight long-range network at $256 \times 256$.

Furthermore, we propose a novel generation strategy through which we introduce metadata into the image generation process, where the metadata can provide basic information for target images (e.g., the category and basic texture of objects), stabilize the training process, and thus free the generator from a complex architecture and speed up the training process.

Finally, we evaluate our model on the FFHQ (Karras et al., 2019), CUB bird (Wah et al., 2011), and ImageNet (Russakovsky et al., 2015) datasets, which demonstrates that our method can produce high-quality images with a great efficiency.

## 2 Related Work

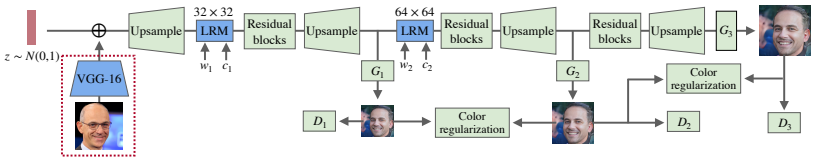
**Generative adversarial networks** have achieved much success in generating realistic images (Karras et al., 2017; 2019; 2020), which is widely adopted in various tasks, including text-to-image generation (Zhang et al., 2017; He et al., 2019; Li et al., 2019), image-to-image translation (Chen & Koltun, 2017; Isola et al., 2017; Wang et al., 2018a; Li et al., 2020b), text-guided image manipulation (Dong et al., 2017; Nam et al., 2018; Li et al., 2020a), and super resolution (Sønderby et al., 2016; Ledig et al., 2017). However, to produce high-quality results, these methods usually have a rather complex architecture with many parameters, and require a quite long time for optimization and inference, which is especially undesirable for memory-limited devices, such as mobile phones.

**Long-range dependencies** play a critical role in deep neural networks. To capture long-range dependencies, recurrent operations (Rumelhart et al., 1986; Hochreiter & Schmidhuber, 1997) are adopted for sequential data in language, and large receptive fields formed by deep stacked convolution layers are implemented for pixel data in images. Recently, self-attention (Vaswani et al., 2017) is proposed to discard costly recurrent operations by attending to all positions in a sequence, and is widely adopted in various tasks, including machine translation (Vaswani et al., 2017), video classification (Wang et al., 2018b), and image generation (Zhang et al., 2019). However, the generation of self-attention relies heavily on the implementation of the softmax function, and so almost all values in an attention map are greater than $0$. Thus, attention can only produce positive relations between pixels in an image. Unfortunately, not all pixels have a positive effect on others, and it is more reasonable to keep the negative effect instead of converting all relations into positive ones.

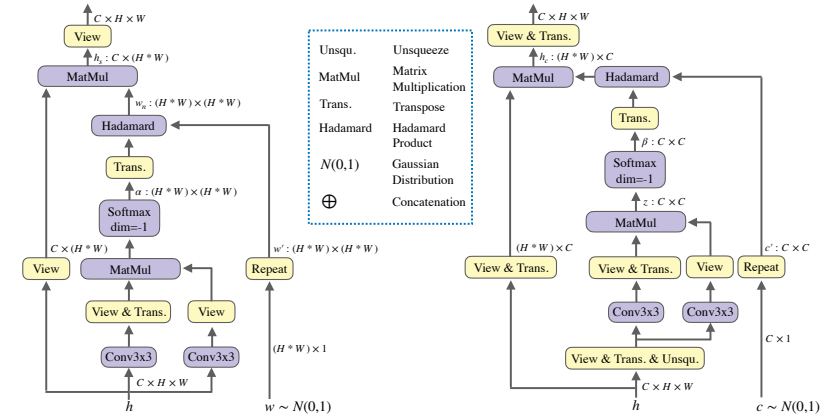
**Receptive field.** How to increase the receptive field of the convolution operator has drawn much attention. One complementary technique is to use dilated convolutions (Chen et al., 2014; Yu & Koltun, 2015). With dilated convolutions, the number of parameters does not change, but the receptive field grows exponentially if the number of parameters grows linearly in successive layers. The other is to use deformable convolution (Dai et al., 2017; Zhu et al., 2019), which learns the offset to achieve a deformation of the sampling grid. However, the number of sampling locations are still fixed, and the weights of deformable convolution operators are shared across all different regions.

## 3 Lightweight Long-Range Generative Adversarial Networks

Given a random noise $z$ sampled from the Gaussian distribution $\mathcal{N}(0, 1)$, our method aims to generate novel high-quality images, and at the same time, the model should be efficient and small enough




(a) The architecture of our proposed lightweight long-range generative adversarial networks




(b) The long-range module in spatial direction

(c) The long-range module in channel-wise direction

Figure 2: (a): architecture of the proposed lightweight long-range generative adversarial networks; the red dashed box indicates the generation of metadata; (b): implementation of the long-range module in spatial direction; (c): implementation of the long-range module in channel-wise direction.

for memory-limited devices. To achieve this, we propose a novel long-range module and a new generation strategy. The complete architecture is shown in Fig. 2 (a).

### 3.1 Architecture

As shown in Fig. 2 (a), a multi-stage architecture (Zhang et al., 2018; Xu et al., 2018; Li et al., 2019; 2020a) is adopted, where there are multiple generator and discriminator pairs in the model. The reason to have such a design is based on our experimental finding that a model having a lightweight architecture with a few number of parameters is easily prone to break due to the instability of GANs. However, with the implementation of a multi-stage architecture, first, the model distribution generated from coarse low-resolution results has a better probability of intersecting with the real image distribution, enabling a better foundation for higher-quality image generation. Second, low-resolution results can also work as a regularization to constrain the randomness involved in the generation of higher-resolution images, and thus to prevent the mode collapse at higher stages. To build this constraint, color consistency regularization (Zhang et al., 2018) is adopted to keep samples generated from different generators more consistent in color. The regularization $\mathcal{L}_C$ at stage $i$ is defined as:

$$\mathcal{L}_{C_i} = \frac{1}{n}\sum_{j=1}^{n}(\lambda_1 \left\|\mu_{s_i^j} - \mu_{s_{i-1}^j}\right\|_2^2 + \lambda_2 \left\|\sigma_{s_i^j} - \sigma_{s_{i-1}^j}\right\|_F^2), \tag{1}$$

where $i$ is the index of stage with $i > 1$, $n$ is the batch size, $\mu_{s_i^j}$ and $\sigma_{s_i^j}$ are the mean and the covariance for the $j^{th}$ sample generated by the $i^{th}$ generator, $F$ represents the Frobenius norm, and $\lambda_1$ and $\lambda_2$ are hyperparameters.

### 3.2 Long-Range Module

To build a lightweight network for high-quality image generation, it is desirable to have a module that can efficiently capture the long-range dependencies between non-neighboring locations rather than increasing the number of layers in a network. To achieve this, we propose a novel long-range module that can capture the long-range dependencies between distant pixels in both spatial and channel-wise directions, without introducing many additional parameters.

**Long-range module in spatial direction.** As shown in Fig. 2 (b), the long-range module in spatial direction takes two inputs: (1) the hidden feature $h \in \mathbb{R}^{C \times H \times W}$, where $C$, $H$, and $W$ denote the number of channels, the height and the weight of the feature map $h$, respectively, and (2) a learnable weight $w \in \mathbb{R}^{(H*W) \times 1}$, drawn from the Gaussian distribution $\mathcal{N}(0, 1)$, where $w$ is used to perform the linear transformation on a given feature along the spatial direction.

To find the correlation between different spatial locations, we first use the convolution operator to convert the hidden features $h$ into a new semantic space to produce $h_{w1}$ and $h_{w2}$. Then, we change their shape and apply matrix multiplication between them to get a matrix $n \in \mathbb{R}^{(H*W) \times (H*W)}$, denoted as $n = h_1^T * h_2$, where the matrix $n$ contains all possible pairs of different spatial locations. We further normalize it by the softmax function on the last dimension, to get a correlation value $\alpha$,

$$\alpha_{i,j} = \frac{\exp(n_{i,j})}{\sum_{l=1}^{H*W} \exp(n_{i,l})}, \tag{2}$$

where $\alpha_{i,j}$ denotes the extent of correlation between spatial locations $i$ and $j$, regardless of the relations being positive or negative. Meanwhile, we repeat the learnable weight $w$ on the column dimension to get $w' \in \mathbb{R}^{(H*W) \times (H*W)}$, where the following columns have the same values as the first one, reducing the number of learnable parameters. Then, the spatial-relation-aware weight $w_n$ can be obtained by applying matrix multiplication $\times$ on $\alpha$ and $w'$, denoted $w_n = \alpha^T \times w'$, where $w_{n_{i,j}}$ represents a weight scaled by the extent of relations between spatial locations $i$ and $j$. Finally, the hidden feature $h_s$ capturing long-range dependencies in spatial direction can be obtained by applying matrix multiplication on $h$ with a new shape and $w_n$.

**Long-range module in channel-wise direction.** As shown in Fig. 2 (c), the long-range module in channel-wise direction takes two inputs: the hidden feature $h$, and (2) a learnable weight $c \in \mathbb{R}^{C \times 1}$, drawn from the Gaussian distribution $\mathcal{N}(0, 1)$, where $c$ is used to perform the linear transformation on a given feature along the channel-wise direction.

We first use the convolution operator to convert $h$ into a new semantic space to produce $h_{c1}$ and $h_{c2}$. Then, we change their shape, followed by a matrix multiplication to get $z \in \mathbb{R}^{C \times C}$, denoted $z = h_{c1}^T * h_{c2}$. Finally, we normalizes $z$ on the last dimension by applying the softmax function, obtaining the correlation matrix $\beta$,

$$\beta_{i,j} = \frac{\exp(z_{i,j})}{\sum_{l=1}^{C} \exp(z_{i,l})}, \tag{3}$$

where $\beta_{i,j}$ denotes the extent of positive or negative correlation between channels $i$ and $j$. We repeat $c$ on the column dimension to obtain $c' \in \mathbb{R}^{C \times C}$, and each column has the same values. Then, the channel-relation-aware weight $c_n$ is obtained by applying matrix multiplication $\times$ on $\beta$ and $c'$, denoted $c_n = \beta^T \times c'$, where $c_{n_{i,j}}$ represents a weight scaled by the extent of relations between channels $i$ and $j$. Thus, the feature $h_c$ capturing the long-range dependencies in channel-wise direction is obtained by applying matrix multiplication on $h$ with a new shape and $c_n$.

**Why are the long-range dependencies important in the image generation task?** There exist potential relations between pixels in neighboring and non-neighboring locations. These relations can work as cues to help the generator to draw images, where fine-grained details at every location are carefully coordinated with details in distant regions of the image, to keep a global semantic consistency, e.g., when a generator tries to produce an image about a cat, it is better for it to take all related pixels into account to draw different parts (e.g., eyes and mouth) at reasonable locations.

**Why does the long-range module work better?** (1) Comparison with convolution operator: the convolution operator has a fixed geometric structure with local receptive fields, and long-range dependencies on non-neighboring locations can only be captured by passing through several convolution layers, which prevents building a lightweight model with few parameters. One possible solution is to increase the size of the convolution kernels, but it loses the computational efficiency benefited from the local convolution structure. However, our long-range module is able to work as a complement to the convolution operator. At the cost of increasing only a few number of parameters, it helps the model to capture long-range dependencies across image regions. (2) Comparison with self-attention: self-attention is based heavily on the implementation of the softmax function, and thus almost all its values are greater than 0. This means that self-attention utilizes the scales of positive values to highlight or ignore different image regions, i.e., giving high (or low) **positive** weights to

important (or unimportant) regions. However, not all image regions have a positive impact on others, and some negative relations are also vital in the image generation process, especially those negative relations can work as a regularization to stabilize the training process and prevent mode collapse. To keep both negative and positive effects, our long-rang module only highlights or ignores the **relations** between different image regions, rather than the **actual values** of a hidden feature. More specifically, $\alpha$ (Eq. 2) and $\beta$ (Eq. 3) can be treated as scaling weights to highlight or ignore negative and positive relations contained in two learnable $w$ and $c$, and then $w$ and $c$ are used to achieve a transformation on the hidden features.

### 3.3 Generation Strategy with Metadata

To speed up the training process, we suggest to incorporate metadata into the model to provide the generator with basic information about target images, where this information may contain cues for the desired real image distribution that the generator finally has to generate, helping it to know what kinds of objects to synthesize in advance and speeding up the training (Fig. 2 (a), red dashed box).

Also, to prevent the provided metadata enforcing the model to achieve the identity transformation from the meta-image, we use the global image features as the metadata, extracted from the given meta-image using a deep layer of the pretrained VGG-16 network (Simonyan & Zisserman, 2014). Thus, the metadata only keeps summarized spatial information. The reason to have such a design is mainly because normal spatial features contain too many details about target images, such as color, shape, pose, and location of objects. Therefore, according to averaging all spatial features in each channel, those fine details can be filtered, and only elementary information is kept, ensuring a good diversity of the model.

### 3.4 Objective Functions

We train the generator and the discriminator alternatively by minimizing the generator loss $\mathcal{L}_G$ and the discriminator loss $\mathcal{L}_D$.

**Generator objective.** The complete generator objective has an unconditional adversarial loss and a color consistency regularization $\mathcal{L}_C$ (Eq. 1), where the adversarial loss encourages the generator to produce realistic fake images:

$$\mathcal{L}_G = \sum_{k=1}^{K}(-\frac{1}{2}E_{I'_k\sim P_{G_k}}[\log(D_k(I'_k))]) + \sum_{i=2}^{K}(\lambda_3\mathcal{L}_{C_i}), \tag{4}$$

where $K$ is the number of stages, $I'_k$ is the synthetic images sampled from the model distribution $P_{G_k}$ at stage $k$, $D_k$ is the discriminator at stage $k$, and $\lambda_3$ is a hyperparameter.

**Discriminator objective.** The final discriminator objective is defined as:

$$\mathcal{L}_D = \sum_{k=1}^{K}(-\frac{1}{2}E_{I_k\sim P_{\text{data}}}[\log(D_k(I_k))] - \frac{1}{2}E_{I'_k\sim P_{G_k}}[\log(1-D_k(I'_k))]), \tag{5}$$

where $I_k$ is the real images sampled from the true image distribution $P_{\text{data}}$ at stage $k$.

## 4 Experiments

To evaluate our method, we conduct extensive experiments on the FFHQ (Karras et al., 2019), CUB bird (Wah et al., 2011), and ImageNet (Russakovsky et al., 2015) datasets, comparing with two approaches, PGGAN (Karras et al., 2017) and SAGAN (Zhang et al., 2019), where PGGAN is able to produce high-quality images with a relatively simpler architecture compared with StyleGAN (Karras et al., 2019) and StyleGAN2 (Karras et al., 2020), and SAGAN implements self-attention to capture long-range dependencies. Note that we do not compare our method with StyleGAN and StyleGAN2, because both approaches are based on PGGAN but have a more complex architecture with a larger number of parameters. However, the purpose of our method is to achieve a lightweight architecture with a small number of parameters, which allows the network to be implemented in memory-shortage devices.

| Method | FFHQ | CUB | ImageNet Church | IT(s) | NoP-G | NoP-D |
|---|---|---|---|---|---|---|
| PGGAN | 18.41 | 38.73 | 72.13 | 17.24 | 23.3M | 23.3M |
| SAGAN | 79.81 | 70.93 | 84.41 | 1.89 | 13.3M | 51.6M |
| **Ours** | 23.97 | 24.85 | 56.50 | 0.83 | 7.0M | 7.6M |
| Ours w/o Meta | 35.35 | 26.47 | 68.70 | - | - | - |
| Ours w/o LRM | 156.59 | 42.25 | 177.28 | - | - | - |
| Ours w/ Residual | 43.83 | 43.21 | 90.41 | - | - | - |
| Ours w/ SA | 187.61 | 35.76 | 84.47 | - | - | - |

Table 1: Quantitative comparison: Fréchet inception distance (FID), inference time (IT) for generating 100 new results, and number of parameters in the generator (NoP-G) and the discriminator (NoP-D) of these approaches and our method on FFHQ, CUB, and ImageNet Church. "Ours w/o Meta" denotes without the provision of metadata. "Ours w/o LRM" denotes without implementing the long-range module. "Ours w/ Residual" denotes using the residual blocks to replace our long-range module. "Ours w/ SA" denotes using self-attention instead of our long-range module. For FID, lower is better. All models are benchmarked on a single Quadro RTX 6000 GPU.


Figure 3: Qualitative results generated by our method on FFHQ (left), CUB (middle), and ImageNet Church (right). The top row shows the images that are used to provide metadata for the model.

**Implementation.** There are three stages in our model, and the scale of the output images is $256 \times 256$. However, the number of stages and the size of the output image are easily adjusted to satisfy users' preference. The hyperparameters $\lambda_1$, $\lambda_2$, and $\lambda_3$ are set to 1, 5, and 50, respectively. The models are trained for roughly two days on a single GPU, using the Adam optimizer (Kingma & Ba, 2014) with learning rate $0.0002$. To have the best lightweight structure and also a good performance, for the FFHQ dataset, we suggest to add our proposed long-range module to the $32 \times 32$ feature map, and for the CUB bird and ImageNet datasets, we suggest to add it to the $64 \times 64$ feature map. To have a fair comparison on the architecture and inference time, all models are implemented in PyTorch to reduce the influence from different machine learning libraries. The reproduction of these approaches is based on the source code released by authors with a careful fine-tuning. Note that to have a fair comparison, we also restrict the training time for all three methods, which is two days on a single Quadro RTX 6000 GPU.

### 4.1 Quantitative and Qualitative Comparison

**Quantitative comparison.** To evaluate the quality and diversity of synthetic images, the Fréchet inception distance (FID) (Heusel et al., 2017) is adopted. Also, we record the inference time for

Karras et al., 2017 Zhang et al., 2019 Ours

Figure 4: Qualitative comparison of three methods on the FFHQ (top row), CUB (middle), and ImageNet Church (bottom) datasets.

generating 100 images (IT) and the number of parameters in both the generator (NoP-G) and the discriminator (NoP-D) (Million) to verify the efficiency of our method.

As shown in Table 1, our method has better FID values than SAGAN and competitive results compared with PGGAN. However, our method has a much smaller inference time (one order of magnitude speedup over PGGAN and two times faster than SAGAN) and fewer number of parameters. This indicates that (1) our method can generate high-quality results with a good diversity, and (2) our method is suitable for memory-limited devices, based on the size of model and inference time.

**Qualitative results.** As shown in Fig. 3, we presents example results generated by our lightweight long-range network at $256 \times 256$ along with images that are used to provide metadata for the model. As we can see, our model is able to produce high-quality images with a good diversity. Also, we can easily observe that the generated results are obviously different from the images providing metadata, and our model can even produce highly diverse results on the same meta-images, for example, the synthetic churches have a quite different structure, shape, texture, color, and background from those churches shown in meta-images. This indicates that our method can completely filter fine details contained in the meta-images, and effectively avoid copying and pasting from them.

Fig. 4 shows the visual comparison between PGGAN, SAGAN, and our method on the FFHQ, CUB bird, and ImageNet datasets. As we can see, SAGAN fails to produce realistic images at the scale $256 \times 256$ on three datasets, and compared with PGGAN, our method can achieve a competitive performance on the FFHQ dataset, but has better results on the CUB bird and ImageNet datasets. We think that the better performance on both datasets is mainly because (1) the training time is restricted, which may have a bigger impact on the performance of PGGAN and SAGAN, (2) the size of CUB and some specific categories in ImageNet are small, which may not be enough to fully optimize a heavy model with a large number of parameters. There are about 8k training images in CUB and on the average 2k images for specific categories in ImageNet tested in the paper. However, thanks to the long-range module and the provision of metadata, our model only has a small number of parameters to optimize and also gains some training cues from the metadata in advance, and (3) images in both datasets may involve the generation of multiple objects with complicated interactions rather than generally unified faces with similar texture patterns, and capturing the long-range dependencies is important in such complex image generation. This can be verified in Table 1: the model without the long-range module has poor FID values.

### 4.2 Ablation Studies

**Effectiveness of the metadata.** To verify the effectiveness of the metadata, we first conduct an ablation study, shown in Table 1 and Fig. 5. As we can see, with the implementation of metadata, the FID values are improved. Also, shown in Fig. 5, the FID value for the model without metadata has a significant change after epoch $= 1500$ in ImageNet Church (bottom), which indicates that the model may break due to the instability of GANs. On the contrary, the curve of our full model is

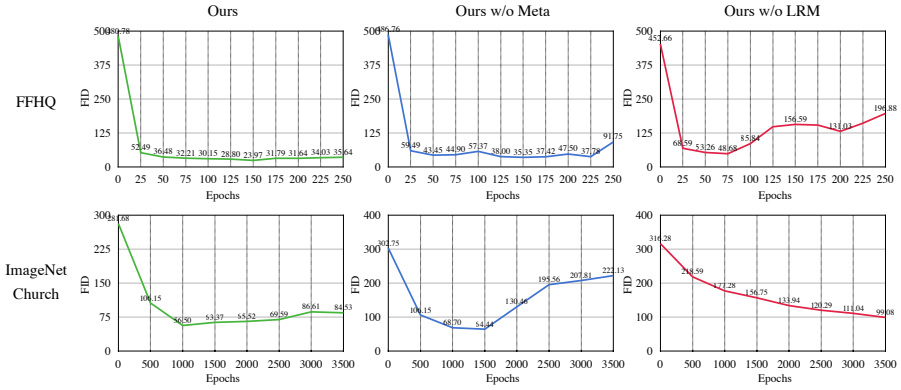



Figure 5: FID values at different epochs on the FFHQ (top) and ImageNet Church (bottom) datasets.

stable, which demonstrates that the implementation of metadata can stabilize the training process and prevent mode collapse.

Besides, as shown in Fig. 5, compared with the model without the provision of metadata, our method can achieve a similar FID value but using much fewer epochs. For example, to reach the FID $=$ $36.48$ in FFHQ (top), our full model only uses $50$ epochs, while the model without the metadata spends about $150$ epochs. This demonstrates that the adoption of metadata can speed up the training process, enabling the possibility to fast optimize the model in memory-limited devices.

**Effectiveness of the long-range module.** To verify the effectiveness of the long-range module, we first conduct an ablation study, shown in Table 1 and Fig. 5. As we can see, without the implementation of the long-range module, the model has poor FID values in all three datasets. Also, as shown in Fig. 5, the model without long-range module has significantly changeable FID values in FFHQ dataset (top red), and is hard to converge on ImageNet Church (bottom red), as the FID is still decreasing when epoch $=$ $3500$, while our full model has achieved a better FID value at epoch $=$ $1000$. This demonstrates that (1) our long-range module can effectively capture long-range dependencies to achieve a fast high-quality image generation, and (2) our proposed long-range module can work as a regularization to stabilize the training process and to prevent mode collapse.

Furthermore, to verify the performance of the long-range module, we conduct a comparison study that we use a residual block to replace our long-range module in the model (see Table 1), where the residual block has a similar number of parameters as the long-range module. We can easily observe that the model with the residual block has relatively worse FID values on three datasets. This comparison study demonstrates that the performance improvement achieved by using our long-range module is not simply because of an increase in model depth and capacity.

Besides, Table 1 presents another comparison study, where the long-range module is replaced by self-attention. As we can see, compared with our full model, the model with self-attention has worse FID values on three datasets, and is even broken on FFHQ, because the FID value is 7 times larger than the value achieved by our method. This may indicate that the negative relationships preserved by our proposed long-range module can improve the performance of the model.

## 5 Conclusion

We have proposed novel lightweight long-range generative adversarial networks, which can efficiently generate realistic results without sacrificing image quality. More specifically, our model has a much smaller number of parameters and shorter inference time, but can still produce high-quality synthetic results. To achieve this, we have proposed a novel long-range module to capture long-range dependencies, which can also work as a regularization to prevent mode collapse. Besides, we have incorporated metadata into the image generation process to provide basic information about target images, which can stabilize the model and significantly speed up the training process. Extensive experimental results demonstrate the competitive performance of our method on three benchmark datasets, in terms of both visual fidelity and efficiency.

## REFERENCES


Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C Cobo, and Karen Simonyan. High fidelity speech synthesis with adversarial networks. *arXiv preprint arXiv:1909.11646*, 2019.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1511–1520, 2017.

Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer VTeroision*, pp. 764–773, 2017.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 933–941, 2017.

Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5706–5714, 2017.

Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 2019.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

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

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

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690, 2017.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. In *Advances in Neural Information Processing Systems*, pp. 2065–2075, 2019.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7880–7889, 2020a.

Bowen Li, Xiaojuan Qi, Philip HS Torr, and Thomas Lukasiewicz. Image-to-image translation with text guidance. *arXiv preprint arXiv:2002.05235*, 2020b.

Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: manipulating images with natural language. In *Advances in Neural Information Processing Systems*, pp. 42–51, 2018.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

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

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

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

Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pp. 4790–4798, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. 2011.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807, 2018a.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018b.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1316–1324, 2018.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5907–5915, 2017.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1947–1962, 2018.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pp. 7354–7363, 2019.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, 2017.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9308–9316, 2019.

# A ARCHITECTURE

As shown in Fig. 2, the image encoder for metadata generation is a pretrained VGG-16 (Simonyan & Zisserman, 2014) network. To extract more semantic information instead of fine contextual details, the deep neural network layer relu5_3 of VGG-16 is adopted to derive the global visual representations, which contains more basic semantic information, such as the category and texture of objects, instead of fine-grained color, location, and shape information.

## A.1 RESIDUAL BLOCK

Each residual block contains two convolutional layers, two instance normalizations (INs) (Ulyanov et al., 2016), and one GLU (Dauphin et al., 2017) non-linear function. The architecture of the residual block is shown in Fig. 6.

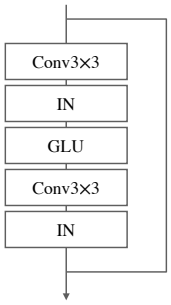



Figure 6: Architecture of the residual block.

## A.2 UPSAMPLING BLOCK

Each upsampling block contains one upsample function with nearest mode, one instance normalization (IN), one convolutional layer, and one GLU non-linear function. The architecture of the upsampling block is shown in Fig. 7.

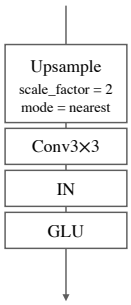



Figure 7: Architecture of the upsampling block.

# B ADDITIONAL RESULTS ON FFHQ AND CUB

Figs. 8 and 9 present additional results generated by our method on the FFHQ and CUB bird datasets, respectively.

# C ADDITIONAL IMAGENET RESULTS

Figs.10 and 11 presents additional generated examples from the ImageNet dataset. A separate network is trained for each category using identical parameters.

Figure 8: Additional results generated by our method on the FFHQ dataset at $256 \times 256$.

Figure 9: Additional results generated by our method on the CUB bird dataset at $256 \times 256$.

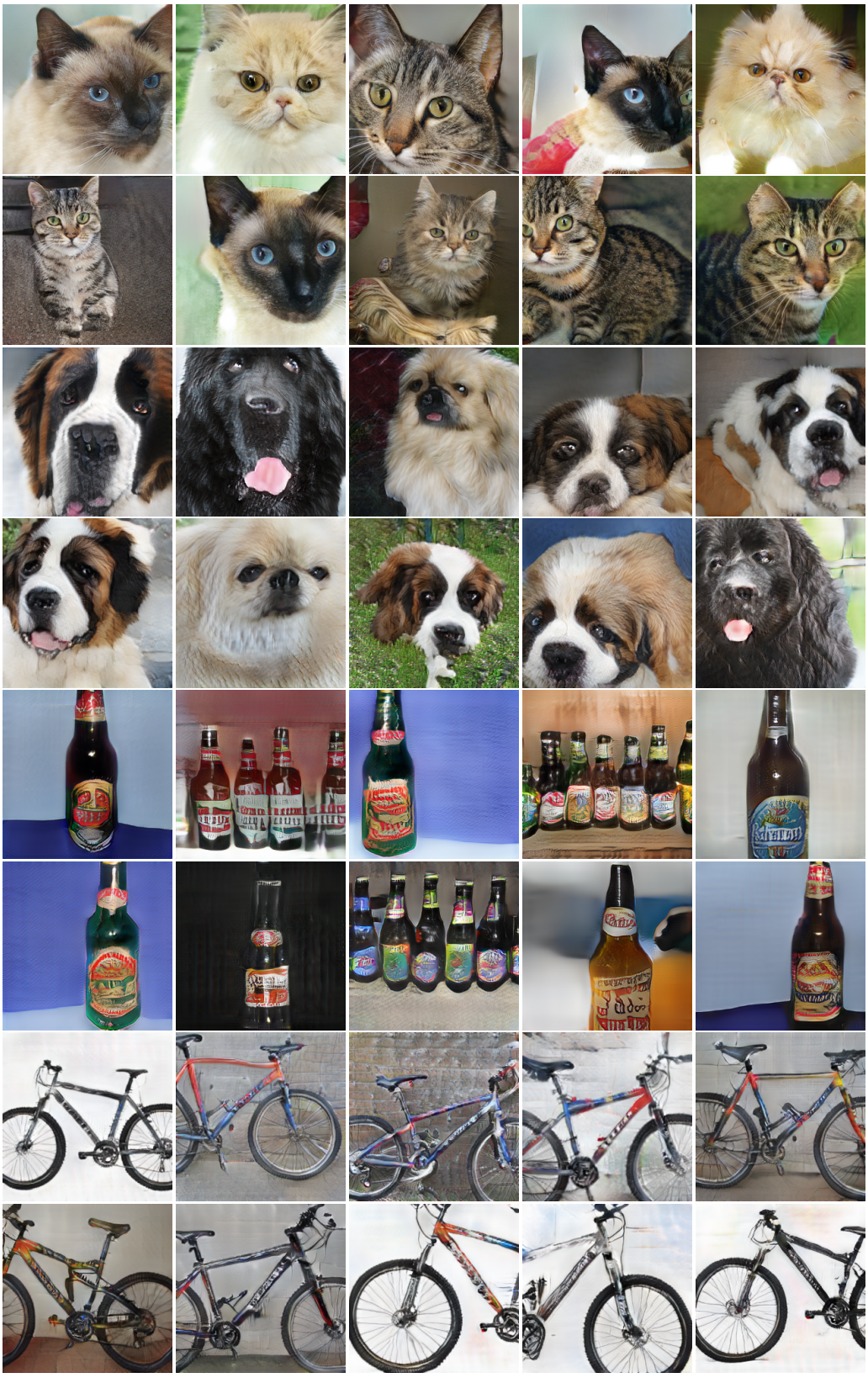


Figure 10: Additional results generated by our method on the ImageNet dataset at $256 \times 256$. The categories are cat, dog, beer bottle, and bike.

Figure 11: Additional results generated by our method on the ImageNet dataset at $256 \times 256$. The categories are baobab, valley, bus, and tiger.