Controlling GAN via PCA of latent vectors

김 현 성

Vision & Display Systems Lab.

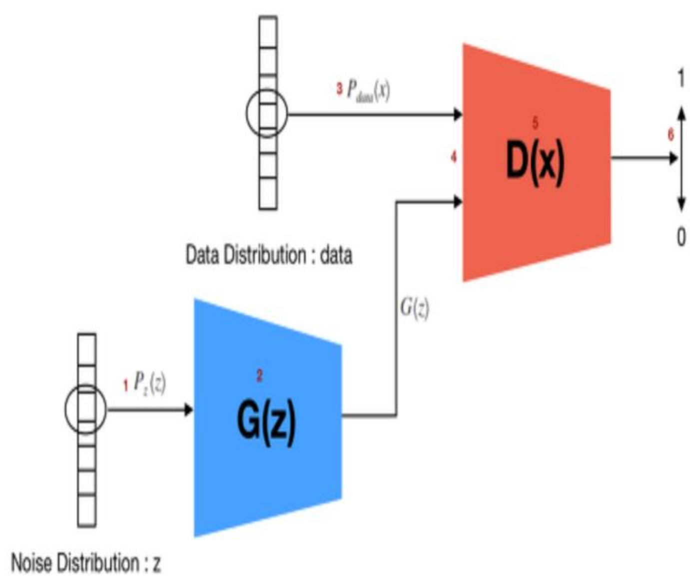
Dept. of Electronic Engineering, Sogang University

Outline

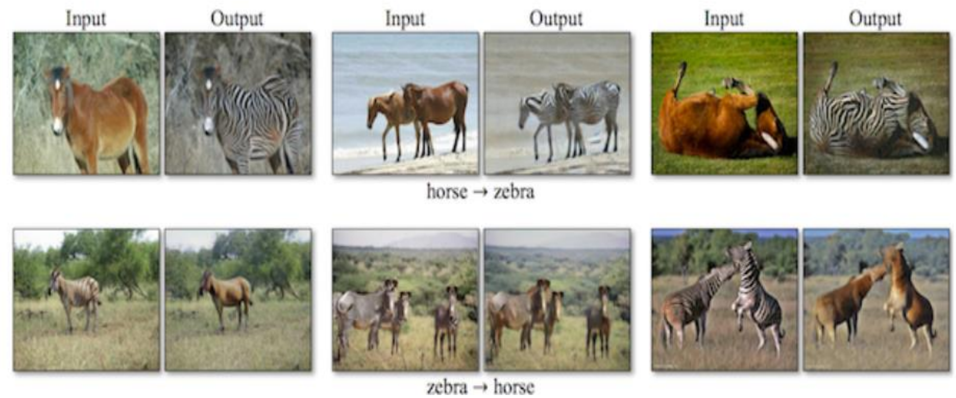
- Introduction
  - GAN: good at image synthesis
  - GAN: controllable?
- Background
  - StyleGAN[1]
  - BigGAN[2]
  - PCA of latent vectors
- GAN and PCA
  - Results and findings
- Conclusion

Introduction

- GAN: good at image synthesis

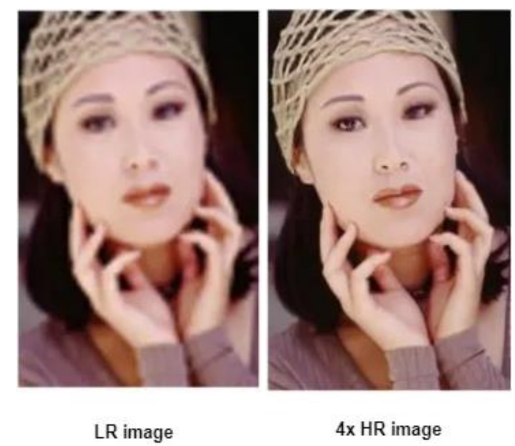


Basic GAN structure



CycleGAN[3]

BigGAN[2]



SRGAN[4]

Introduction

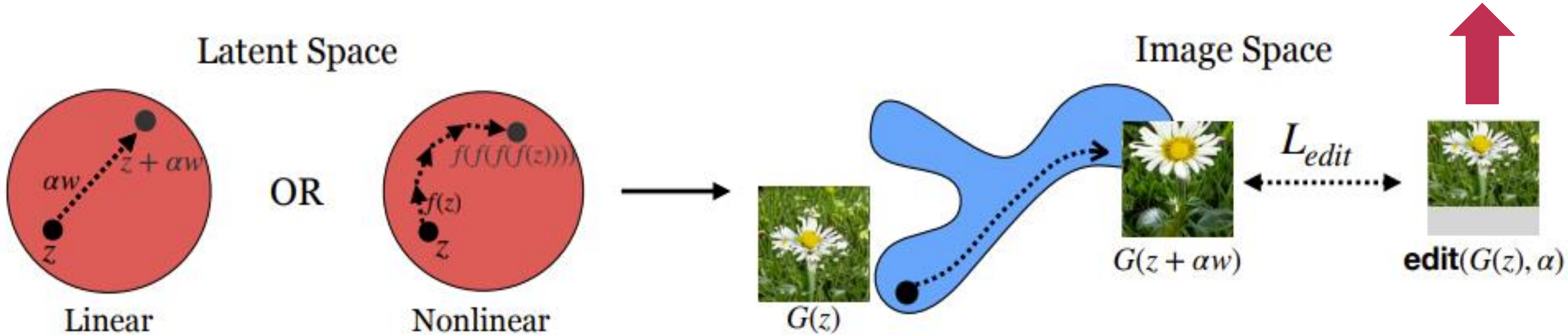
- GAN: controllable?
  - Supervised learning of latent direction[5]
    - Shorten the distance between the generated image after taking "α–step" in the latent direction G(z + αw) and the target edit(G(z), α)
      - New loss for G

$$\mathcal{L}_{edit} = L2\,(G(z+\alpha w) - \texttt{edit}(G(z),\alpha))$$

      - Loss for D

$$\mathcal{L}_{GAN} = \max_{D}\,(\mathbb{E}_{z,\alpha}[D(G(z+\alpha w))] - \mathbb{E}_{x,\alpha}[D(\texttt{edit}(x,\alpha))])$$

Need additional object detector



서강대학교 SOGANG UNIVERSITY

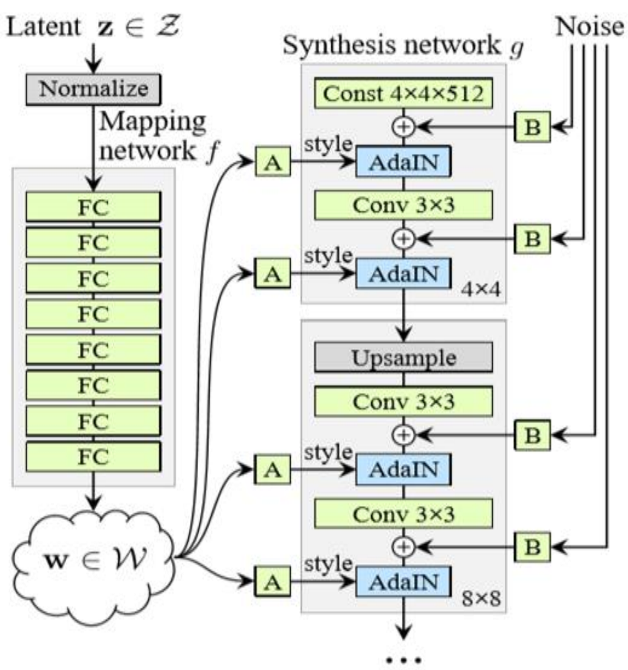
VDS LAB

Introduction

- GAN: controllable?
  - Unsupervised identification of interpretable directions

Background

- StyleGAN[1]
  - Network

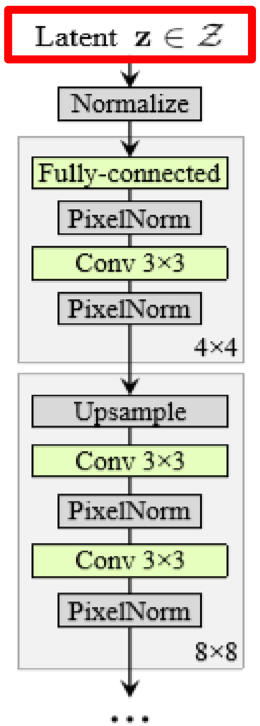


  - Experimental results

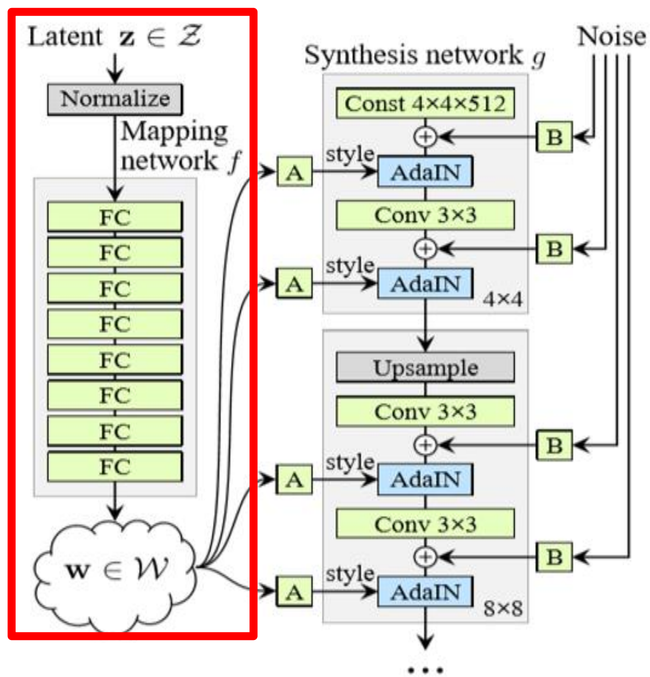
Not-curated set of images in 1024^2 resolution

Background

- StyleGAN[1]
  - Mapping Network



Traditional



StyleGAN

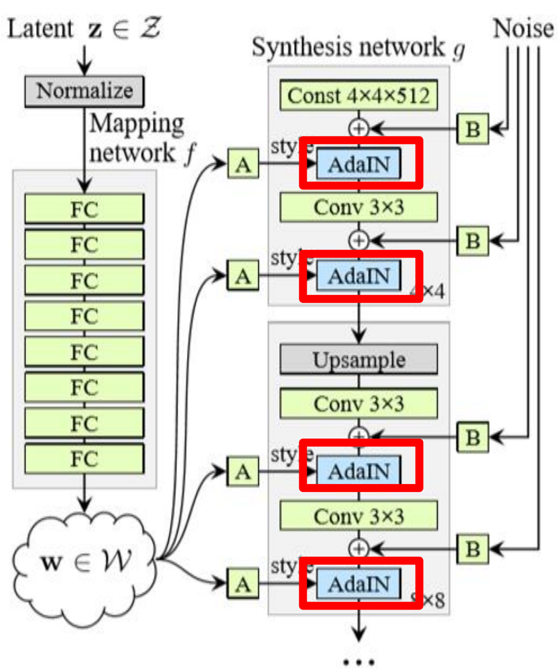
$\boldsymbol{z} \in Z$
(latent code, 512 dim.)

Non-linear Mapping

$\boldsymbol{w} \in W$
(Intermediate latent space, 512 dim.)

Background

- StyleGAN[1]
  - Adaptive Instance Normalization



StyleGAN

Normalize first using momentums taken across spatial axes

✓ $$\boldsymbol{AdaIN}(\boldsymbol{x}_i, \boldsymbol{y}) = \boldsymbol{y}_{s,i} \frac{\boldsymbol{x}_i - \mu(\boldsymbol{x}_i)}{\sigma(\boldsymbol{x}_i)} + \boldsymbol{y}_{b,i}$$

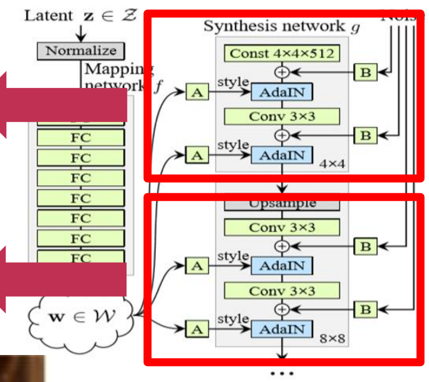
✓ $\boldsymbol{w} \in \boldsymbol{W}$ (Intermediate latent space, 512 dim.) **FC-Layer** → $\boldsymbol{concat}(\boldsymbol{y}_s, \boldsymbol{y}_b)$ (scale & bias parameter, 2*C dim.)

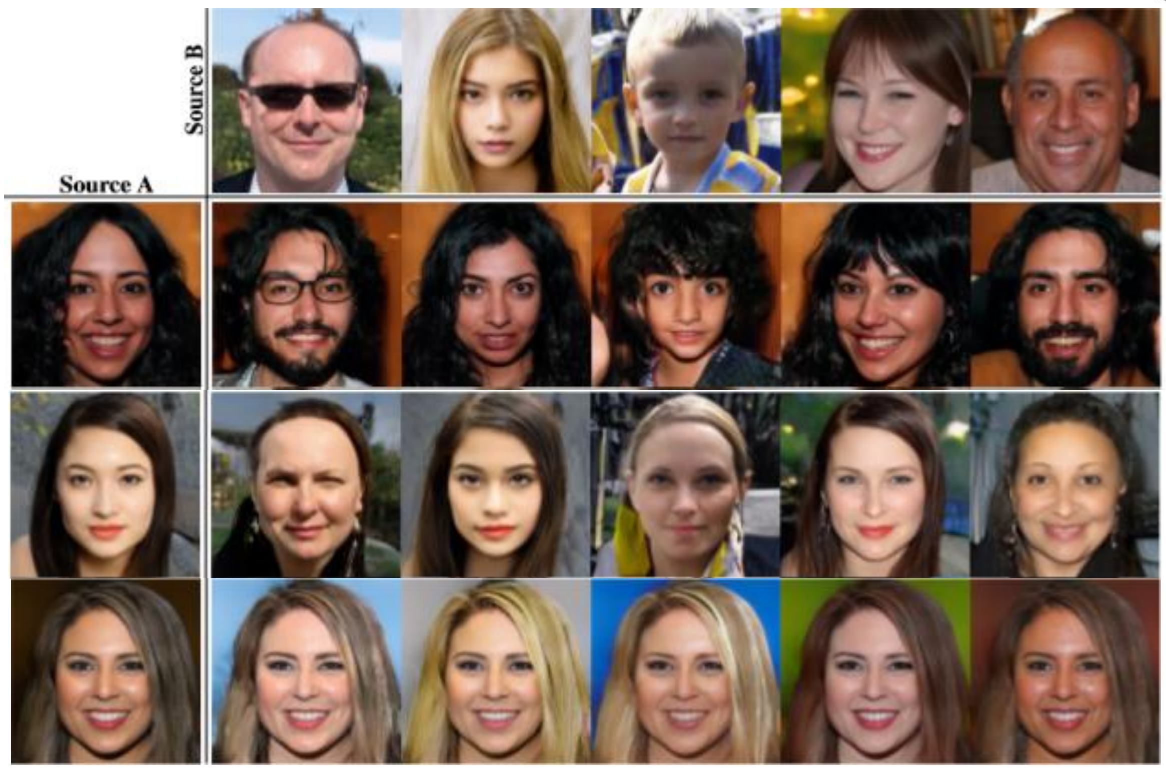
Background

- StyleGAN[1]
  - Style Mixing

$z_1 \sim p(z) \to w_1$

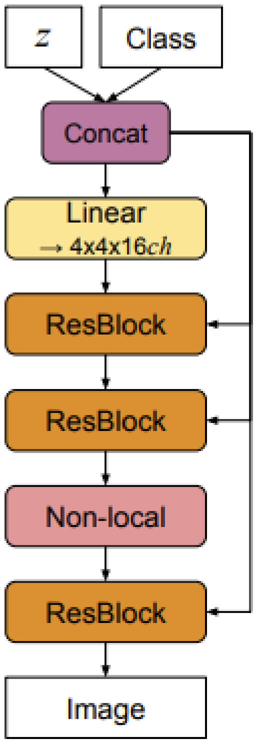
$z_2 \sim p(z) \to w_2$





Background

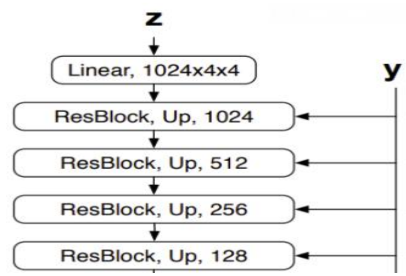
- BigGAN[2]
  - Feed-Forward Network

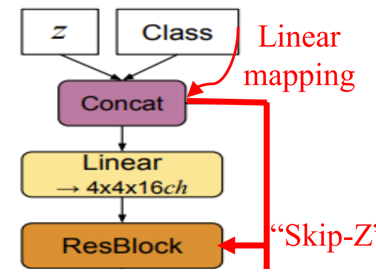


  - Experimental results

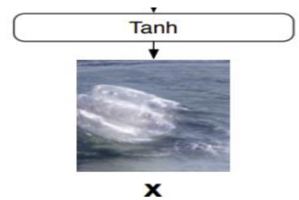
Background

- BigGAN[2]
  - Batch-size & Number of channels
  - Shared class embedding
  - Skip-z connection
  - Orthogonal regularization

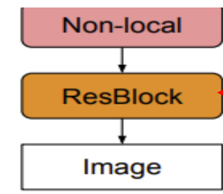




$$R_\beta(W) = \beta \| W^\top W \odot (\mathbf{1} - I) \|_F^2$$



From [6]



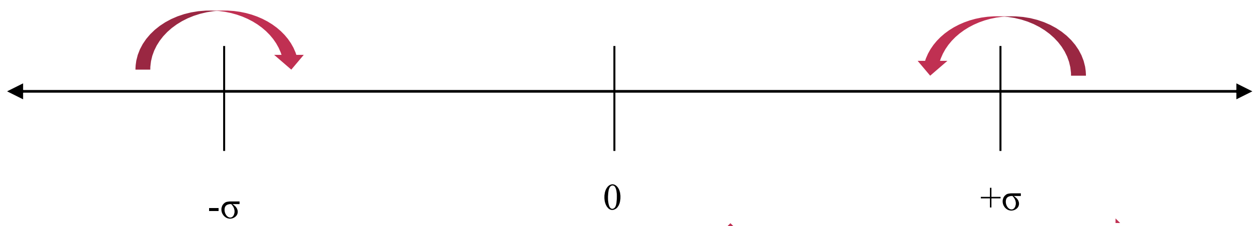
Batch	Ch.	Param (M)	Shared	Skip-z	Ortho.	Itr $\times 10^3$	FID	IS
256	64	81.5	SA-GAN Baseline			1000	18.65	52.52
512	64	81.5	✗	✗	✗	1000	15.30	58.77(±1.18)
1024	64	81.5	✗	✗	✗	1000	14.88	63.03(±1.42)
2048	64	81.5	✗	✗	✗	732	12.39	76.85(±3.83)
2048	96	173.5	✗	✗	✗	295(±18)	9.54(±0.62)	92.98(±4.27)
2048	96	160.6	✓	✗	✗	185(±11)	9.18(±0.13)	94.94(±1.32)
2048	96	158.3	✓	✓	✗	152(±7)	8.73(±0.45)	98.76(±2.84)
2048	96	158.3	✓	✓	✓	165(±13)	8.51(±0.32)	99.31(±2.10)
2048	64	71.3	✓	✓	✓	371(±7)	10.48(±0.10)	86.90(±0.61)

Background

- BigGAN[2]
  - Orthogonal regularization
    - Truncation trick
      - Resample components of z whose magnitudes are out of range [-threshold , threshold]

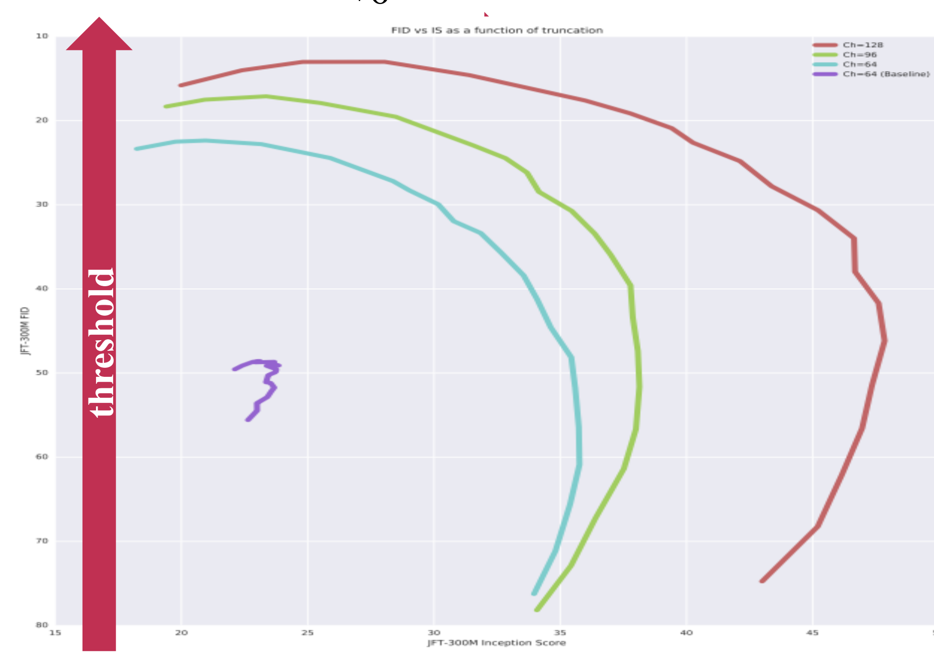
Model	Res.	FID/IS	(min FID) / IS	FID / (valid IS)	FID / (max IS)
SN-GAN	128	27.62/36.80	N/A	N/A	N/A
SA-GAN	128	18.65/52.52	N/A	N/A	N/A
BigGAN	128	$8.7 \pm .6/98.8 \pm 3$	$7.7 \pm .2/126.5 \pm 0$	$9.6 \pm .4/166.3 \pm 1$	$25 \pm 2/206 \pm 2$
BigGAN	256	$8.7 \pm .1/142.3 \pm 2$	$7.7 \pm .1/178.0 \pm 5$	$9.3 \pm .3/233.1 \pm 1$	$25 \pm 5/291 \pm 4$
BigGAN	512	8.1/144.2	7.6/170.3	11.8/241.4	27.0/275
BigGAN-deep	128	$5.7 \pm .3/124.5 \pm 2$	$6.3 \pm .3/148.1 \pm 4$	$7.4 \pm .6/166.5 \pm 1$	$25 \pm 2/253 \pm 11$
BigGAN-deep	256	$6.9 \pm .2/171.4 \pm 2$	$7.0 \pm .1/202.6 \pm 2$	$8.1 \pm .1/232.5 \pm 2$	$27 \pm 8/317 \pm 6$
BigGAN-deep	512	7.5/152.8	7.7/181.4	11.5/241.5	39.7/298

Score on ImageNet



Ch.	Param (M)	Shared	Skip-z	Ortho.	FID	IS	(min FID) / IS	FID / (max IS)
64	317.1	✗	✗	✗	48.38	23.27	48.6/23.1	49.1/23.9
64	99.4	✓	✓	✓	23.48	24.78	22.4/21.0	60.9/35.8
96	207.9	✓	✓	✓	18.84	27.86	17.1/23.3	51.6/38.1
128	355.7	✓	✓	✓	13.75	30.61	13.0/28.0	46.2/47.8

(a)



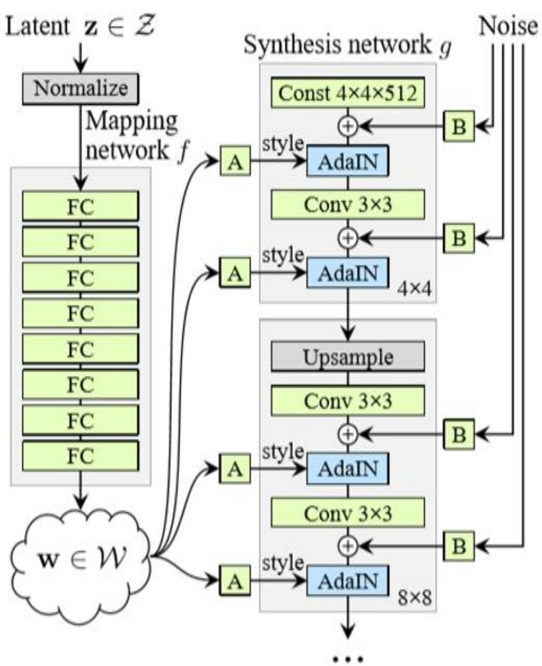
Background

- BigGAN[2]
  - Other experiments…
    - Parameter initialization
      - $N(0, 1)$, $u(-1, 1)$, $Bernoulli\{0, 1\}$, $\max(N(0, 1), 0)$, …
    - Instability: Generator
      - Importance of top-three singular values of each matrix
      - How to counteract spectral explosion
      - Which value is good for clamping the first singular value
    - Instability: Discriminator
      - Why the loss of D jumps when training collapse
      - D is memorizing the training set?

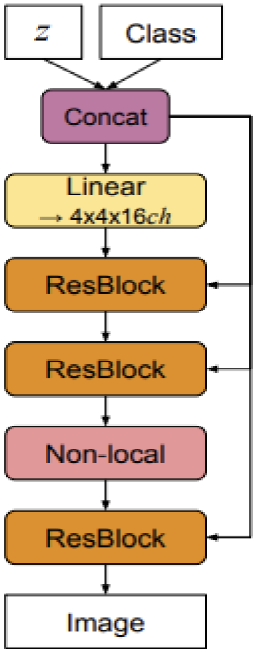
Background

- StyleGAN[1] & BigGAN[2]
  - Inference
    - BigGAN
      - $y_i = G_i(y_{i-1}, \boldsymbol{z})$
    - StyleGAN
      - $y_i = G_i(y_{i-1}, \boldsymbol{w})$ with $\boldsymbol{w} = M(z)$
  - Able to mix styles?
    - StyleGAN
      - yes
    - BigGAN
      - no

$\boldsymbol{z}$: *noise vector*
G_i : i th layer of G
y_i : *output of* G_i
M : 8 − *MLP*



StyleGAN



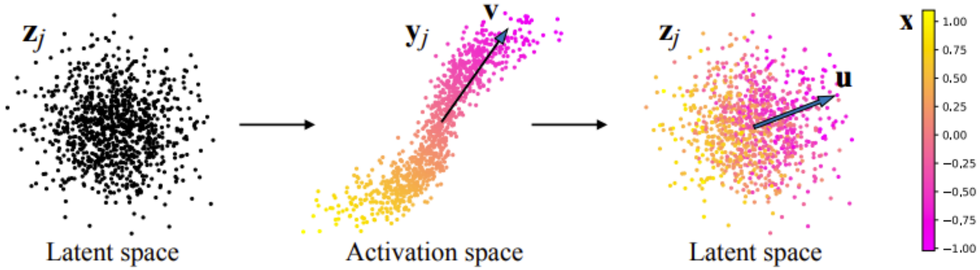
BigGAN

Background

- PCA of latent vectors
  - StyleGAN[1]
    - Goal is to identify the principal axes of p(**w**)
      - Since the distribution of **z** is not learned, which distribution of **z** is isotropic
    - How to do?
      - Sample N random vectors $\boldsymbol{z}_{1:N}$
      - Compute $\boldsymbol{w}_i = \mathrm{M}(\boldsymbol{z}_i)$
      - Compute PCA of these $\boldsymbol{w}_{1:N}$ values
      - Get matrix **V**: basis matrix
    - Edit **w**
      - $\boldsymbol{w}' = \boldsymbol{w} + \boldsymbol{V}\boldsymbol{x}$
        - ✓ **x** is composed of constants to control the value of **w** on new axes, and set by user
      - Assume only one **w** is used for image generation

Background

- PCA of latent vectors
  - BigGAN[2]



– Perform PCA at an intermediate network layer i

- The output of 1st layer is used, as the performance was better than using any other output tensor

– How to do?

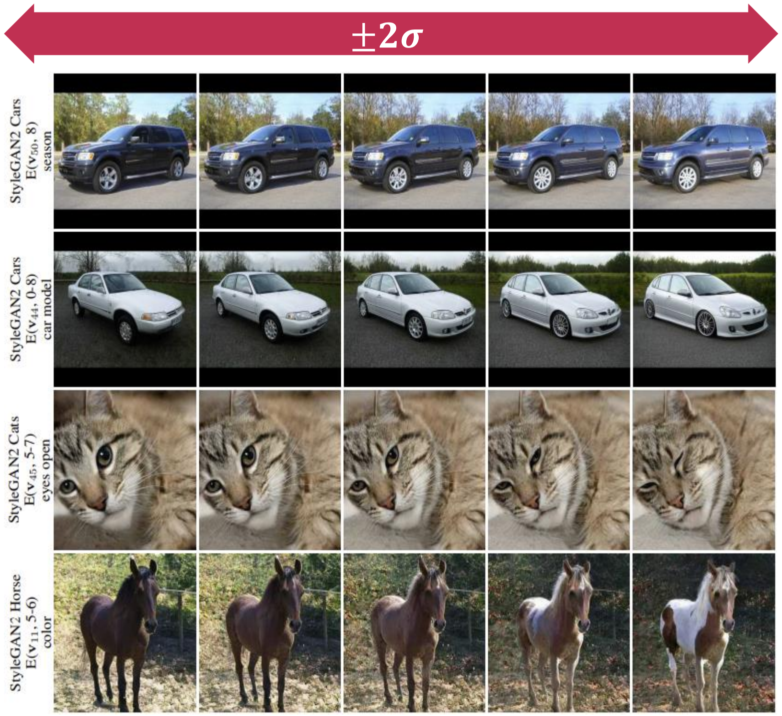
- Sample N random vectors $\boldsymbol{z}_{1:N}$
- Compute $\boldsymbol{y}_j = G_i(\boldsymbol{z}_j)$
- Compute PCA of these $\boldsymbol{y}_{1:N}$ values
  - ✓Get low-rank basis matrix V, data mean μ
- Get PCA coordinates $\boldsymbol{x}_j$ of each feature tensor: $\boldsymbol{x}_j = \boldsymbol{V}^{\boldsymbol{T}}(\boldsymbol{y}_j - \mu)$
- $\boldsymbol{u}_k = \text{argmin} \sum_j ||\boldsymbol{u}_k x_j^k - \boldsymbol{z}_j||^2$, where x_j^k is the k th value of $\boldsymbol{x}_j$

– Edit $\mathbf{z}$

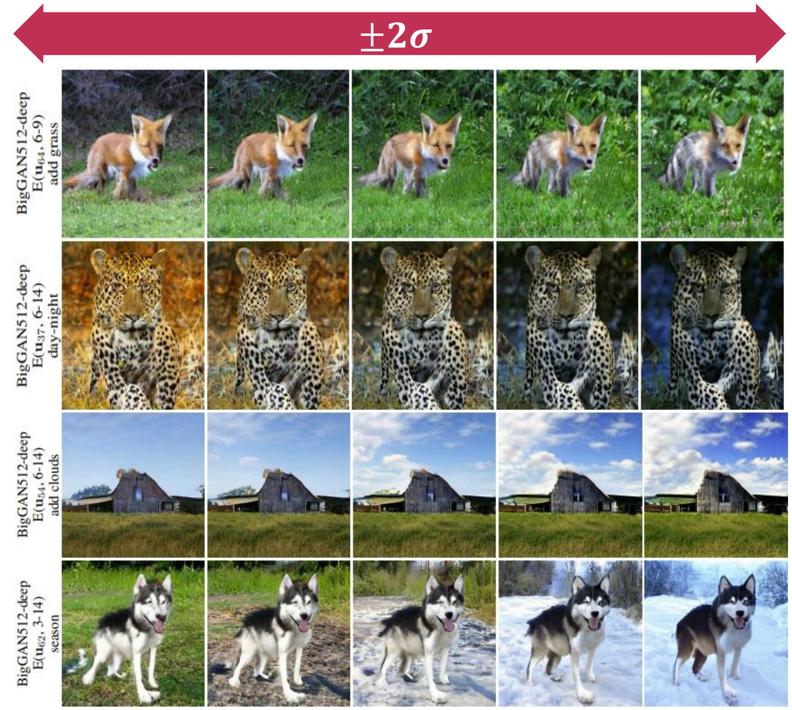
- $\boldsymbol{z}' = \boldsymbol{z} + \boldsymbol{U}\boldsymbol{x}$
  - ✓ $\mathbf{x}$ is composed of constants to control the value of $\mathbf{z}$ on new axes, and set by user

GAN and PCA

- ***Results*** and findings
  - Find specific principal axes & layers
    - $E(v_j, start - end)$: edit latent vector using j th principal axis from 'start' layer to 'end' layer



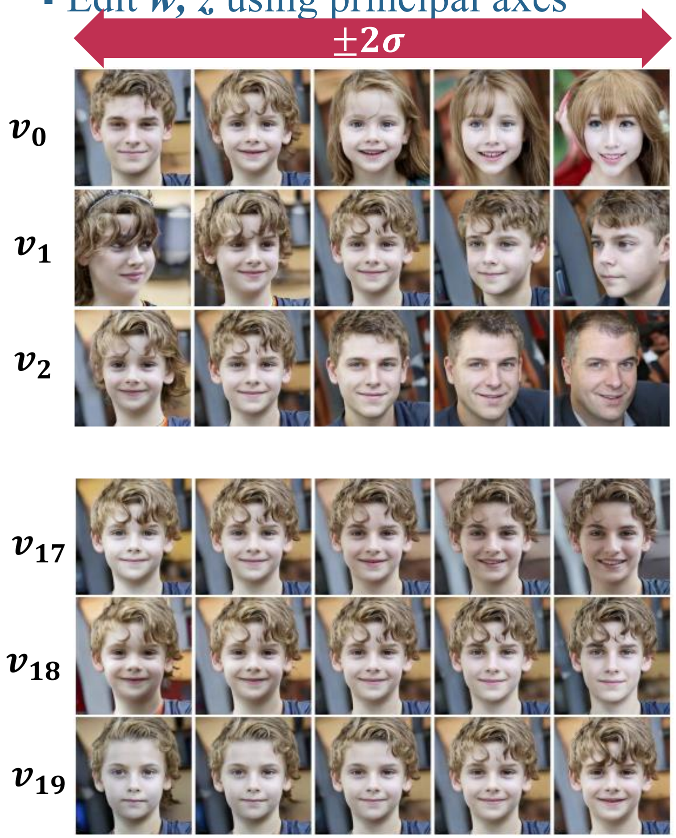
StyleGAN



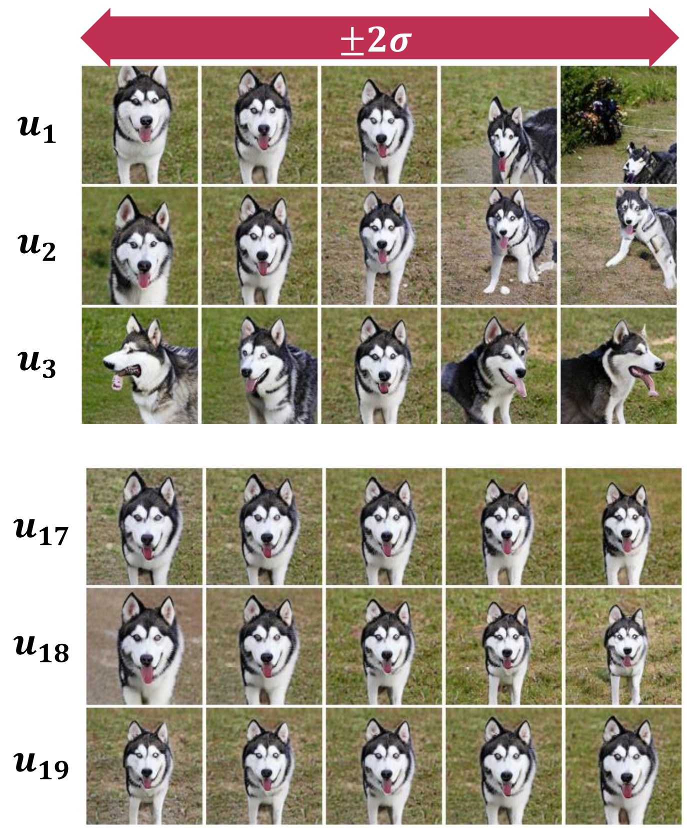
BigGAN

GAN and PCA

- ***<u>Results</u>*** and findings
  - Edit ***w***, ***z*** using principal axes



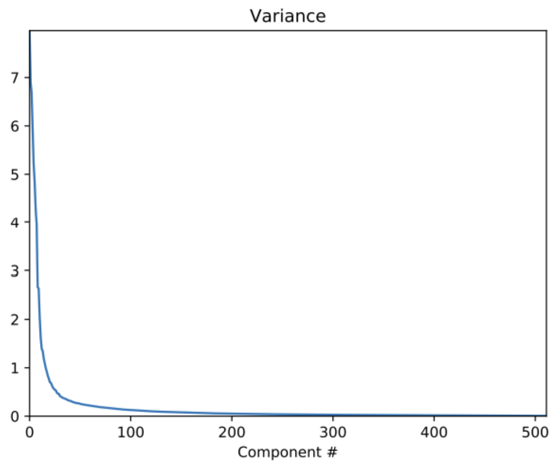
StyleGAN



BigGAN

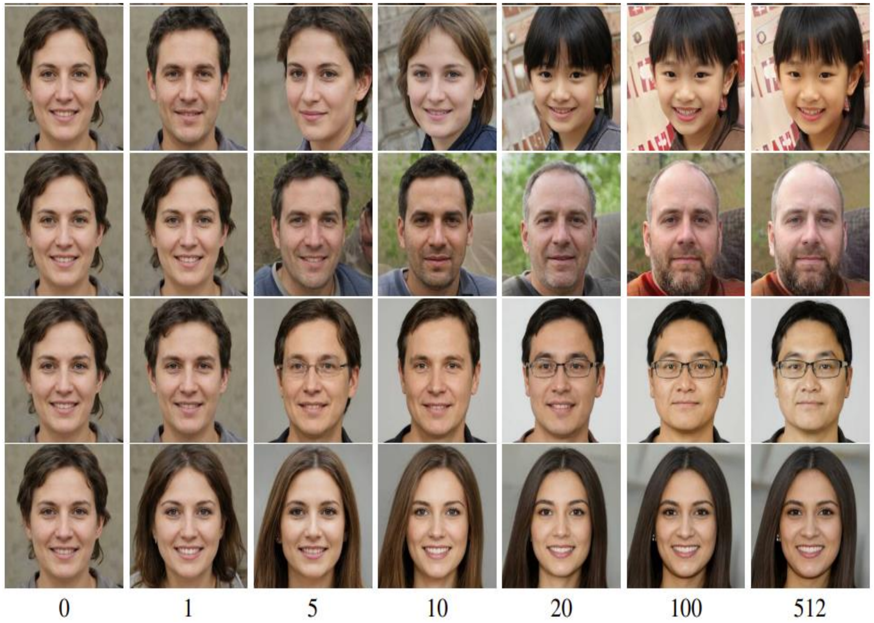
GAN and PCA

- Results and ***findings***
  - How many dimensions are important to image synthesis?
    - Investigation of variance captured in each dimension of the PCA for the FFHQ StyleGAN model
      - First 100 dimensions capture 85% of the variance; first 200 dimensions capture 92.5%



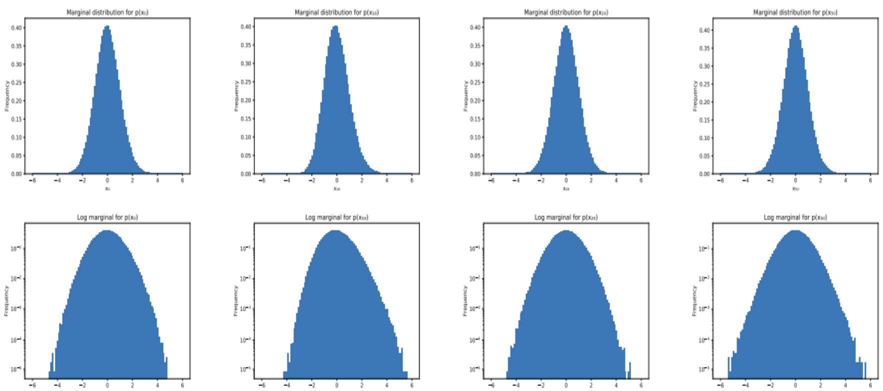
$$\boldsymbol{w} \leftarrow \mathbf{V}_K \mathbf{V}_K^T(\boldsymbol{w} - \boldsymbol{\mu}) + \boldsymbol{\mu},$$

where V_K are the columns for the first K principal components



GAN and PCA

- Results and ***findings***
  - Where are the ***ws***?
    - Investigation of distribution of ***w***: ***p(w)***
      - By investigating of $p(x^0, x^1, \ldots, x^{511})$



Marginal distribution of $\boldsymbol{x^0}$, $\boldsymbol{x^{18}}$, $\boldsymbol{x^{20}}$, $\boldsymbol{x^{50}}$

Mutual Information of $\boldsymbol{x^j}$, $\boldsymbol{x^k}$

$$= \mathrm{KL}(\mathrm{p}(x^j, x^k) \,\|\, \mathrm{p}(x^j)\mathrm{p}(x^k))$$

$$= -H(x^j, x^k) + H(x^k) + H(x^j)$$

$$MI(x^j, x^k) \in [6.9, 8.7]\,, if\ j = k$$
$$\in [0, 0.3]\,, else$$

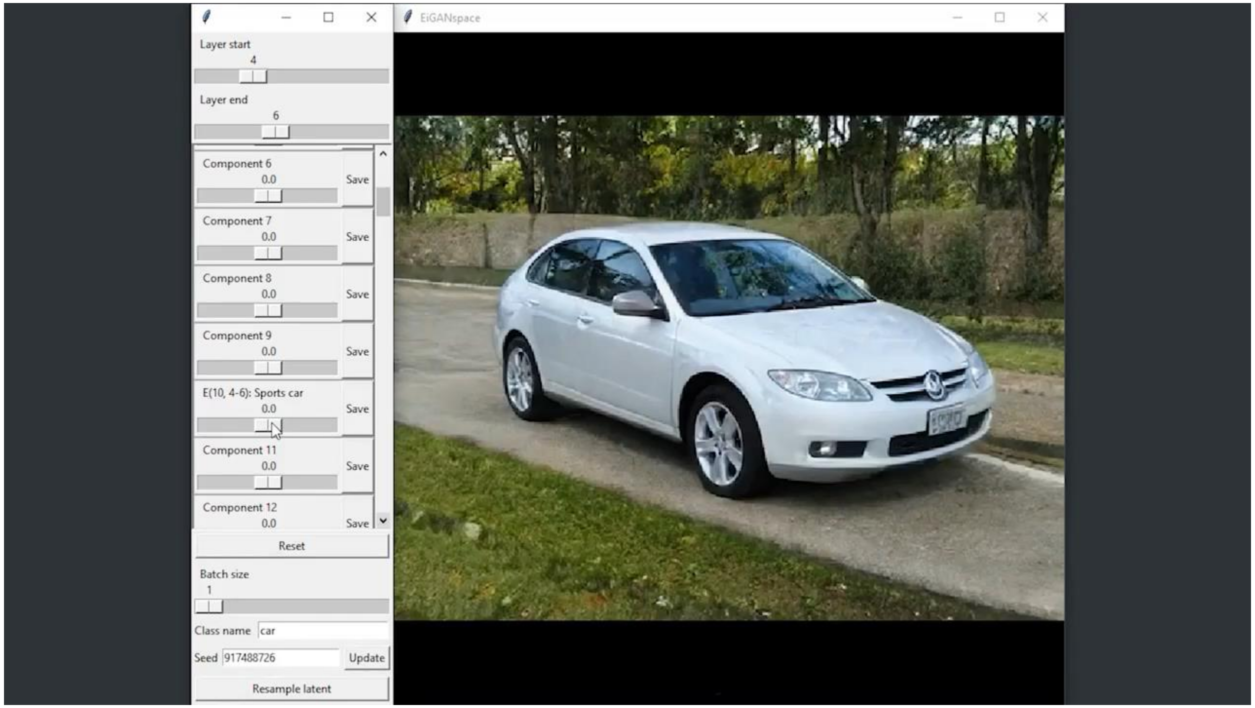
Almost Independent!

$$x^j \sim p(x^j)$$

$$\mathbf{y} = \mathbf{V}\mathbf{x} + \mu$$

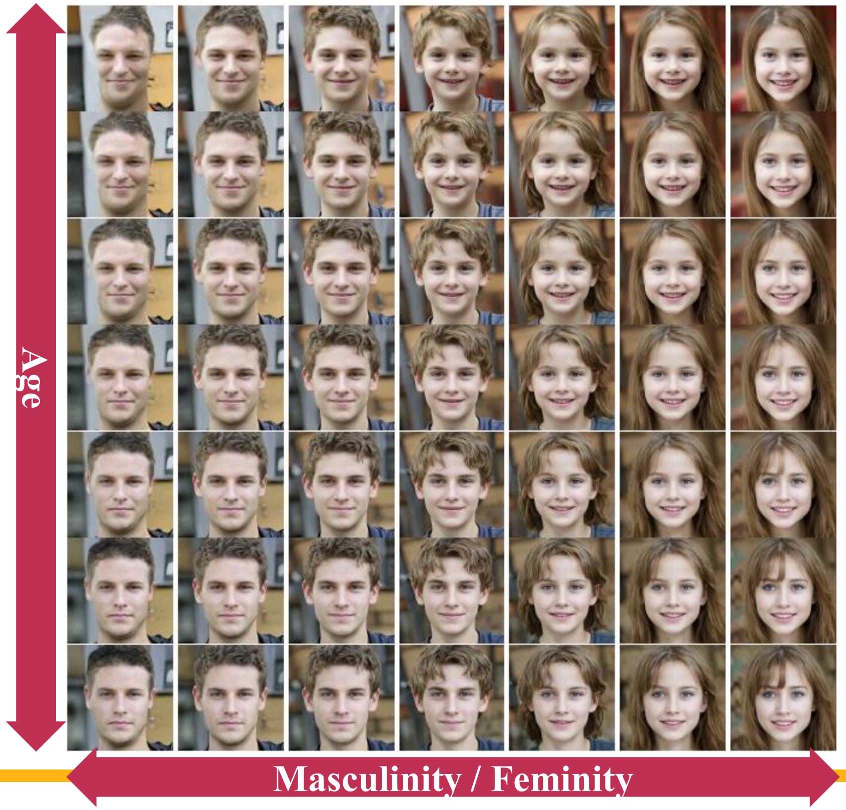
GAN and PCA

- Results and ***findings***
  - Undesirable properties inherited from GAN's training set
    - Entanglement



GAN and PCA

- Results and ***<u>findings</u>***
  - Undesirable properties inherited from GAN's training set
    - Disallowed combination



Conclusion

- **GAN control using PCA**
  - We can control contents of images generated by GAN whose layers share same input like **z, w** in BigGAN, StyleGAN.
  - Even, for StyleGAN, we can get **w** w/o computing 8-MLP when we do inference.
- **Limitations**
  - Heavily depending on GAN's training dataset.
  - Need to find every direction & layers for specific control.

Reference

1. T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In Proc. CVPR, 2019.
2. A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In Proc. ICLR, 2019.
3. J. Zhu, T. Park, P. Isola, A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. ICCV 2017
4. C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunnungham, A. Acosta, A. Aitken, A. Tejani, J., Totz, Z. Wang, W. Shi. Pthoto-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
5. A. Jahanian, L. Chai, and P. Isola. On the "steerability" of generative adversarial networks. In Proc. ICLR, 2020.
6. Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In ICLR, 2017.