Styleformer: Transformer based Generative Adversarial Networks with Style Vector

2021.10.06

백서인

Abstract

- Propose Styleformer, which is a style-based generator for GAN architecture, but a convolutional-free transformer-based generator

- Explain how a transformer can generate high-quality images, overcoming the disadvantage that convolution operations are difficult to capture global features in an image

- Change the demodulation of StyleGAN2 and modify the exisiting transformer structure to create a strong style-based generator with a convolution-free structure

Abstract

- Make Styleformer lighter by applying Linformer, enabling Styleformer to generate higher resolution images and result in improvements in terms of speed and memory

- Experiment with the low-resolution image dataset such as CIFAR-10, as well as the high-resolution image dataset like LSUN-church

Introduction

- GANsformer combines convolution and transformer to improve the performance of generation
- TransGAN presents a GAN structure that allows image generation without a convolution network, using only a transformer
- However, despite these efforts, convolution-based models still have the upper hand, rather than models using a transformer in GANs

Introduction

- State-of-the-art GANs like StyleGAN2 and BigGAN can generate realistic, high-fidelity images

- However, since all of these models are based on convolution backbones, they have a locality problem, which makes it difficult to capture global features

Introduction

- We propose Styleformer, a style-based generator only consisting of transformer modules that shows superior performance over other generative models

- 1) MLP-Mixer have similar effect with convolution

- 2) Change the demodulation operation, which is treated as a core in StyleGAN2, to fit the transformer

- 3) Modify the style injection method in StyleGAN2 and the structure of the transformer such as residual connection and layer normalization, to make the efficient transformer-based generator

- 4) Apply Linformer to Styleformer to generate high-resolution images

MLP-Mixer

- Although not a convolution network structure, architecture with pixel-to-pixel and channel-to-channel operations can be efficient in learning information about images

- We divide the self-attention mechanism in the original transformer into prepare module and main module

  - Prepare module is a module that creates query, key, and value to conduct attention

  - Main module can be divided again into a pixel section and a channel section

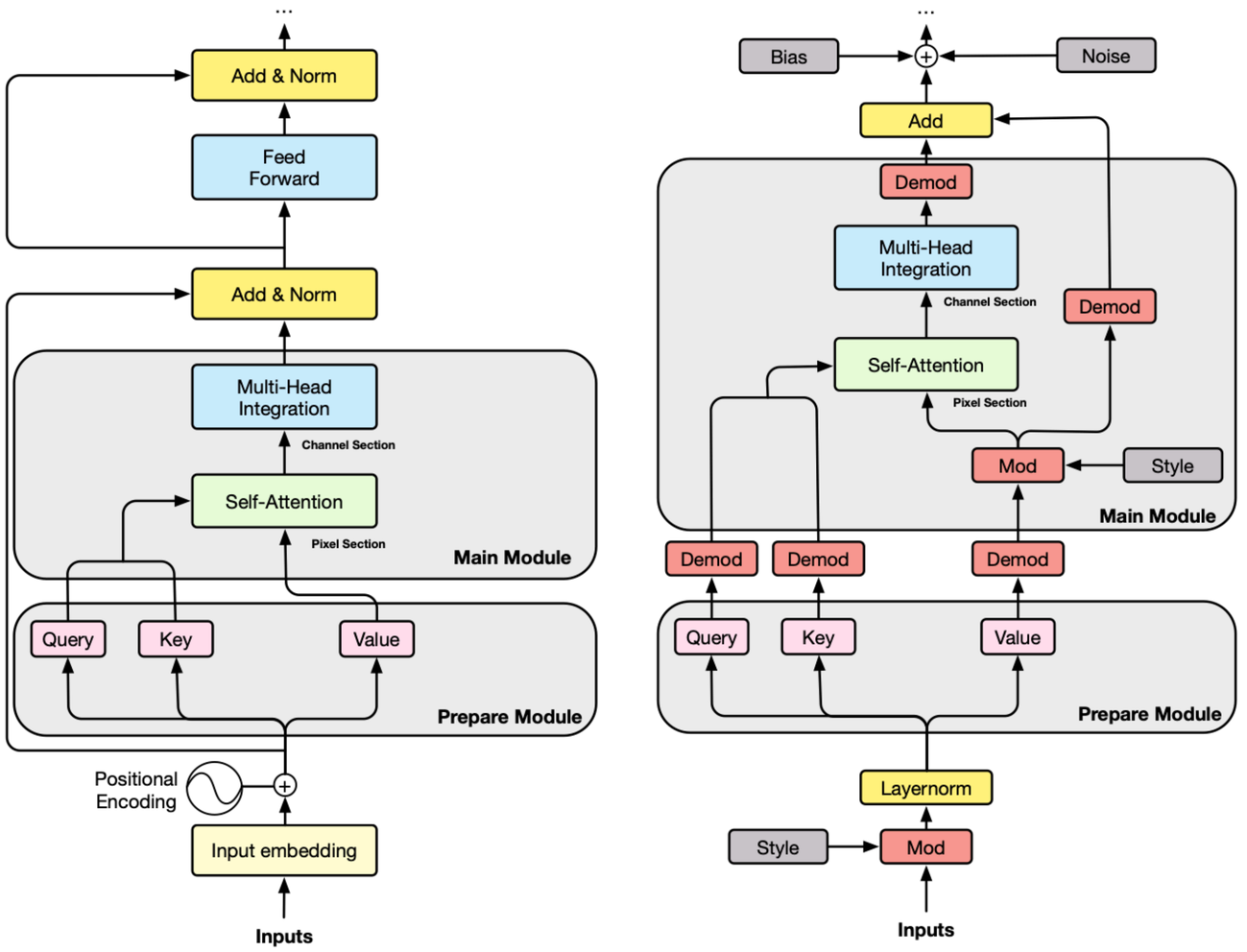


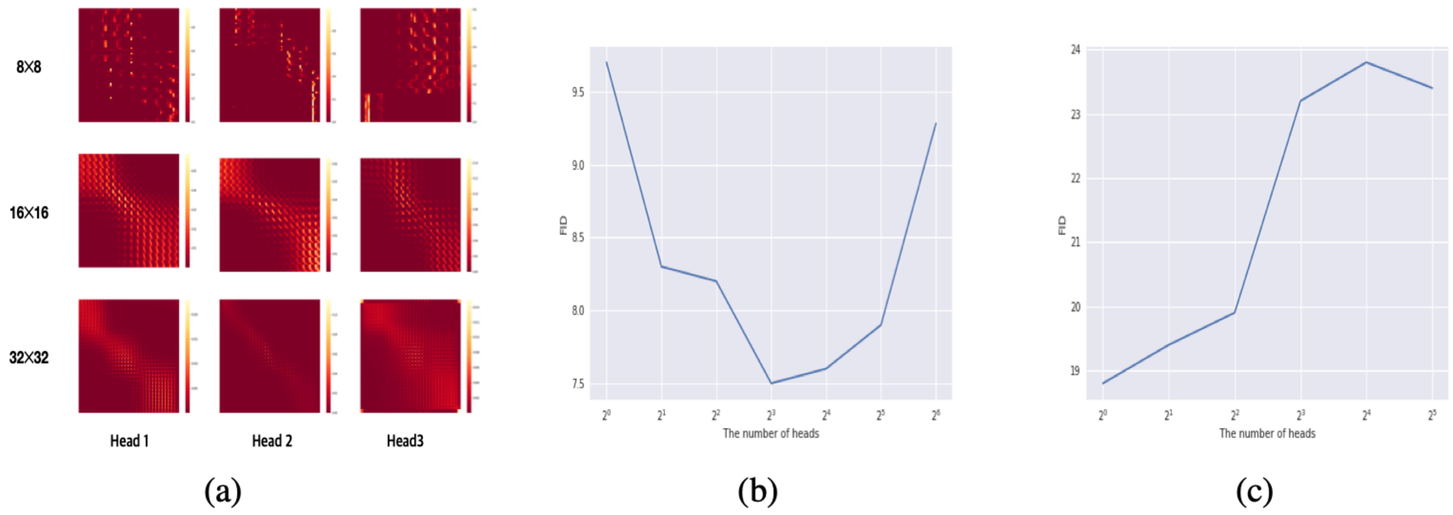
Figure 1: We propose Styleformer, which is a style-based generator for GAN architecture, but a convolution-free transformer-based generator. (a) Original transformer encoder structure. (b) Styleformer encoder structure.

MLP-Mixer

- Pixel section corresponds for the self-attention operation between pixels
- Channel section corresponds for integration of multi-head with linear layer, which operates between channels
- In a transformer, the pixel section is slightly different form depthwise convolution
  - In depthwise convolution, kernel weights exist for each channel
  - In transformer, attention map A acts like one huge kernel, which means applying equal kernel weight to all different channels in V
  - It is difficult to create a powerful generator using a transformer because the same attention kernel is applied for each channel, unlike the generator using depthwise separable convolution

MLP-Mixer

- Using multi-head attention, this problem could be overcome
- We can generate various attention maps by increasing the number of heads
- However, increasing the number of heads inevitably leads to smaller depth, where depth is hidden channel dimension divided by the number of heads
- There exists a minimum depth required
- Figure 2 shows that increaing the number of heads improves performance only when the depth is 32



(a) (b) (c)

Figure 2: (a) Attention maps for Styleformer-Large. The x-axis denotes the number of heads, and the y-axis denotes the resolution to which the attention is applied. (b,c) It shows FID on CIFAR-10 with one layer Styleformer, which hidden dimension size is fixed as 256 and 32, respectively. Both experiments show the best result when the depth is 32.

MLP-Mixer

- Channel section is a layer to integrate the multi-head together
- Unlike convolution, the attention kernel is a kernel generated by the input itself, so it can create a more dense kernel, and it is advantageous to capture global features because it consider the relationship between all pixels
- Therefore, by expanding multi-head attention, the Styleformer can play a more powerful role than depthwise separable convolution, enabling generate high-quality images

Demodulation with Transformer

- StyleGAN generates an image with layer-wise style vectors as inputs
- In generating process, the style vector scales the input feature map for each layer, making certain feature maps amplified
- For scale-specific control, the amplified effect must be removed before entering the next layer
- StyleGAN allows scale-specific control through a normalization operation called AdaIN

Demodulation with Transformer

- StyleGAN2 is an advanced form of StyleGAN and presents the artifact problem caused by the AdaIN operation, solving it by the demodulation operation

- While the AdaIN operation does normalize directly to the output feature map, the demodulation normalization operation is based on statistical assumptions about the input feature map

- Similar to the goal of normalization operation to remove the effect of style vector on output feature map, demodulation operation aims to have an output feature map unit standard deviation while assuming that input feature maps have unit standard deviation

Demodulation with Transformer

- Our goal is to design a transformer-based generator that generates images through style
- Therefore, we propose a method for applying the demodulation operation to a transformer

Demodulation with Transformer

Demodulation for prepare module: creates Q, K , and V

$$w'_{ij} = s_i \cdot w_{ij},$$

Where w is original linear weight to make (query, key, value)

w’ is modulated linear weight

s_i is ith component of style vector

Demodulation with Transformer

- Since the created Q, K, and V become input to the main module, the demodulation operation must proceed, removing the effect of the style vector

$$\sigma_j = \sqrt{\sum_i {w'_{ij}}^2}.$$

- We scale output activations for each dimension of Q, K, and V by 1/σ, making output back to unit standard deviation

Demodulation with Transformer

Demodulation for main module: performs style modulation to input V, multiply with attention map, and then performs linear operation

$$\sigma'_{lk} = \sqrt{\sum_{\cdot} A_{l\cdot}{}^2 \cdot \sum_{j} w'_{jk}{}^2},$$

- w’ is modulated weight
- A_l is attention score vector for lth pixel

Demodulation with Transformer

- However, there are two problems with demodulation by simply scaling each flattened output feature map k with 1/σ’
- First, the concept of this demodulation that normalizes **each pixel** as a unit differs from the AdaIN operation, which normalizes **each feature map** as a unit
- Second, the attention map, which is a matrix derived from the self-attention mechanism, is dependent on the input
  - With input dependent variables, demodulation operations based on statistical assumption can not be applied

Demodulation with Transformer

- Therefore we scale flattened output feature map k with 1/σ”, where $\sigma_k'' = \sqrt{\sum_j {w'_{jk}}^2}$, normalizing each feature map as a unit, and excluding input dependent variables A_l

- Then the standard deviation of output activations will be

$$\sigma_{lk} = \sqrt{\sum_{\cdot} {A_l}_{\cdot}^2}.$$

- However in this way, the problem is that output does not have a unit standard deviation, and actually, standard deviation approaches zero when the numbers of pixels increases

- To prevent this effect, we have applied **residual connenction**

Styleformer Architecture

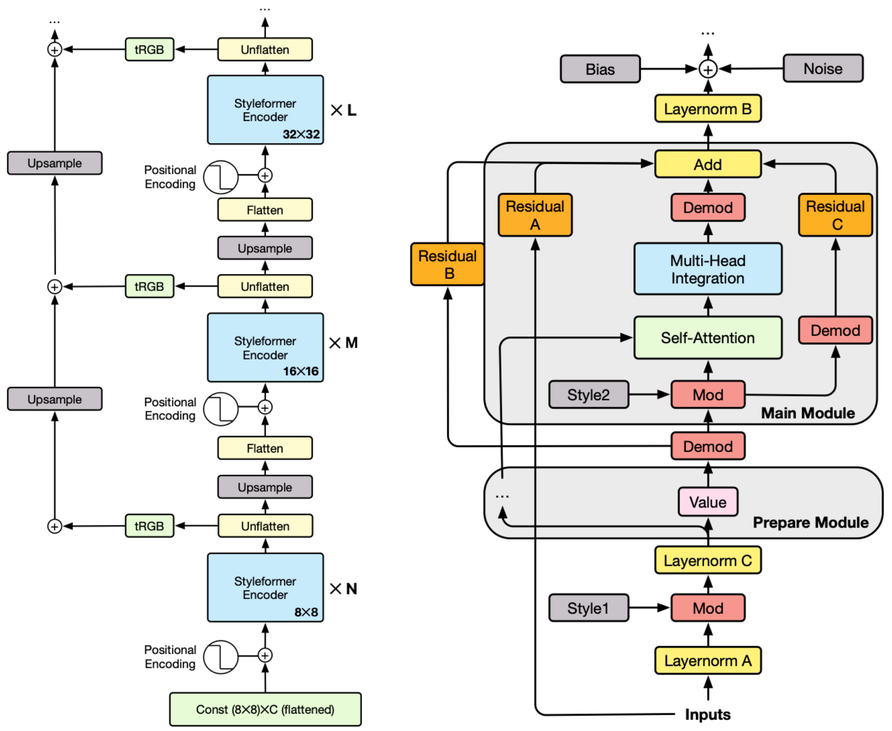


Figure 3: We introduce the overall architecture of the Styleformer and we also show various variants of the Styleformer encoder architecture for convenient ablation study. (a) Overall Architecture of Styleformer. (b) Styleformer encoder structure for ablation.

Styleformer Components

- Style injection
  - Style vector are essential for each operation
  - Silmilar to StyleGAN, we observe there should be different style vector for each module to learn effectively
  - Style 1 is a style vector for prepare module, and Style 2 is a style vector for main module

Styleformer Components

- Residual connection
  - Because style vector should exist for each operation, we determine that residual connection should be carried with style modulation
  - Therefore, we apply residual connection like “Residual C”
  - During the residual connection, we perform demodulation to maintain the standard deviation of input

Styleformer Components

- Layer normalization
  - We also change the position of layer normalization
  - We observe that the role of layer normalization in a transformer is the preparation of generating an attention map by self-attention mechanism and that if we perform layer normalization after tha main module (Layernorm B), style modulation is applied before entering the prepare module at next encoder, which can disturb learning attention map
  - To solve this problem, we proceed with layer normalization after the part of modulation in the prepare module (Layernorm C)

Styleformer Components

- Feed-Forward network
  - We remove the feed-forward structure becasue eliminating it makes the generator more efficient while the performance of the generator increases

Applying Linformer

- The main problem of applying a transformer to image generation is the efficiency problem
- Therefore, we apply Linformer to our model, which projects key and value to the k dimension when applying self-attention, reducing the time and space complexity from O(n^2) to O(nk)
- Linformer explains that this new self-attention mechanism succeeds because the attention map matrix is low-rank
- Applying Linformer creates a more dense attention map, and also reduces computation

Experiments

Table 3: Results on CIFAR-10.

Method	FID ↓	IS ↑
Conditional		
BigGAN [5]	14.73	9.22
FQ-GAN [38]	5.59	8.48 ± 0.03
Unconditional		
Progressive-GAN [23]	15.52	8.80 ± 0.05
AutoGAN [16]	12.42	8.55 ± 0.10
TransGAN-XL [22]	11.89	8.63 ± 0.11
StyleGAN V2 [26]	11.07	9.18
Adversarial NAS-GAN [16]	10.87	8.74 ± 0.07
StyleGAN-ADA [24]	2.92	9.83 ± 0.04
Styleformer-Large	**2.82**	**9.94 ± 0.14**

Table 4: Results on STL-10 and CelebA.

Dataset	Method	FID ↓	IS ↑
STL-10	SN-GAN [31]	40.1	9.16 ± 0.12
	Improving MMD-GAN [36]	37.64	9.23 ± 0.08
	AutoGAN [16]	31.01	9.16 ± 0.12
	Adversarial NAS-GAN [13]	26.98	9.63 ± 0.19
	TransGAN-XL [22]	25.32	10.10 ± 0.17
	Styleformer-Medium	**15.17**	**11.01 ± 0.15**
CelebA	PAE [6]	49.2	-
	BEGAN-CS [7]	34.14	-
	TransGAN-XL [22]	12.23	-
	HDCGAN [9]	8.77	-
	NCP-VAE [1]	5.25	-
	Styleformer	**3.66**	

Experiments

- Applying Linformer for high resolution image

(a) (b)

Figure 4: High resolution figures generated by Styleformer-Linformer. (a) and (b) is generated samples form model trained on CelebA and LSUN-church, respectively.

Table 5: Results on Styleformer which applies Linformer. "Memory" is measured on 4 Titan-RTX with 16 batch size per GPU and "Speed" means seconds for processing 1k images(sec/1kimg). We use the same hidden dimension and the number of layers in Styleformer-Original and Styleformer-Linformer.

Dataset	Model	FID ↓	Memory per GPU ↓	Speed ↓
CelebA	Styleformer-Original	4.84	14668MiB	6.46
	Styleformer-Linformer	**3.66**	5316MiB	4.93
LSUN church	Styleformer-Original	-	OOM	-
	Styleformer-Linformer	**7.99**	8118MiB	9.81