Learning Inception Attention for Image Synthesis and Image Recognition

Jianghao Shen and Tianfu Wu
Department of Electrical and Computer Engineering, North Carolina State University, USA
{jshen27, tianfu.wu}@ncsu.edu

Abstract
Image synthesis and image recognition have witnessed remarkable progress, but often at the expense of computationally expensive training and inference. Learning lightweight yet expressive deep model has emerged as an important and interesting direction. Inspired by the well-known split-transform-aggregate design heuristic in the Inception building block, this paper proposes a Skip-Layer Inception Module (SLIM) that facilitates efficient learning of image synthesis models, and a same-layer variant (dubbed as SLIM too) as a stronger alternative to the well-known ResNeXts for image recognition. The proposed method is built on the idea of developing spatially-adaptive feature modulation/attention in a network. Specifically, SLIM is done from an input source feature map to a target one, either skip-layer or same-layer. In SLIM, the input feature map is first split into a number of groups (e.g., 4). Each group is then transformed to a latent style vector (via channel-wise attention) and a latent spatial mask (via spatial attention). The learned latent masks and latent style vectors are aggregated to modulate the target feature map. For generative learning, SLIM is built on a recently proposed lightweight Generative Adversarial Networks (i.e., FastGANs) which present a skip-layer excitation (SLE) module. For few-shot image synthesis tasks, the proposed SLIM achieves better performance than the SLE work and other related methods. For one-shot image synthesis tasks, it shows stronger capability of preserving image structures than prior arts such as the SinGANs. For image classification tasks, the proposed SLIM is used as a drop-in replacement for convolution layers in ResNets (resulting the ResNeXt-like models) and achieves better accuracy in the ImageNet-1000 dataset, with significantly smaller model complexity.

1. Introduction
1.1. Motivation and Objective
Image synthesis and classification are classic problems in computer vision and machine learning, and they remain challenging problems. Remarkable progress has been made since the recent resurgence of deep neural networks (DNNs). Super-human performance have been reported for image classification in ImageNet-1000 [9], and the high quality synthesized images even triggered serious concerns of the so-called deep fake [12]. However, synthesizing and classifying images typically entails computationally expensive training and inference, which may even lead to some environmental issues due to the carbon footprint [47]. Thus, learning light-weight yet highly-expressive deep models has emerged as an important and interesting research direction, especially with less data. This paper focuses on learning few-shot and one-shot image synthesis models and on designing and learning smaller yet expressive models from scratch for image recognition.

Designing and/or searching better neural architectures has been always one main fundamental problem in deep learning. Among many remarkable work, the ResNets [14] and the ResNeXts [56] are representative architectures which have been deployed in numerous applications and used as strong baselines in state-of-the-art DNN development.

Consider generative adversarial networks (GANs), state-of-the-art methods such as BigGANs [4] and StyleGANs [24, 26] utilize ResNets as their backbones. Although powerful, as the resolution of synthesized images goes higher, the width and the depth of a generator network goes wider and deeper accordingly, leading to much increased memory footprint and longer training time. Recently, by exploiting the differentiable data augmentation methods [63], Liu et al [35] present a FastGAN approach which further introduces a skip-layer channel-wise excitation (SLE) module as a drop-in component for both generator and discriminator networks, as well as a decoder enhanced design for self-supervised discriminator networks. FastGANs have shown exciting results which outperform the well-known and powerful StyleGANv2 [26] under the low-shot training settings. In this paper, we propose to extend the SLE module in FastGANs by leveraging both channel-wise and spatial attention mechanisms to improve the quality of synthesized images, especially at the resolu-
tion of 1024×1024, while retaining the efficiency and stability of training.

In the meanwhile, learning generative models with less data has been pushed to the extreme using a single image. SinGANs have shown very promising and exciting results [46]. However, state-of-the-art methods lack the capability of sufficiently preserving dominant image structures. To address the structure preservation challenge in SinGANs, in this paper, we adopt the same module that we propose to replace the SLE in FastGANs [35] to learn structure-aware SinGANs, while retaining the efficiency.

In addition to developing the skip-layer feature modulation/attention in image synthesis tasks, this paper also investigates the similar idea under the same-layer settings for DNNs in image classification tasks.

1.2. Method Overview

SLIM for Image Synthesis. Fig. 1 (Right) illustrates the proposed Skip-Layer Inception modulation (SLIM) module. Unlike the channel-wise 1-D re-calibration weights for a target feature map X in both the SE and SLE modules (i.e., C × 1 × 1), the proposed SLIM aims to learn spatial-block-wise 3-D re-calibration weights (i.e., C × H × W), which are spatial-block-wise due to the nearest upsampling operation from an h × w map to an H × W (where H > h and W > w). The learning of 3-D re-calibration weights enables richer information flow from a source feature map to a target one, which leads to better synthesizing quality.

More specifically, the proposed SLIM is motivated by the split-transform-aggregate design heuristic popularized by the well-known Google’s Inception networks [51]. In the SLIM module, an input source feature map is first split into a number of groups (e.g., 4). Each group is then transformed to a latent style vector (via channel-wise attention, the right-top of Fig. 1) and a latent spatial mask (via spatial attention, the right-bottom of Fig. 1). An optional noise injection layer is introduced to enhance the style. The learned latent masks and latent style vectors are aggregated to form the block-wise 3-D re-calibration weights modulating the target feature map. The Inception design also induces fine-grained style learning and mixing between different groups split from a source feature map.

The proposed SLIM is deployed in FastGANs [35] and SinGANs [46]. For FastGANs, the proposed SLIM module is a drop-in replacement for the original SLE module, and it obtains better performance, especially for high-resolution image synthesis at the resolution of 1024×1024, while retaining comparable efficiency. For SinGANs, it shows a stronger capability of preserving image structures than the Concurrent SinGANs (ConSinGANs) [16].

SLIM for Image Recognition. Fig. 2 illustrates the Same-Layer Inception Module (dubbed as SLIM too). Similarly, we first split the input feature map into a number of groups (e.g., 4). For input group, the transformation consists of a standard convolution (with a 3 × 3 kernel), the channel attention (based on SE), and the spatial attention (which can be treated as a spatially-adaptive variant of SE). The spatial-channel attention is the product between the channel attention and the spatial attention, resulting in a 3D attention matrix. Softmax is applied along the channel dimension of the 3D attention matrix. It is then used to re-calibrate the output of the convolution branch in an element-wise / spatially-adaptive way. It is used to replace all the 3x3 convolution in a feature backbone (e.g., ResNet-50). It achieves better accuracy with significantly smaller model complexity in ImageNet-1000 [9].
Figure 2. Illustration of the Same-Layer Inception Model (dubbed as SLIM too) for image recognition. It can be treated as an strong alternative for the widely used SE module and as a new realization of the ResNeXts [56]. See text for detail.

2. Related Work and Our Contributions

Unconditional image synthesis and GANs. Unconditional image synthesis is a generative learning task which seeks probabilistic models that are capable of explaining away data under the maximum likelihood estimation (MLE) framework. To this end, introducing multiple layers of hidden variables is often entailed, which in turn popularizes DNNs. There are two types of realizations in the literature: top-down generator models [11, 13, 28, 44, 57, 61, 56, 57] that are non-linear generalization of factor analysis [45], and bottom-up energy-based models [33, 40, 10, 27, 62, 58, 41, 29, 32] that are in the form of a Gibbs distribution or Boltzmann distribution. It is well-known that training a DNN-based generator model individually is an extremely difficult task, mainly due to the difficulty of sampling the posterior for real images. Adversarial training is a popular workaround and GANs [11] are widely used in practice, which are formulated under a two-player minmax game setting. The proposed SLIM module can be used as a drop-in component in the generator network of GANs.

Light-weight GANs with Few-Shot Learning. Compared to the extensive research on light-weight neural architectures in discriminative learning for mobile platforms, much less work has been done in generative learning [30]. The residual network [14] is the most popular choice, on top of which powerful generative models such as BigGANs [4] and StyleGANs [25, 26] have been built with remarkable progress achieved. For high-resolution image synthesis, these models will be very computational expensive in training and inference. In the meanwhile, training these models typically require a big dataset, which further increase the training time. Few-shot learning is appealing, but very challenging in training GANs, since data augmentation methods that are developed for discriminative learning tasks are not directly applicable. To address this challenge, differentiable data augmentation methods and variants [63, 23, 52, 64] have been proposed in training GANs with very exciting results obtained. Very recently, equipped with the differentiable data augmentation method [63], a FastGAN approach [35] is proposed to realize light-weight yet sufficiently powerful GANs with several novel designs including the SLE module. The proposed SLIM is built on the SLE in FastGANs by exploiting the well-known Google’s Inception [50, 51] building block design and the Atrous convolution [7].

Learning Unconditional GANs from a Single Image. There are several work on learning GANs from a single texture image [2, 22, 34]. Recently, a SinGAN approach [46] has shown surprisingly good results on learning unconditional GANs from a single non-texture image. It is further improved in ConSinGANs [16] which jointly train several stages in progressively growing the generator network. However, it remains a challenging problem of preserving image structure in synthesis. The proposed SLIM is applied to the vanilla SinGANs [46], leading to a simplified workflow that can be trained in a stage-wise manner and thus more efficient than ConSinGANs, and facilitating a stronger capability of preserving image structures.

Attention Mechanism in Deep Networks. Attention re-allocates the available computational resources to the most relevant components of a signal to the task [42, 20, 19, 31, 38, 53]. Attention mechanisms have been widely used in computer vision tasks [21, 5, 60, 6]. The SE module [17] applies a lightweight self gating module to facilitate channel-wise feature re-calibration. Our proposed SLIM module incorporates spatially-adaptive feature modulation, while maintaining the light weight design, improving the representation power for efficient discriminative learning in addition to generative learning.

Our Contributions. In summary, this paper makes the follow four main contributions in unconditional image synthesis using GANs and image analysis: (i) It proposes a Skip-Layer Inception Modulation (SLIM) module to facilitate better learning of generative models from few-shot / one-shot images, and to provide visually-interpretable visualization of the skip-layer information flow. (ii) It shows significantly better performance for high-resolution image synthesis at the resolution of 1024 x 1024 when deployed in the FastGANs [35], while retaining the efficiency. (iii) It enables a simplified workflow for SinGANs [46], and shows a stronger capability of preserving image structures than prior arts. (iv) It further presents a Same-Layer Inception Module for image classification, which serves as a stronger new realization of the ResNeXt-like models when deployed in the ResNets.
3. Approach

In this section, we present details of the proposed SLIM modules for image synthesis and image recognition.

3.1. The SLIM for Image Synthesis

Unconditional Image Synthesis. The goal is to learn a generator network which maps a latent code to an image,

\[ x = G(z; \Theta_G), \]

where \( z \) represents a 1-D latent code (e.g., in FastGANs [35]) or a 2-D latent code (e.g., in SinGANs [46]), which is typically drawn from standard Normal distribution (i.e., white noise). \( \Theta_G \) collects all the parameters of the generator network \( G \). Given a latent code, it is straightforward to generate an image.

FastGANs. As shown in Fig. 3, the generator network used in FastGANs [35] adopts a minimally-simple yet elegantly chosen design methodology. Given an input latent code, the initial generator applies the transpose convolution to map the latent code to a \( 4 \times 4 \) feature map. Then, a composite block (UpCompBlock) and a plain block (UpBlock) are interleaved to map the \( 4 \times 4 \) feature map to the one at a given target resolution (e.g., \( 1024 \times 1024 \)). Batch normalization [18] and gated linear unit (GLU) [8] are used in the building blocks. Spectral normalization [36] also is used for all the convolution and linear layers. In a composite upsample block, noise injection is used right after the convolution operation. As aforementioned, the plain feed-forward architecture will suffer from possible gradient vanishing and style effect vanishing issues. The SLIM module (the middle of Fig. 1) is proposed to mitigate these issues. In training, an auxiliary image is also synthesized at the resolution of \( 128 \times 128 \), which further helps alleviate potential gradient vanishing issues and collects multi-scale feedback from the discriminator. The discriminator in FastGANs also includes some novel designs with the SLE module applied. Since our focus is to re-design the SLE module in the generator and we keep the discriminator unchanged in the experiments, we omit the discussion of the discriminator due to space limit. Please refer the original paper [35] for details.

SinGANs. Fig. 4 (top) shows a stage in the generator of SinGANs [46]. A SinGAN is progressively trained from a chosen low resolution (based on the predefined image pyramid settings) to the resolution of the single input image. Following the notation usage in SinGANs [46], the start resolution is indexed by \( N \) and the end resolution by \( 0 \). At the very beginning, a 2-D latent code, \( z_N \) is sampled and the initial generator \( \tilde{x}_N = G_N(z_N; \theta_N) \) is trained under the vanilla GAN settings. Then, at the stage \( n \) \((N > n \geq 0)\), the generator \( G_n \) has been progressively grown from \( G_N \), and we have \( \tilde{x}_n = G_n(z_n, \tilde{x}_{n+1} \uparrow; \theta_n) \), where \( \tilde{x}_{n+1} \uparrow \) represents the up-sampled synthesized image from the previous stage \( n + 1 \) with respect to the predefined ratio used in the construction of the image pyramid. More specifically, \( \tilde{x}_n = (\tilde{x}_{n+1} \uparrow) + \psi_n(z_n + (\tilde{x}_{n+1} \uparrow); \theta_n) \). Batch normalization [18] and leaky ReLU [59] are used in a convolution block, while spectral normalization is not used. With the proposed SLIM module, we substantially change the workflow of the generator as shown in the bottom of Fig. 4. Each stage of the vanilla SinGAN is to learn the residual image on top of the output from the previous stage. As we shall elaborate, the proposed SLIM module is spatially-adaptive in modulating a target feature using a source feature map, so we remove the residual learning setting. Similarly, we keep the discriminator of SinGANs unchanged in our experiments. Due to space limit, we omit the discussion of it and more details are referred to the original paper [46].

The SLIM Module. Fig. 4 (Right) illustrates the proposed SLIM module. We first compare the formulation between the SE module [17], the SLE module [35] and the proposed SLIM module. Focusing on how an input target feature map \( X \) is transformed to the output feature map \( \tilde{X} \), denote by \( X = (x_1, \cdots, x_C) \) where \( x_c \) represents a single
channel slice of the tensor $X$ for $1 \leq c \leq C$, we have,
\[
\text{SE: } \tilde{X} = (\alpha_1 \cdot x_1, \cdots, \alpha_C \cdot x_C),
\]
where the channel importance coefficient $\alpha_c = F_{\text{SE}}(X)$ in the SE module, and $\beta_c = F_{\text{SLE}}(Y)$ in the SLE module. So, the SE module realizes self-attention between channels (a.k.a. “neurons”), while the SLE module realizes cross-attention. And, the former is a special case of the latter when $Y = X$. Both $\alpha_c$ and $\beta_c$ are scalar and shared by all spatial locations in the same channel slice. For discriminative learning tasks such as image classification, this channel-wise feature re-calibration works very well since spatial locations will be discarded by the classification head sub-network (typically via a global average pooling followed by a fully-connected layer). For image synthesis tasks whose outputs are location-sensitive, it may not be sufficient to deliver the entailed modulation effects.

The proposed SLIM module aims to facilitate spatially-aware re-calibration by extending the SLE module. It learns a 3-D weight matrix $W_{C \times H \times W}$ from the source feature map in modulating the target feature map $X_{C \times H \times W}$ (that is to “pay full attention”), and we have,
\[
\text{SLIM: } \tilde{X} = W \odot X,
\]
where $\odot$ represent the Hadamard product.

**Learning the spatially-adaptive weight matrix $W$ from $Y$.** We want to distill two types of information: One represents 1-D latent style codes that are informed by the source feature map $Y$, and then used to induce the modulated source feature map $\tilde{X}$ to focus on; The other reflects 2-D latent spatial masks that are used to distribute the latent style codes. Decoupling these two is beneficial to enable them learning faster and more accurate. In the vanilla SLE module (the middle of Fig. 1), a coarse $4 \times 4$ spatial attention map also is computed, followed by a channel-wise operation. This sequential transformation may entangle them in a less effective way, as well as the $4 \times 4$ spatial pooling may discard some useful spatial information.

**Decoupling the Style and Layout.** We decouple the channels (“neurons”) in an input source feature map by splitting them into a number of groups (e.g., 4), that is to exploit mixture modeling or clustering of the “neurons” in a building block, as suggested by the Inception module \[51\] which in turn is inspired by the theoretical study of how to construct an optimal neural architecture in a layer-wise manner with a set of constraints satisfied \[1\]. For each group, we apply the decoupled channel-wise and spatial transformation for learning the latent style codes and the latent spatial masks concurrently.

**In learning the latent style codes**, we use the transformation (the right-top of Fig. 1) similar in spirit to the vanilla SE except for adding BatchNorm \[18\] after the first fully-connected layer and removing the last sigmoid operation. We remove the sigmoid operation since the outputs will be interpreted as latent style codes, rather than the channel-wise importance weights.

**In learning the latent spatial layout/masks**, we exploit the Atrous convolution \[17\] to capture long-range contextual information. The dilation rate is chosen based on the size of an input source feature map such that the receptive field of the Atrous convolution is sufficiently large. To overcome the potential collapse issue of the learned latent spatial masks, especially in synthesizing high-resolution images, we apply a noise injection operation before the sigmoid transformation (the bottom-right of Fig. 1).

To sum up, from the channel-wise attention branches, we compute a group of $g$ latent style codes and concatenate them to form the latent style code tensor, $C_{g \times C \times 1 \times 1}$. From the spatial attention branches, we compute a group of $g$ latent spatial masks and concatenate them to form the latent spatial mask tensor, $M_{g \times 1 \times H \times W}$. Then, the 3-D weight matrix $W$ in Eqn. \[4\] is computed by,
\[
W = \frac{\text{sum}(C' \odot M', \text{dim} = 0)}{\text{sum}(M', \text{dim} = 0)},
\]
where $C'$ is a tensor of the size $g \times C \times H \times W$ by repeating $C_{g \times C \times 1 \times 1}$ along the last two dimensions, and $M'$ is a tensor of the size $g \times C \times H \times W$ by repeating $M_{g \times 1 \times H \times W}$ along the second dimension. \text{sum}(M', \text{dim} = 0) is used to normalize the spatial mask between the $g$ latent objects.

Note that the SE module, the SLE module and the proposed SLIM module all are instance specific for different inputs in a batch.

### 3.2. Comparing the SLIM with Alternative Designs in Image Synthesis

**Comparing with the weight modulation in StyleGANv2 \[26\].** The weight modulation method in StyleGANv2 is an elegantly designed operation to achieve detailed style tuning effects. The style code is used to directly modulate the filter kernels (as model parameters) in an instance specific, and then modulated filter kernels are used in computing the convolution. Although being highly expressive, this weight modulate is not spatially-adaptive. And, it increases the computational burden and the memory footprint in execution. The proposed SLIM directly modulates the feature map in a light-weight manner.

**Comparing with the SPADE in GauGANs \[43\] and the ISLA-Norm in LostGANs \[48\].** Both the SPADE and the ISLA-Norm exploit spatially-adaptive modulation, but apply it inside the BatchNorm. They replace the vanilla channel-wise affine transformation in the BatchNorm with spatially-adaptive affine transformation. The spatially-adaptive affine transformation coefficients are learned either
Figure 5. Illustration of deploying the SLIM in ResNets (right), compared with the SE (left) and the vanilla ResNeXt (middle).

from the input semantic masks in GauGANs or the generated latent masks from the input layouts in LostGANs. The proposed SLIM is similar in spirit to the ISLA-Norm, but is formulated under the Inception architecture together with the skip-layer idea proposed in FastGANs [35].

3.3. The SLIM for Image Recognition

Fig. 2 illustrates the SLIM for substituting a vanilla 3x3 Convolution in a network such as the ResNet [14]. Let \(X_{C \times H \times W}\) be an input feature map. Fig. 5 shows the deployment of the SLIM in the residual building block. For the SLIM, at a very high-level, the outputs of the channel-attention and the spatial-attention, and the convolution in a group play the roles of query, key and value in the Transformer model [53]. The multi-group represents the multi-head design.

4. Experiments

In this section, we test the proposed SLIM in few-shot image synthesis using FastGANs [35], one-shot image synthesis using SinGANs [46], and in ImageNet-1000 classification using ResNets [14]. Our PyTorch source code will be released soon, which is built on officially released PyTorch codes of FastGANs [1], SinGANs [2], and the MMClassification Github Repo [3], respectively.

4.1. Results of SLIM-FastGANs

Data. We adopt the datasets used in the vanilla FastGANs [35]. There are 5 categories tested at the resolution of \(256 \times 256\) each of which uses around 100 training images (see part one of Table 4). There are 7 categories tested at the resolution of \(1024 \times 1024\) each of which uses around 1000 training images (see part two of Table 4).

Settings: We follow settings in the official code of FastGANs. We keep the first SLE module from the initial block considering the small spatial size that may not be helpful for learning spatial masks and replace the remaining SLE by the proposed SLIM. And, the dimension of the input latent code is set to 256 (Fig. 3). One thing to highlight is the output size of the discriminator. There are two different settings used in report: \(1 \times 1\) or \(5 \times 5\), which show different performance on different categories. We use \(5 \times 5\) as the output size (the default in the official code), so some of the results of the proposed SLIM module could be further improved. At the resolution of \(1024 \times 1024\), we apply

![Synthesized faces at 1024x1024 in FFHQ](https://github.com/tamarott/SinGAN)

Learned latent spatial masks in the proposed SLIM

| Model / Training Time (hours) | Art | FFHQ | Flower | Pokemon | AnimeFace | Skull | Shell |
|------------------------------|-----|------|--------|----------|-----------|-------|-------|
| StyleGAN2 / 24               | 74.56 | -    | -      | 190.23   | 152.73    | 127.98 | 241.97 |
| StyleGAN2 finetune / 24      | -    | -    | -      | 60.12    | 61.23    | 107.68 | 228.45 |
| SLE-FastGAN / 8              | 45.08 | 44.30 | 31.71  | 59.19    | 59.56    | 130.05 | 155.92 |
| SLIM-FastGAN (ours) / 9      | 43.46 | 39.59 | 29.90  | 51.2     | 54.22    | 101.16 | 140.45 |

Table 1. FID comparisons (smaller is better) on the 7 categories with models trained with 1000 images and tested at the resolution of \(1024 \times 1024\). \(†\) means that results are obtained by retraining the vanilla FastGANs. The 1000 images are randomly sampled from the original much larger datasets: FFHQ and Oxford-Flower.

| Resolution         | SLIM-FastGAN | SLE-FastGAN |
|--------------------|--------------|-------------|
| \(256 \times 256\) | **28.5M**    | 29.1M       |
| \(1024 \times 1024\)| **28.5M**    | 29.2M       |

Table 2. #Parameters of generators of SLIM-FastGAN and SLE-FastGAN, in the models for \(256 \times 256\) and \(1024 \times 1024\) resolutions, respectively.
the noise injection layer in the SLIM (the right-bottom of Fig.1). At the resolution of $256 \times 256$, no noise injection is applied in the SLIM. The dilation rates for the Axrous convolution in the spatial branches are set to 2, 2 and 4 for the SLIM modules from the stage $8^2$, $16^2$ and $32^2$ respectively. The learning rates follow the default value (0.0002) except for the FFHQ which uses 0.0001. The Adam optimizer is used with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. A single GPU is used in training.

We compare the proposed SLIM with several baselines: 1)FastGAN [35]. 2)The adaptive data augmentation method [23], in which we follow the original paper to set the augmentation probability such that it can increase from 0 to 1 within 10k iterations (1/5 of the total training time). 3)The SPAP module [49] which leverages multi-scale spatial attention for GAN learning [39]. 4)SPAP module [49] which leverages multi-scale spatial attention for GAN learning [39]. For 3) and 4), we replace the original SLE, CBAM and SPAP modules with models trained with more data (2k, 5k and 10k) at the resolutions of $1024 \times 1024$. A single GPU is used.

### Metrics

We evaluate our methods on FID [15], KID [3] for image quality. To assess the overfitting of the methods, we further evaluate them on latent recovery [54], density and coverage [59] metrics.

### Efficiency

The proposed SLIM module increases the training time in training models for high-resolution image synthesis (e.g., 1/8 relative increase in time in Table 1). The training time increase is negligible for training image synthesis models at lower resolutions such as $256 \times 256$. We further compare the number of parameters of our proposed SLIM method with original FastGAN at Table 7 we can see that the proposed SLIM has less parameter than the SLE-FastGAN. This is because in SLIM, we down scale the channel number by a factor of 16 through a linear layer before extracting the style and spatial information from the input, resulting in less parameters than SLE; This shows the effectiveness of our design.

### Results

Table 4 shows the comparison of our proposed SLIM module with related methods. The method with '-'
our SLIM is complementary to ADA in improving performance in few-show image synthesis. For overfitting evaluation, in terms of MRE/p-value, our SLIM is worse than SLE in low-res image dataset. Note that a p-value greater than 0.01 means no major concerns of memorization. Both SLE and SLIM seem work reasonably well. It shows our SLIM is more conservative towards the training distribution when less data (e.g., 100) and less information (low-resolution) are used in training and higher fidelity of generated images are expected. This conservative synthesis may be potentially less biased in generative learning. Table 3 shows the comparison results.

4.2. Results of SLIM-SinGANs

Data. We tested the SLIM-SinGAN on 14 images used by the vanilla SinGANs: Brandenberg, bridge, Golden gate, tower, angkorwat, balloons, birds, colusseum, mountains, starry night, tree, cows, volcano, zebra. The images are selected to cover different image structures which are difficult for one-shot GANs.

Settings. We follow the settings provided by the vanilla SinGANs [46]. Two baselines are used: (i) ConSinGANs for which we follow the suggestions in the paper to try different combinations between the learning rate and the number of stages jointly trained and select the best results. (ii) SLE-SinGANs in which the SLE module is used, instead of SLIM, in the bottom of Fig. 4.

Metrics. We evaluate our methods with single image FID (SIFID) and Diversity Score proposed in the vanilla SinGANs [46].

Efficiency. The proposed SLIM-SinGANs are as efficient as the vanilla SinGANs in terms of training speed. Results. Table 5 shows the comparison results. Fig. 7 shows synthesized images for the Egyptian pyramid image. In terms of diversity score, Our SLIM obtains lower diversity in the trend similar to ConSinGAN. The testing images are structure-rich images for which our goal is to study how to preserve the structure. The diversity score should be interpreted jointly with the SIFID. We also observe that SLIM-SinGANs may suffer from the training image memorization problem while trying to preserve the structures in images, especially for the images with dominant foreground structural objects, which we leave to investigate in future work. More qualitative results will be provided in the supplementary material.

4.3. Results of the SLIM in ImageNet-1000

Table 6 (top) shows the results of applying the SLIM in ResNet-50. Compared with the SE, our SLIM obtains 0.32% top-1 accuracy increase, while significantly reducing the model parameters (by 9M) and FLOPs. Compared with ResNeXt-32x4d-50, our SLIM obtains 0.16% top-1 accuracy increase with much less parameters too. These results clearly show the effectiveness of the proposed SLIM.

Ablation study. To test if the skip-layer connections help classification tasks too and if the SLIM module helps more than the SLE module, we apply the vanilla SLE and our SLIM in the VGG-11+BatchNorm network which does not have skip connections as in ResNets. The skip-layer connections are introduced between the 5 stages in VGG-11 with stage 1/2/3 connected to stage 3/4/5 respectively. Our objective is. Table 6 (bottom) shows the results, which shows that skip-layer connections can also help in classification, and our SLIM works better than the vanilla SLE.

5. Conclusion

This paper proposes a method of learning Inception Attention in both few-shot/one-shot image synthesis and large-scale image recognition. A Skip-Layer Inception Module (SLIM) is presented for the former, and a Same-Layer Inception Module (dubbed as SLIM too) for the latter. Both share the same idea, that is to explore and exploit spatially-adaptive feature modulation/recalibration by jointly learning both channel-wise attention (based on the popular Squeeze-Excitation module) as latent style representation and spatial attention as latent layout representation. The resulting Inception Attention is a 3D attention matrix in modulating/recalibrating the input feature map. In experiments, the proposed SLIM module is tested in few-shot image synthesis using FastGANs [35], one-shot image synthesis using SinGANs [46], and ImageNet-1000 classification using ResNets. The SLIM-FastGANs are consistently better than the vanilla FastGANs, SPAP, CBAM, and obtain significantly better performance at high-resolution image synthesis. The SLIM-SinGANs show stronger ca
Figure 7. Synthesized images by the vanilla SinGAN, ConSinGAN, SLE-SinGAN and our SLIM-SinGAN. SLIM-SinGAN is better in terms of preserving structure, while producing meaningful semantic variations (note the change of number of Sphinx statues of SLIM-SinGAN results).

pabilities in preserving image structures than prior arts such as the ConSinGANs [16]. The SLIM-ResNets show better performance than the SE variant and the vanilla ResNeXt with model complexity significantly reduced.

Acknowledgements

This work were supported in part by NSF IIS-1909644, ARO Grant W911NF1810295, NSF IIS-1822477, NSF CMMI-2024688, NSF IUSE-2013451 and DHHS-ACL Grant 90IFDV0017-01-00. The views presented in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[1] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In International conference on machine learning, pages 584–592. PMLR, 2014.

[2] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. Learning texture manifolds with the periodic spatial gan. arXiv preprint arXiv:1705.06566, 2017.

[3] Mikolaj Binkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. arXiv preprint arXiv:1801.01401, 2018.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In International Conference on Learning Representations, 2019.

[5] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In Proceedings of the IEEE international conference on computer vision, pages 2956–2964, 2015.

[6] Long Chen, Hanwang Zhang, Jun Xiao, Lijiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5659–5667, 2017.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017.

[8] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In International conference on machine learning, pages 933–941. PMLR, 2017.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[10] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial net-
works, inverse reinforcement learning, and energy-based models. arXiv preprint arXiv:1611.03852, 2016.

[11] 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, pages 2672–2680, 2014.

[12] David Guerra and Edward J Delp. Deepfake video detection using recurrent neural networks. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–6. IEEE, 2018.

[13] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Alternating back-propagation for generator network. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pages 1976–1984, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[15] 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. Advances in neural information processing systems, 30, 2017.

[16] Tobias Hinz, Matthew Fisher, Oliver Wang, and Stefan Wermter. Improved techniques for training single-image gans. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1300–1309, 2021.

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, pages 448–456, 2015.

[19] Laurent Itti and Christof Koch. Computational modelling of visual attention. Nature reviews neuroscience, 2(3):194–203, 2001.

[20] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on pattern analysis and machine intelligence, 20(11):1254–1259, 1998.

[21] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. Advances in neural information processing systems, 28:2017–2025, 2015.

[22] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. Texture synthesis with spatial generative adversarial networks. arXiv preprint arXiv:1611.08207, 2016.

[23] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. arXiv preprint arXiv:2006.06676, 2020.

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

[25] 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, pages 4401–4410, 2019.

[26] 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/CVF Conference on Computer Vision and Pattern Recognition, pages 8110–8119, 2020.

[27] Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. arXiv preprint arXiv:1606.03439, 2016.

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In Proceedings of the 2nd International Conference on Learning Representations (ICLR), 2014.

[29] Ritesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. arXiv preprint arXiv:1901.08508, 2019.

[30] Karol Kurach, Mario Lucic, and Xiaohua Zhai Marcin Michalski Sylvain Gelly. The gan landscape: Losses, architectures, regularization, and normalization. arXiv preprint arXiv:1807.04720, 2018.

[31] Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. Advances in neural information processing systems, 23:1243–1251, 2010.

[32] Justin Lazarow, Long Jin, and Zhuowen Tu. Introspective neural networks for generative modeling. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 2793–2802. IEEE Computer Society, 2017.

[33] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. Predicting structured data, 1(0), 2006.

[34] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In European conference on computer vision, pages 702–716. Springer, 2016.

[35] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. arXiv e-prints, pages arXiv–2101, 2021.

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

[37] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In International Conference on Machine Learning, pages 1791–1799, 2014.

[38] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In Advances in neural information processing systems, pages 2204–2212, 2014.

[39] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In International Conference on Machine Learning, pages 7176–7185. PMLR, 2020.
6. Supplementary Materials

6.1. Training details of SLIM-FastGANs

We use the Adam optimizer for training, with $\beta_1=0.5$, $\beta_2=0.999$. We use learning rate of 0.0002 for all datasets except for FFHQ and the panda datasets, in which we use 0.0001. For the architecture of Discriminator, we adopt the output size of $5 \times 5$; and for SLIM-FastGAN model on the $1024 \times 1024$ datasets, we apply Gaussian noise injection to the spatial masks of SLIM, with 0 mean and unit variance; For the convolution of spatial branch of SLIM, we set the dilation rates as 2, 2, 4 at stage $8 \times 8$, $16 \times 16$, $32 \times 32$, respectively.

6.2. More results of SLIM-FastGANs

6.2.1 Clarification on results on FFHQ-1k

We notice there is a gap of the performance on FFHQ-1K between our retrained version based on the official FastGAN code and the reported one in the paper. Thanks to the author’s feedback via emails, the best configuration for reproducing the FFHQ-1k result of FID=24.45 will NOT be released since it is deployed on a commercial platform. So the FID of the SLE-FastGAN we trained based on the FastGAN code is worse than the one reported in the paper.

|                          | FID     |
|--------------------------|---------|
| SLE-FastGAN [35] (reported in the paper) | 24.45   |
| Retrained from the official FastGAN code | 44.31   |
| Retrained from the official FastGAN code (dataset-specific version) | 42.82   |
| Our SLIM-FastGAN built on the official FastGAN code | 39.59   |

Table 7. FFHQ-1k performance comparison between the reported result in FastGAN paper, our retrained version based on their code, and the proposed SLIM-FastGAN

6.2.2 Synthesis results of SLIM-FastGAN on $1024 \times 1024$ datasets

Fig. 16 shows the example synthesized $1024 \times 1024$ images of our proposed SLIM-FastGAN.

6.2.3 Backtracking results

Settings (Thanks to the clarification by the authors via emails) 1) We first split the dataset into train/test ratio of 9:1. 2) Train the model on the splitted training set. 3) Pick the trained generator checkpoint at iteration (20k, 40k, 80k) respectively, and do latent backtracking for 1k iterations on test set. 4) Compute the mean LPIPS between the test images and the reconstructed images from backtracking of the corresponding checkpoints. Where LPIPS is the average perceptual distance between two set of images; in this test, a lower LPIPS is better because it means that the model that is trained on the training set can backtrack an unseen test set with small reconstruction error, indicating less overfitting.

Results Fig. 17 shows the example backtracking results on several of the few-shot datasets. The smooth transition of the interpolated images between the backtracked test images show that our model hasn’t overfit to the training set.

6.2.4 Style mixing results

Settings To demonstrate that the proposed SLIM is able to disentangle the high level semantic attributes of features at different scales, we conduct the style mixing experiment as done in the FastGAN paper [35], in which for a pair of style and content images, we extract channel weights from style images, and use them to modulate the features of content images, while retaining the spatial masks of the content images. The resulting effects as shown in Fig. 18 is that the appearance and color scheme of the style image is propagated to the content image, and the spatial structure of the content image is unchanged.

6.3. More results of SLIM-SinGANs

In order to demonstrate the strength of the proposed SLIM module in one-shot generative learning. We present the comparison of synthesis results of SLIM-SinGAN with other related methods on 9 images. We also show the results of image harmonization and editing under one-shot setting.
6.3.1 Synthesis with self-collected images

Fig. 8 to Fig. 15 are the synthesis results comparison on 9 images. We can see that compared to ConSinGAN, SinGAN and SLE-SinGAN, SLIM-SinGAN captures the global layout of the image better, while producing meaningful local semantic variations.

6.3.2 Harmonization

Fig. 19 shows the comparison on one-shot image harmonization task as done in [46]. It shows that our proposed SLIM-SinGAN can realistically blend an object into the background image.

6.3.3 Editing

Fig. 20 shows the comparison on one-shot image editing task as done in [46]. It shows that our proposed SLIM-SinGAN is able to produce a seamless composite in which image regions have been copied and pasted in other locations. Note that SLIM-SinGAN shows more realistic composite within the edited regions.

Figure 8. One-Shot synthesis comparison on the bridge image. Note how the synthesized images of SLIM-SinGAN capture the global layout of the real image, and at the same time produces semantically meaningful variations (size, number of towers at top.)
Figure 9. One-Shot synthesis comparison on the Great Wall image.

Figure 10. One-Shot synthesis comparison on the capitol hill image.
Figure 11. One-Shot synthesis comparison on the ancient Chinese tower image.

Figure 12. One-Shot synthesis comparison on the Lincoln memorial image.
Figure 13. One-Shot synthesis comparison on the Temple of Heaven image.

Figure 14. One-Shot synthesis comparison on the Temple of ancient Chinese tower image.
Figure 15. One-Shot synthesis comparison on the Temple of Heaven image.
Figure 16. Examples of synthesized images at the resolution of $1024 \times 1024$. Best viewed in magnification.
Figure 17. Example backtracking results,
Figure 18. Examples style mixing results. Best viewed in magnification.
Figure 19. One-Shot harmonization comparison on example images.
Figure 20. One-Shot Editing comparison on example images.