Generative Networks; Unconditional GANs

YouTube Playlist

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

Boulder

Generative Adversarial Nets

YouTube Video

$p_z(z) \rightarrow$ prior on noise variable z

$G(z;\theta_g) \rightarrow$ generative model (differentiable function)

$p_g(x) \rightarrow$ generator's distribution over data x

$D(x;\theta_d) \rightarrow$ discriminative model

(probability that x came from the data rather than p_g)

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

$\log(1 - D(G(z))) \rightarrow$ saturates early in the learning

(D can reject generated samples with high confidence)

$$D(G(z)) \approx 0 \implies 1 - D(G(z)) \approx 1 \implies \log(1 - D(G(z))) \approx 0$$

$$\max_G \mathbb{E}_{z\sim p_z(z)} \log D(G(z))$$

Proposition: For fixed G, the optimal discriminator D is

$$D^*_G(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

Proof:

$$V(G,D) = \int_x \left[p_{\text{data}}(x) \log D(x) + p_g(x) \log(1 - D(x))\right] dx$$

$$y^* = \frac{a}{a+b} = \arg\max_y \left[a \log y + b \log(1-y)\right]$$

$$C(G) := \max_D V(G,D)$$

$$C(G) = \mathbb{E}_{x\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}\right] + \mathbb{E}_{x\sim p_g}\left[\log \frac{p_g(x)}{p_{\text{data}}(x) + p_g(x)}\right]$$

$$C(G) = -\log 4 + KL\left(p_{\text{data}} \Big\| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \Big\| \frac{p_{\text{data}} + p_g}{2}\right)$$

$$C(G) = -\log 4 + \underbrace{2JSD\left(p_{\text{data}} \| p_g\right)}$$

Jensen–Shannon divergence; ≥ 0; $= 0$ iff $p_{\text{data}} = p_g$

a) b) c) d)

Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.

Unsupervised representation learning with deep convolutional generative adversarial networks

Boulder

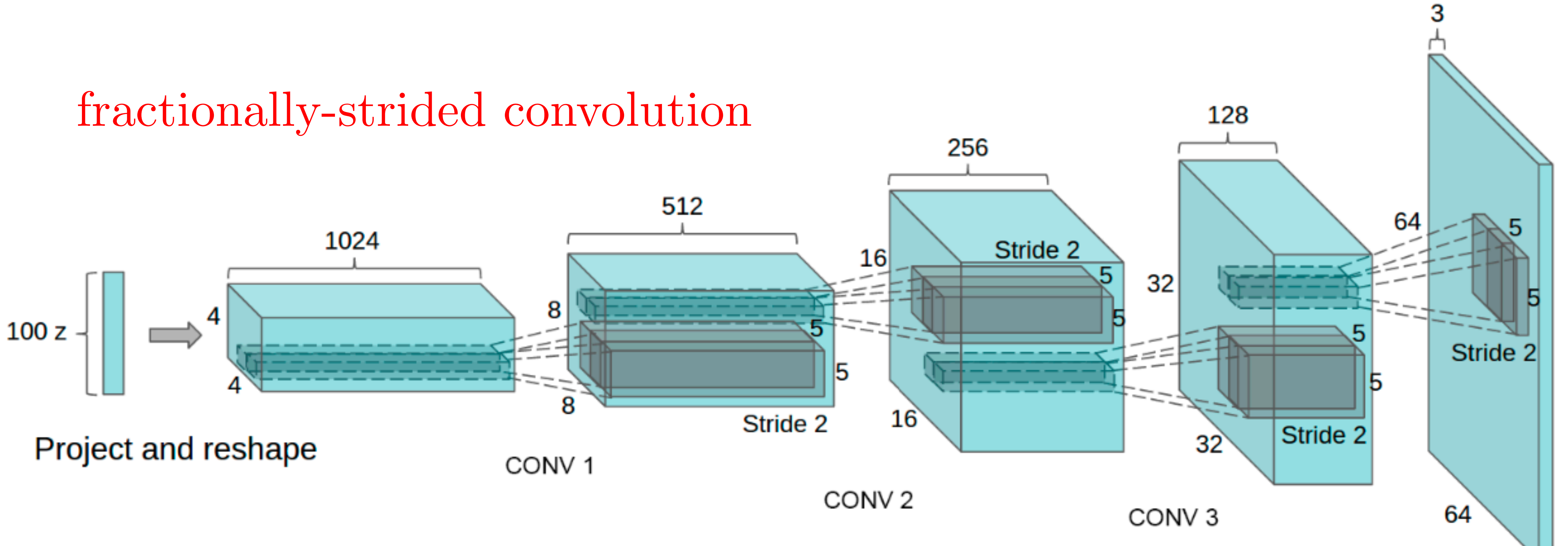
YouTube Video

DCGANs $\rightarrow$ deep convolutional generative adversarial networks

Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

fractionally-strided convolution



$$Y(x,y) = \sum_{i=1,2,3} \sum_{j=1,2,3} W(i,j) X(\underbrace{(x-i)/s+2}, \underbrace{(y-j)/s+2})$$

Only integer-valued indices participate in the summation!

$$Y(x,y) = \sum_{i=1,2,3} \sum_{j=1,2,3} W(i,j) X(\underbrace{s(x-2)+i}_{=:\,x'}, s(y-2)+j)$$

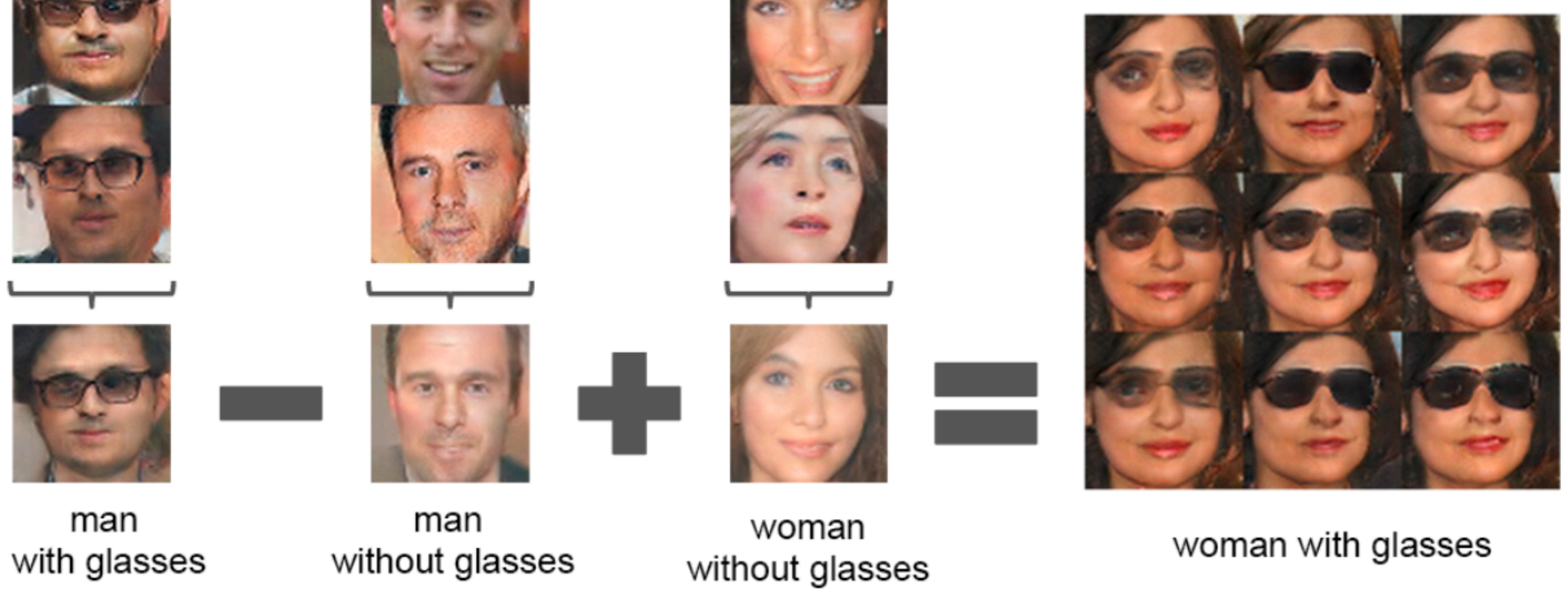
$$\implies x = (x'-i)/s + 2$$

$$\frac{\partial L}{\partial X(x',y')} = \sum_{i=1,2,3} \sum_{j=1,2,3} W(i,j)^T \frac{\partial L}{\partial Y((x'-i)/s+2, (y'-j)/s+2)}$$

DCGAN trained on Imagenet-1k. Features are used to classify CIFAR-10 images.

Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% ($\pm$0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% ($\pm$0.7%)	3200
View Invariant K-means	81.9%	72.6% ($\pm$0.7%)	6400
Exemplar CNN	84.3%	77.4% ($\pm$0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% ($\pm$0.4%)	512

discriminator's convolutional features from all layers, maxpooling each layers representation to produce a 4 × 4 spatial grid



Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

Boulder

Improved Techniques for Training GANs

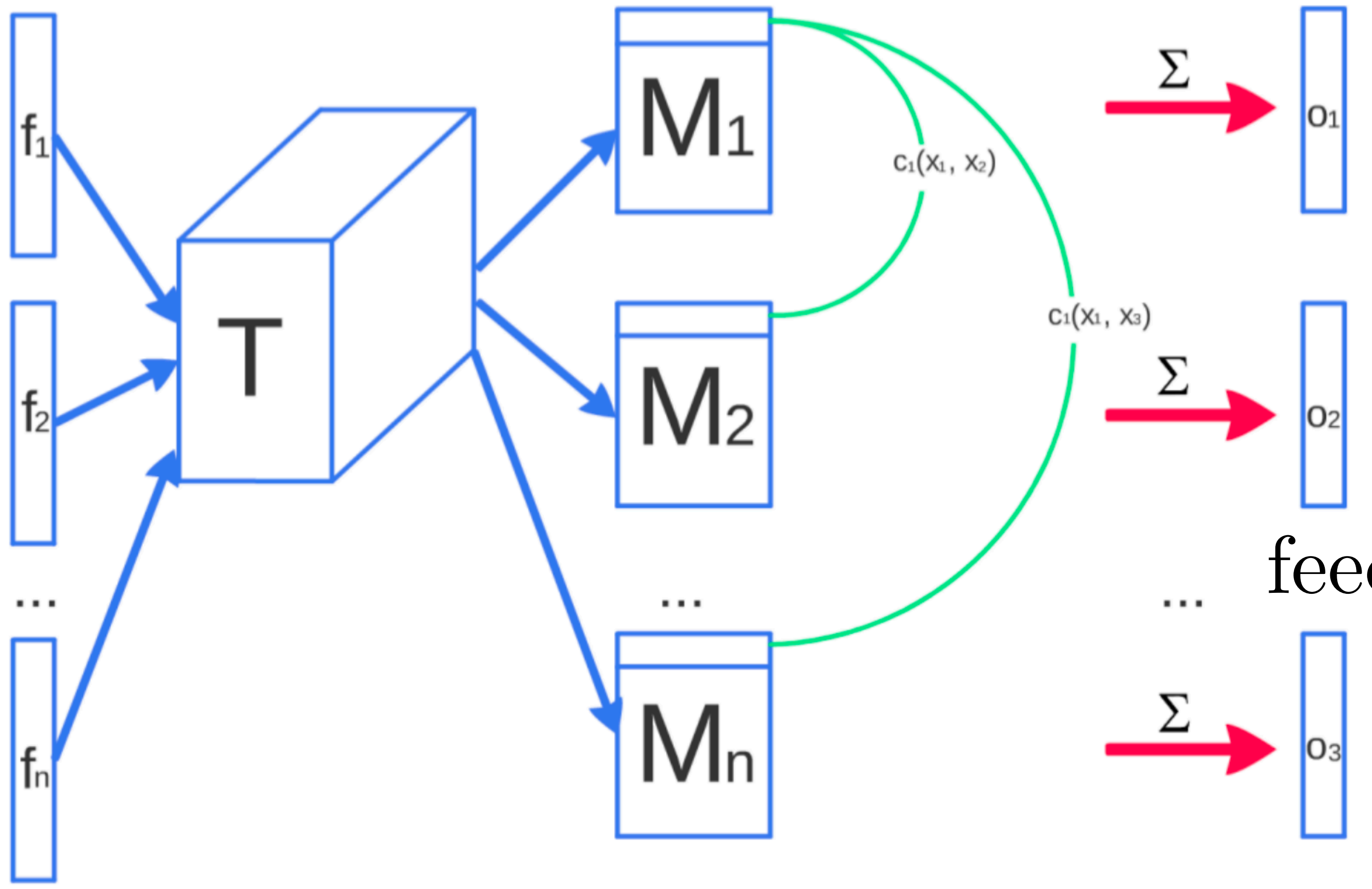
YouTube Playlist

Feature Matching

$f(x) \rightarrow$ activations on an intermediate layer of the discriminator

$\|\mathbb{E}_{x\sim p_{\text{data}}} f(x) - \mathbb{E}_{z\sim p_z(z)} f(G(z))\|_2^2 \rightarrow$ new objective for the generator

Minibatch discrimination (to deal with mode-collapse)



feed $[f(x_i), o(x_i)]$ to the next layer

$M_i = f(x_i)T, f(x_i) \in \mathbb{R}^A, F \in \mathbb{R}^{A\times B\times C}, M_i \in \mathbb{R}^{B\times C}$

$c_b(x_i, x_j) = \exp(-\|M_{i,b} - M_{j,b}\|_{L_1}) \in \mathbb{R}, M_{i,b} \in \mathbb{R}^C$

$o(x_i)_b = \sum_{j=1}^{n} c_b(x_i, x_j) \in \mathbb{R} \rightarrow$ looks at multiple examples

$o(x_i) = [o(x_i)_1, \ldots, o(x_i)_B] \in \mathbb{R}^B \rightarrow$ output

Historical averaging

$$\|\theta - \frac{1}{t}\sum_{i=1}^{t}\theta[i]\|_2^2$$

One-sided label smoothing

$$D(x) = \frac{\alpha p_{\text{data}}(x) + \beta p_g}{p_{\text{data}}(x) + p_g(x)}$$

$0 \leftrightarrow \beta = 0; 1 \leftrightarrow \alpha$

Virtual batch normalization

Each example x is normalized based on the statistics collected on a $\underbrace{\text{reference batch}}_{\text{collected once and fixed at the start of training}}$ of examples and on x itself

Inception Score

$$\exp(\underbrace{\mathbb{E}_x}_{\text{generated images}} KL(\underbrace{p(y|x)}_{\text{inception model}}\|p(y)))$$

$p(y) = \int p(y|x)p_g(x)dx$ should have high entropy (generate varied images)

$p(y|x)$ should have low entropy (generated images should contain meaningful objects)

Semi-supervised learning

$x \mapsto (l_1, \ldots, l_K) \rightarrow$ logits

$$\begin{aligned}
L &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)}[\log p_{\text{model}}(y|\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim G}[\log p_{\text{model}}(y = K+1|\boldsymbol{x})] \\
&= L_{\text{supervised}} + L_{\text{unsupervised}}, \text{ where} \\
L_{\text{supervised}} &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)} \log p_{\text{model}}(y|\boldsymbol{x}, y < K+1) \\
L_{\text{unsupervised}} &= -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log[\underbrace{1 - p_{\text{model}}(y = K+1|\boldsymbol{x})}_{D(x)}] + \mathbb{E}_{\boldsymbol{x}\sim G} \log[\underbrace{p_{\text{model}}(y = K+1|\boldsymbol{x})}_{1-D(x)}]\},
\end{aligned}$$

$D(\boldsymbol{x}) = \frac{Z(\boldsymbol{x})}{Z(\boldsymbol{x})+1}$, where $Z(\boldsymbol{x}) = \sum_{k=1}^{K} \exp[l_k(\boldsymbol{x})]$. $\rightarrow$ by setting $l_{K+1}(x) = 0, \forall x$

Salimans, Tim, et al. "Improved techniques for training GANs." *Advances in neural information processing systems*. 2016.

InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets

Boulder

YouTube Video

Disentangled Representations
– writing styles from digit shapes on MNIST
– pose from lighting of 3D rendered images
– background digits from the central digit on SVHN
– hair styles, presence/absence of eyeglasses and emotions on CelebA

Regular GAN

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim\text{noise}}[\log(1 - D(G(z)))]$$

InfoGAN

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c; G(z,c))$$

$z \rightarrow$ source of incompressible noise
$c = (c_1, \ldots, c_L) \rightarrow$ latent code (semantic features of data)

In information theory, mutual information $I(X;Y)$ between X and Y, measures the "amount of information" learned from knowledge of random variable Y about the other random variable X.

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

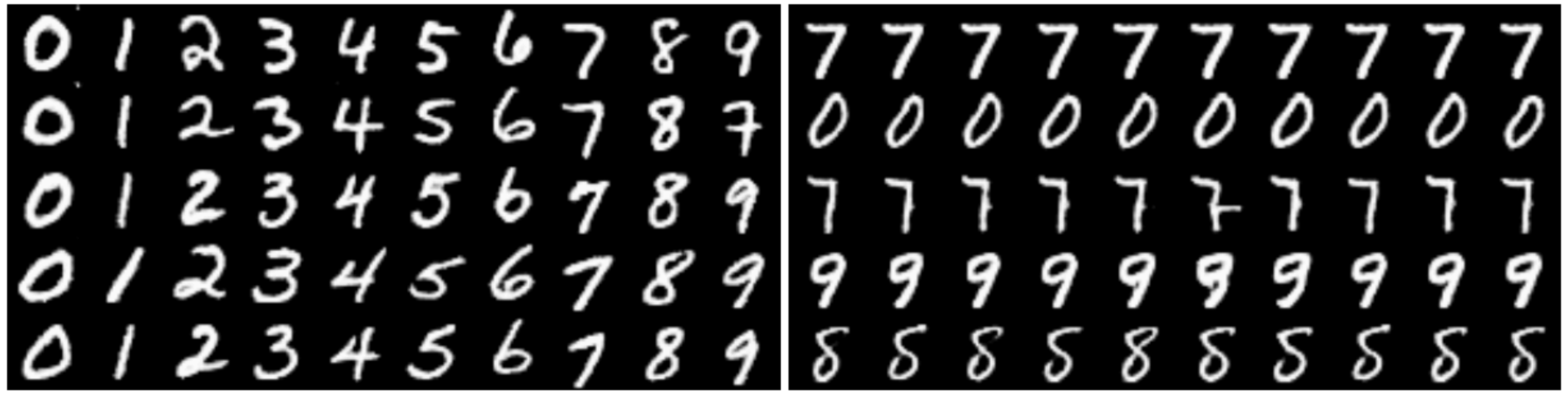
Computing $I(c; G(z,c))$ requires access to the posterior $P(c|x)$
$Q(c|x) \rightarrow$ auxiliary distribution to approximate $P(c|x)$

$$L_I(G,Q) = E_{c\sim P(c), x\sim G(z,c)}[\log Q(c|x)] + H(c) \leq I(c; G(z,c))$$

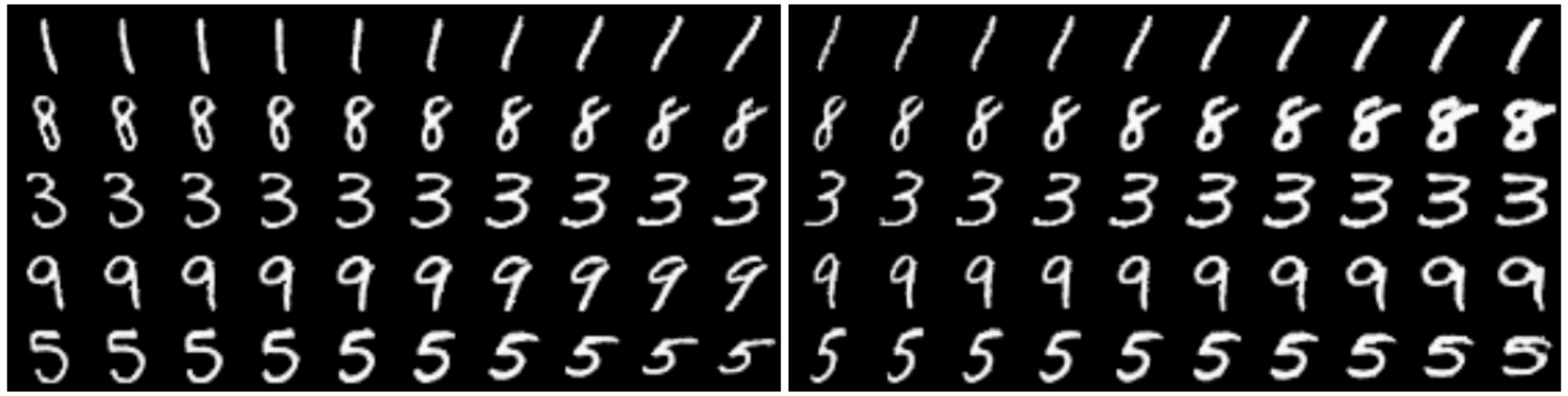
$L_I(G,Q) \rightarrow$ Variational Lower Bound

$$\min_{G,Q} \max_D V_{\text{InfoGAN}}(D,G,Q) = V(D,G) - \lambda L_I(G,Q)$$

MNIST: $c_1 \sim \text{Cat}(K=10, p=0.1)$ and $c_2, c_3 \sim \text{Unif}(-1,1)$.



(a) Varying c_1 on InfoGAN (Digit type) (b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation) (d) Varying c_3 from -2 to 2 on InfoGAN (Width)

(a) Rotation (b) Width (c) Hair style (d) Emotion

Chen, Xi, et al. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets." *arXiv preprint arXiv:1606.03657* (2016).

Least Squares Generative Adversarial Networks

[YouTube Video](#)

Generative Adversarial Networks

$$\min_G \max_D V_{\text{GAN}}(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$C(G) := \max_D V_{\text{GAN}}(D,G)$$

$$C(G) = KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}}+p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}}+p_g}{2}\right.\right) - \log(4)$$

$$C(G) = -\log 4 + 2\underbrace{JSD\left(p_{\text{data}} \| p_g\right)}_{\text{Jensen–Shannon divergence; } \geq 0;\ = 0 \text{ iff } p_{\text{data}} = p_g}$$

Least Squares Generative Adversarial Networks (LSGANs)

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x}) - b)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z})) - a)^2\right]$$

$a = -1, c = 0, b = 1$

$a = 0, b = c = 1$

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z})) - c)^2\right]$$

$a \rightarrow$ label for fake data $\quad b \rightarrow$ label for real data

$c \rightarrow$ the value that G wants D to believe for fake data

$$D^*(\boldsymbol{x}) = \frac{b p_{\text{data}}(\boldsymbol{x}) + a p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \rightarrow \text{optimal discriminator } D \text{ for a fixed } G$$

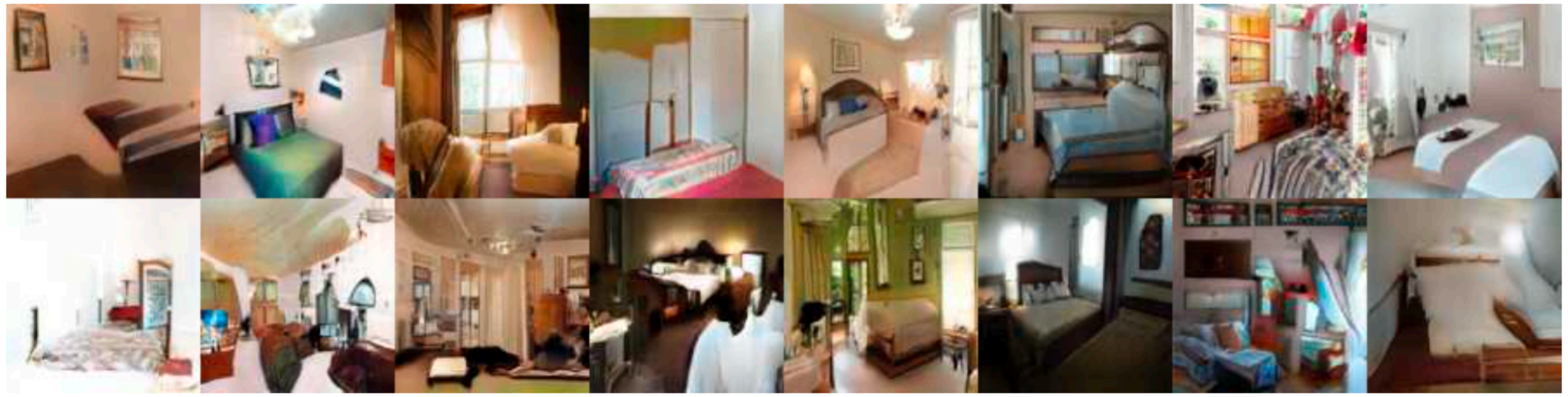
$$C(G) := V_{\text{LSGAN}}(G; D^*)$$

If we set $b - c = 1$ and $b - a = 2$, then

Pearson χ^2 divergence

$$2C(G) = \int_{\mathcal{X}} \frac{\left(2p_g(\boldsymbol{x}) - (p_{\text{d}}(\boldsymbol{x}) + p_g(\boldsymbol{x}))\right)^2}{p_{\text{d}}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \mathrm{d}x = \chi^2_{\text{Pearson}}(p_{\text{d}} + p_g \| 2p_g),$$

(a) Generated images (112 × 112) by LSGANs.

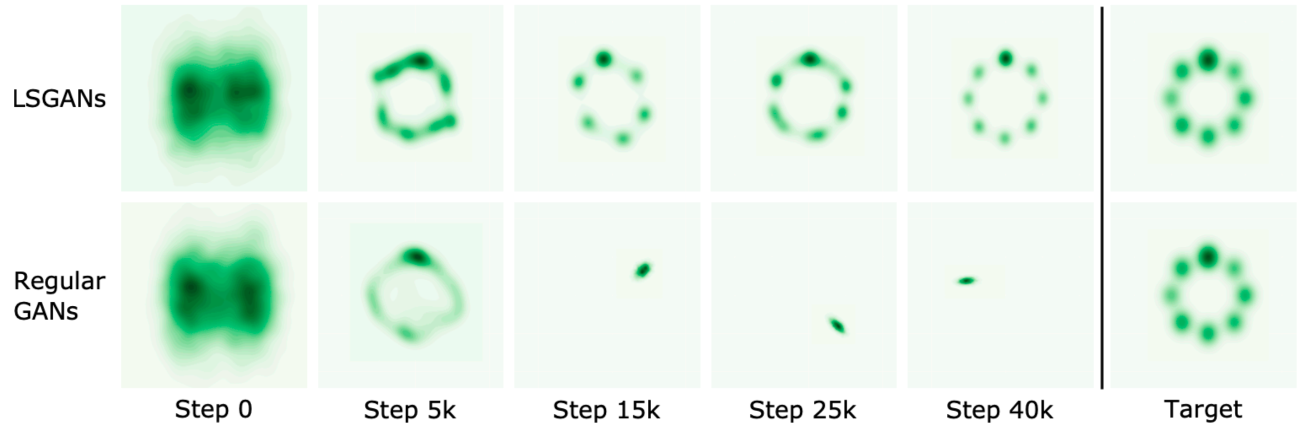


(b) Generated images (112 × 112) by DCGANs.

(b) Generated images (64 × 64) by DCGANs (reported in [25]).
Generated images on LSUN-bedroom.

Method	Inception Score
DCGAN (reported in [10])	6.16
DCGAN	6.22
LSGAN (ours)	**6.47**

Inception scores on CIFAR-10.



Mao, Xudong, et al. "Least squares generative adversarial networks." *Proceedings of the IEEE international conference on computer vision*. 2017.

Wasserstein GAN

YouTube Playlist

$(p_\theta)_{\theta \in \mathbb{R}^d} \to$ parametric family of densities

$\{x^i\}_{i=1}^m \to$ real data examples

$$\arg\max_{\theta \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^m \log p_\theta(x^i) = \arg\min_{\theta \in \mathbb{R}^d} KL(p_r \| p_\theta)$$

$p_r \to$ real data density

$Z \sim p(z)$

$g_\theta : \mathcal{Z} \to \mathcal{X}$

$\rho(p_r, p_\theta) \to$ distance or divergence

$\mathcal{X} \to$ compact metric space (e.g., space of images $[0,1]^d$)

$\sum \to$ set of all the Borel subsets of $\mathcal{X}$

$$\delta(p_r, p_g) = \sup_{A \in \sum} |p_r(A) - p_g(A)| \to \text{total variation (TV) distance}$$

$$KL(p_r \| p_g) = \int \log \left(\frac{p_r(x)}{p_g(x)} \right) p_r(x) d\mu(x) \to \text{Kullback-Leibler (KL) divergence}$$

$$JS(p_r, p_g) = KL(p_r \| \frac{p_r + p_g}{2}) + KL(p_g \| \frac{p_r + p_g}{2}) \to \text{Jensen-Shannon (JS) divergence}$$

$$W(p_r, p_g) = \inf_{\gamma \sim \Pi(p_r, p_g)} \mathbb{E}_{(x,y)}[\|x - y\|] \to \text{Earth-Mover (EM) distance or Wasserstein-1}$$

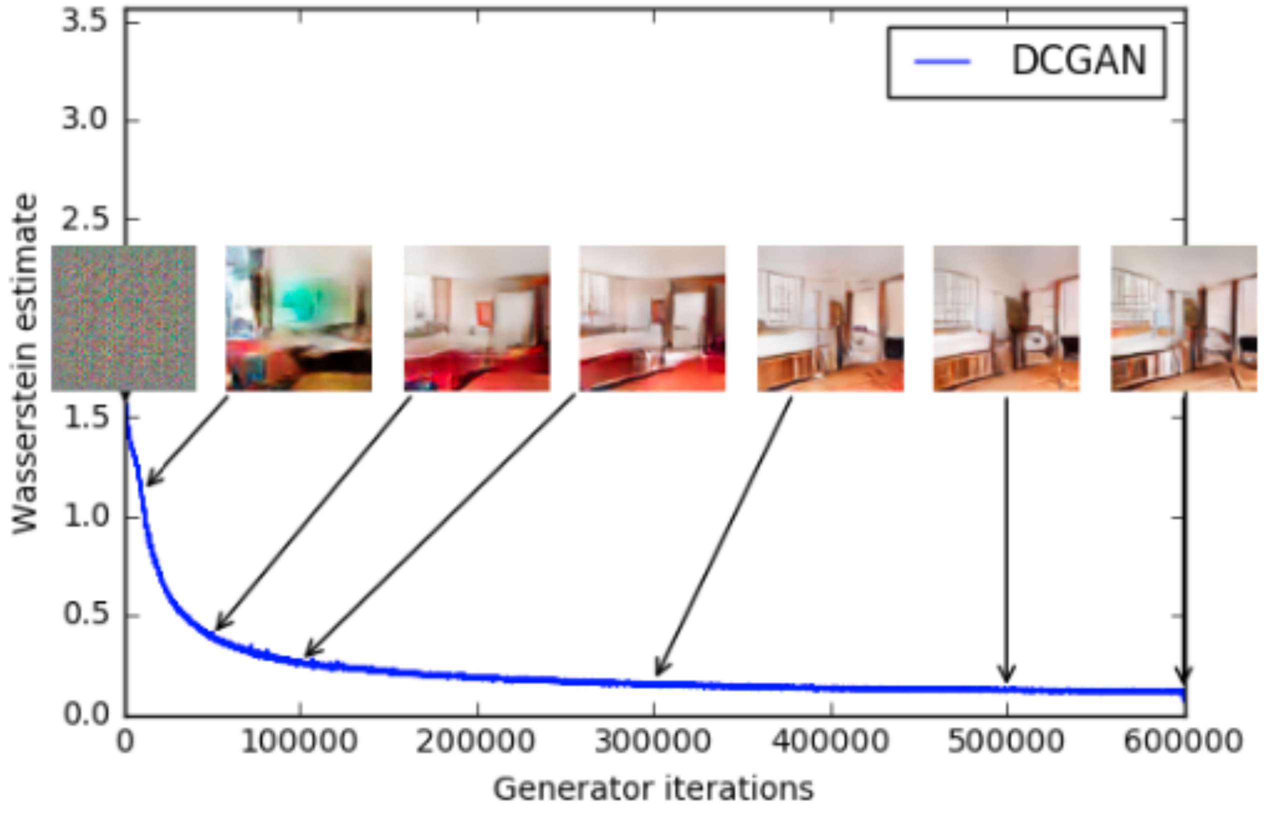
$\Pi(p_r, p_g) \to$ set of all joint distributions $\gamma(x, y)$ whose marginals are respectively p_r and p_g

Kantorovich-Rubinstein duality

$$W(p_r, p_\theta) = \sup_{\|f\|_L \le 1} \mathbb{E}_{x \sim p_r}[f(x)] - \mathbb{E}_{x \sim p_\theta}[f(x)]$$

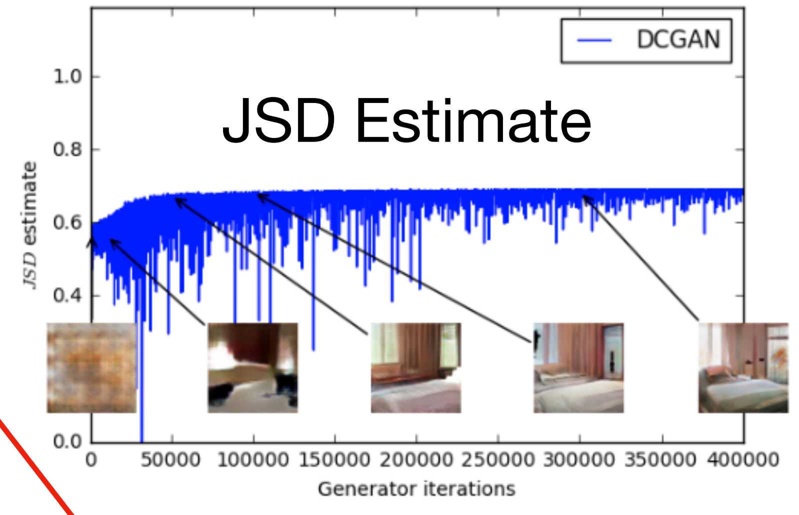
$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim p_r}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z))]$$

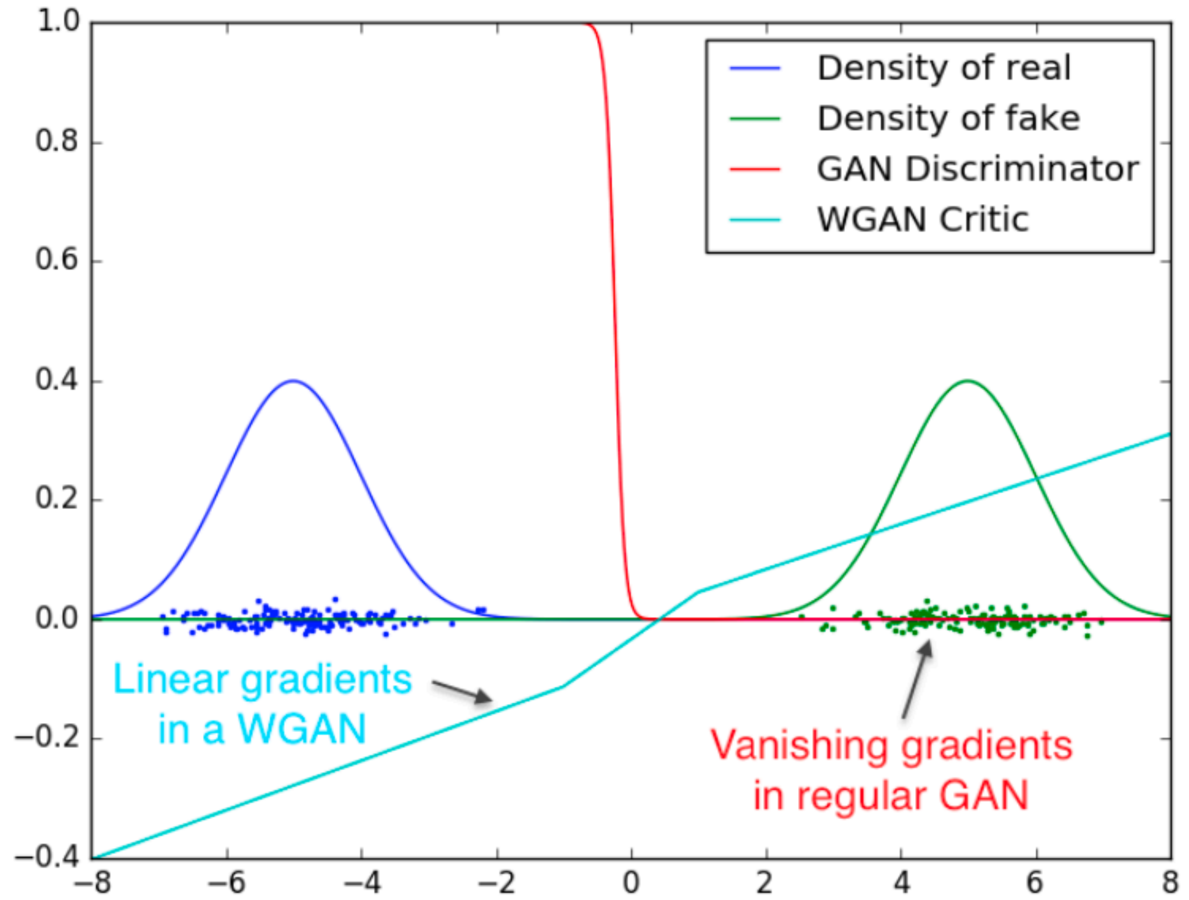
clamp the weights to a fixed box



Integral Probability Metrics (IPMs)

$$d_{\mathcal{F}}(p_r, p_\theta) = \sup_{f \in \mathcal{F}} \mathbb{E}_{x \sim p_r}[f(x)] - \mathbb{E}_{x \sim p_\theta}[f(x)]$$





Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." *arXiv preprint arXiv:1701.07875* (2017).

Boulder

Improved Training of Wasserstein GANs

YouTube Playlist

GANs

$$\min_G \max_D \mathbb{E}_{x\sim p_r}[\log D(x)] + \mathbb{E}_{\tilde{x}\sim p_g}[\log(1 - D(\tilde{x}))]$$

$\tilde{x} = G(z), z \sim p(z)$

Jenson-Shannon Divergence between p_r & p_g

$$\max_G \mathbb{E}_{\tilde{x}\sim p_g}[\log D(\tilde{x})]$$

Wasserstein GANs

Earth-mover (Wasserstein-1) distance $W(q, p)$

minimum cost of transporting mass in order to transform the distribution q into the distribution p

cost is mass times transport distance

$$\min_G \underbrace{\max_{D\in\mathcal{D}}}_{\text{critic}} \mathbb{E}_{x\sim p_r}[D(x)] - \mathbb{E}_{\tilde{x}\sim p_g}[D(\tilde{x})]$$

$\mathcal{D} \rightarrow$ set of 1-Lipschitz functions

clip the weights of the critic

Gradient penalty

A differentiable function is 1-Lipschtiz iff it has gradients with norm at most 1 everywhere

$$L = \underbrace{\mathbb{E}_{\tilde{x}\sim p_g}[D(\tilde{x})] - \mathbb{E}_{x\sim p_r}[D(x)]}_{\text{Original critic loss}} + \lambda \underbrace{\mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]}_{\text{Gradient penalty}}$$

$\hat{x} \rightarrow$ sample uniformly along straight lines between x & $\tilde{x}$

Modeling discrete data with a continuous generator

$$\underbrace{G : z \mapsto \{v_1, \ldots, v_{32}\}}_{\text{1D CNN}}$$

$v_i \in \mathbb{R}^n, n \rightarrow$ vocabulary size
one-hot character vectors

softmax output is passed directly into the critic (i.e., no sampling step)

$$\underbrace{D : \{v_1, \ldots, v_{32}\} \mapsto d \in \mathbb{R}}_{\text{1D CNN}}$$

decoding samples: take the arg max of each output vector

WGAN with gradient penalty (1D CNN)	
Busino game camperate spent odea	Solice Norkedin pring in since
In the bankaway of smarling the	ThiS record (31.) UBS) and Ch
SingersMay , who kill that imvic	It was not the annuas were plogr
Keray Pents of the same Reagun D	This will be us , the ect of DAN
Manging include a tudancs shat "	These leaded as most-worsd p2 a0
His Zuith Dudget , the Denmbern	The time I paidOa South Cubry i
In during the Uitational questio	Dour Fraps higs it was these del
Divos from The ’ noth ronkies of	This year out howneed allowed lo
She like Monday , of macunsuer S	Kaulna Seto consficutes to repor

$$\Delta_n = \{v = (v^1, \ldots, v^n) \in \mathbb{R}^n : v^j \geq 0, \sum_j v^j = 1\}$$

$$V_n = \{v = (v^1, \ldots, v^n) \in \mathbb{R}^n : v^j \in \{0, 1\}, \sum_j v^j = 1\} \subset \Delta_n$$

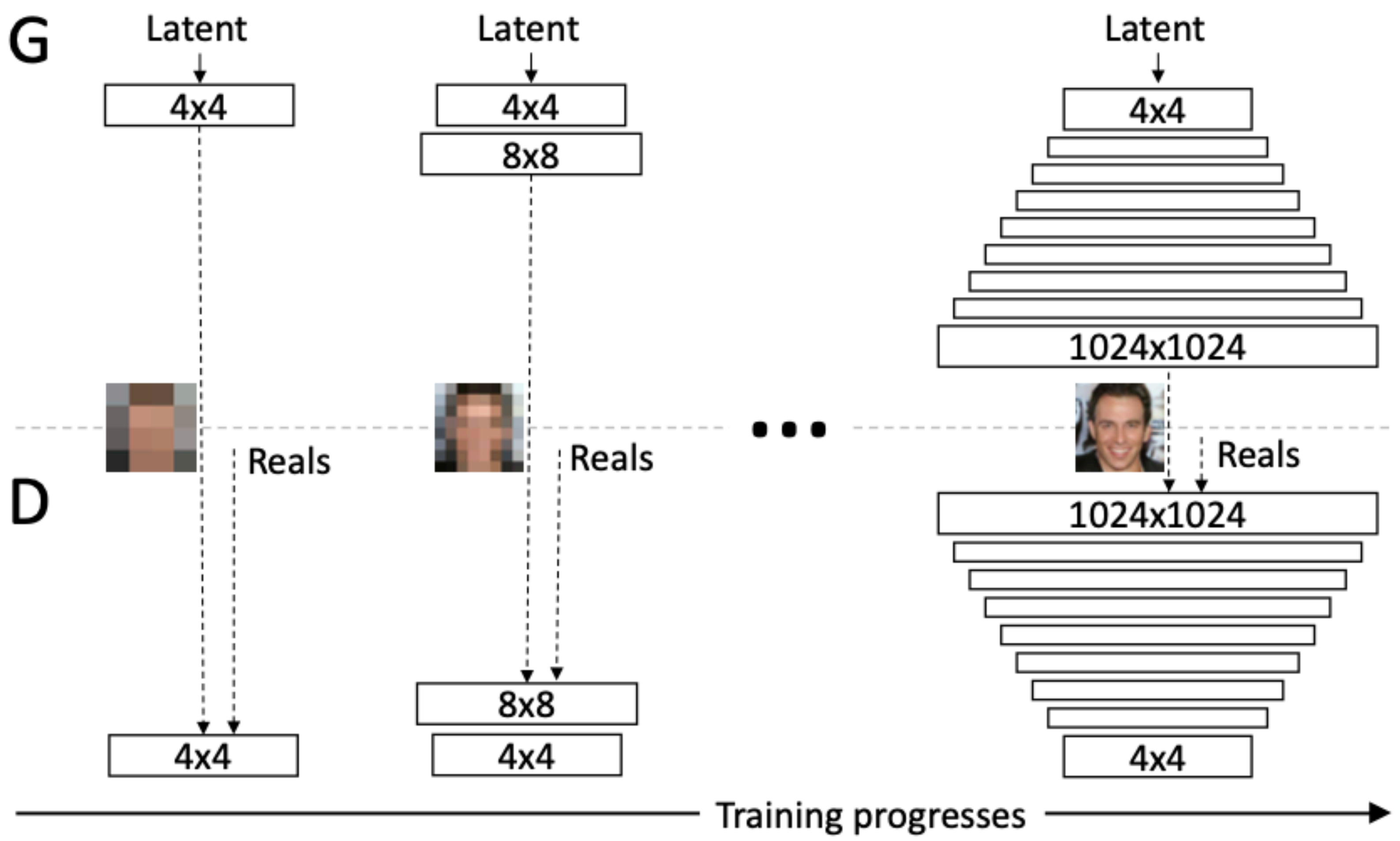
$JS(p_r, p_g)$ saturates on $\Delta_n^{32} \supset V_n^{32}$ while $W(p_r, p_g)$ is well-defined

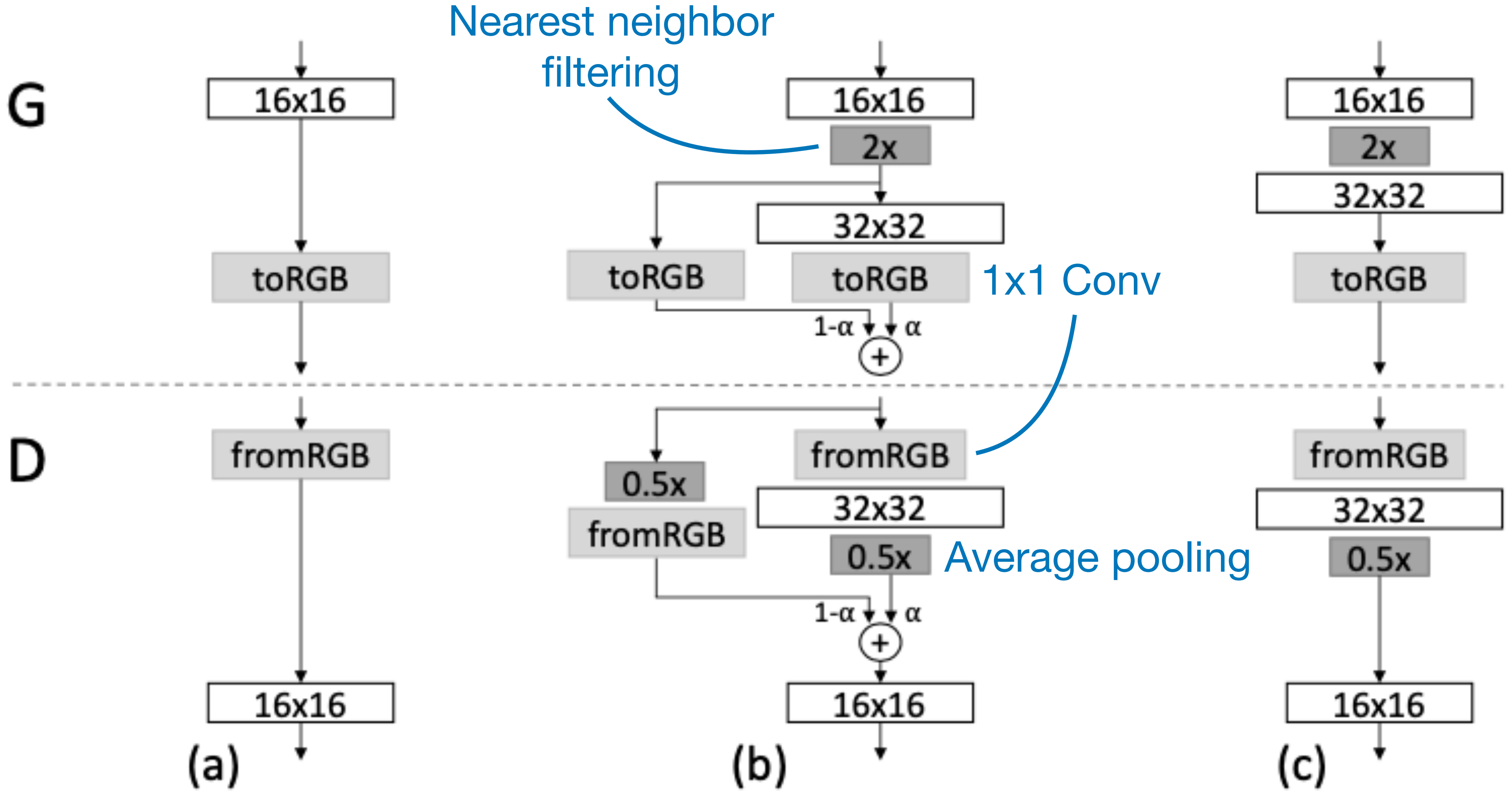
Gulrajani, Ishaan, et al. "Improved training of Wasserstein GANs." *Advances in neural information processing systems*. 2017.

Progressive growing of GANs for improved quality, stability, and variation

Boulder

YouTube Video





Increasing Variation Using Minibatch Standard Deviation

$X \in \mathbb{R}^{N\times 512\times 4\times 4} \rightarrow$ feature maps

$$M(f,x,y) = \frac{1}{N}\sum_{i=1}^{N} X(i,f,x,y) \rightarrow \text{mean}$$

$$S(f,x,y) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[X(i,f,x,y) - M(f,x,y)]^2} \rightarrow \text{standard deviation}$$

$$z = \frac{1}{512\times 4\times 4}\sum_{f,x,y} S(f,x,y) \implies Y(i,f,x,y) := \begin{cases} X(i,f,x,y) & \text{if } f \le 512; \\ z & \text{if } f = 513. \end{cases}$$

Equalized Learning Rate

$w_i \sim \mathcal{N}(0,1)$

$\hat{w}_i = w_i / c$

$c = \sqrt{2/n} \rightarrow$ per layer normalization constant

$n = k^2 d,\ k \rightarrow$ filter size, $d \rightarrow$ channels

Pixelwise Feature Vector Normalization In Generator

$$b_{x,y} = a_{x,y} / \sqrt{\frac{1}{N}\sum_{j=0}^{N-1}(a^j_{x,y})^2 + \epsilon}, \text{ where } \epsilon = 10^{-8}$$

a variant of "local response normalization"

Mao et al. (2016b) (128 × 128) Gulrajani et al. (2017) (128 × 128) (256 × 256)

Karras, Tero, et al. "Progressive growing of GANs for improved quality, stability, and variation." *arXiv preprint arXiv:1710.10196* (2017).

GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium

Boulder

YouTube Video

TTUR: Two Time-scale Update Rule

$D(., w) \rightarrow$ discriminator with parameter vector w

$G(., \theta) \rightarrow$ generator with parameter vector θ

$\tilde{g}(\theta, w) \rightarrow$ stochastic gradient of the discriminator's loss $\mathcal{L}_D$

$\tilde{h}(\theta, w) \rightarrow$ stochastic gradient of the generator's loss $\mathcal{L}_G$

mini-batches of m real world samples x^i
and m synthetic samples z^i

$$\begin{cases} g(\theta, w) = \nabla_w \mathcal{L}_D \\ h(\theta, w) = \nabla_\theta \mathcal{L}_G \end{cases} \rightarrow \text{true gradients}$$

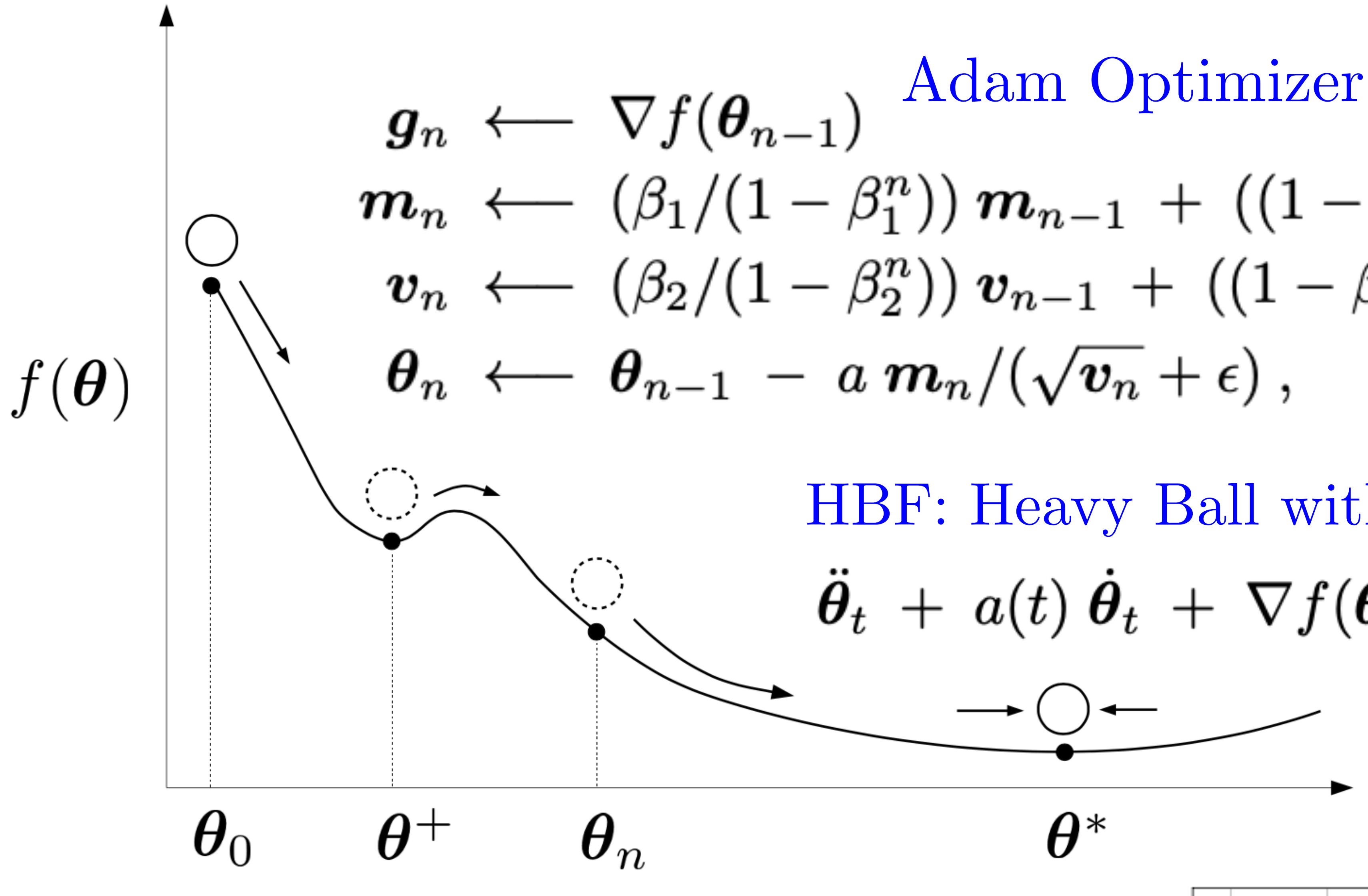
$$\begin{cases} \tilde{g}(\theta, w) = g(\theta, w) + M_w \\ \tilde{h}(\theta, w) = h(\theta, w) + M_\theta \end{cases} \rightarrow \text{random variables } M_w \ \& \ M_\theta$$

$b(n) \rightarrow$ learning rate for discriminator update (fast)

$a(n) \rightarrow$ learning rate for generator update (slow)

The proof relies on the fact that there eventually is a time point when the perturbation of the slow update rule is small enough to allow the fast update rule to converge

Apply the idea of perturbed ODEs ($\dot{\theta} = \nabla f(\theta)$)



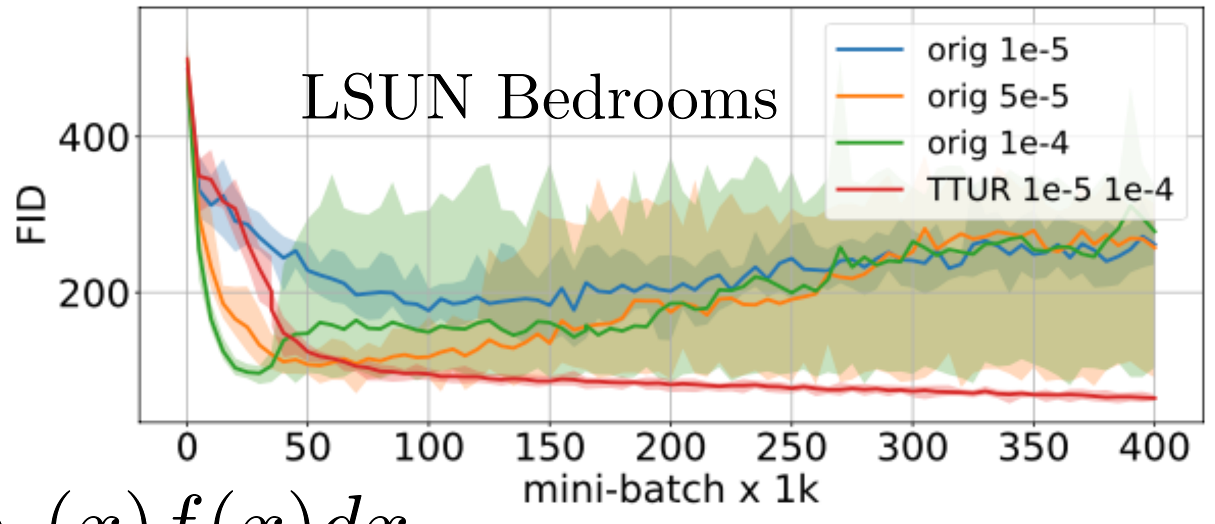
Adam Optimizer

$$\begin{aligned} \boldsymbol{g}_n &\longleftarrow \nabla f(\boldsymbol{\theta}_{n-1}) \\ \boldsymbol{m}_n &\longleftarrow (\beta_1/(1-\beta_1^n))\, \boldsymbol{m}_{n-1} + ((1-\beta_1)/(1-\beta_1^n))\, \boldsymbol{g}_n \\ \boldsymbol{v}_n &\longleftarrow (\beta_2/(1-\beta_2^n))\, \boldsymbol{v}_{n-1} + ((1-\beta_2)/(1-\beta_2^n))\, \boldsymbol{g}_n \odot \boldsymbol{g}_n \\ \boldsymbol{\theta}_n &\longleftarrow \boldsymbol{\theta}_{n-1} - a\, \boldsymbol{m}_n/(\sqrt{\boldsymbol{v}_n} + \epsilon)\,, \end{aligned}$$

HBF: Heavy Ball with Friction

$$\ddot{\boldsymbol{\theta}}_t + a(t)\, \dot{\boldsymbol{\theta}}_t + \nabla f(\boldsymbol{\theta}_t) = \boldsymbol{0}$$

Fréchet Inception Distance



$p_r(.) \rightarrow$ real world data

$p_g(.) \rightarrow$ generating model data

$p_r(.) = p_g(.)$ iff $\int p_r(x) f(x) dx = \int p_g(x) f(x) dx$

$f(x) \rightarrow$ basis spanning the function space in which $p_r(.)$ & $p_g(.)$ live

$f(x) \rightarrow$ polynomials, first & second moments, Gaussian

$x \rightarrow$ coding layer of an Inception model

$$d^2((m_g, C_g), (m_r, C_r)) = \|m_g - m_r\|_2^2 + \mathrm{Tr}(C_g + C_r - 2(C_g C_r)^{1/2})$$

Heusel, Martin, et al. "GANs trained by a two time-scale update rule converge to a local Nash equilibrium." *Advances in neural information processing systems*. 2017.

Spectral Normalization for Generative Adversarial Networks

Boulder

YouTube Video

$$\min_G \max_D V(G,D)$$

$$V(G,D) = \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{x'\sim p_g}[\log(1-D(x'))]$$

$$D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x)+p_g(x)}$$

$$D_G^*(x) = \text{sigmoid}(f^*(x))$$

$$f^*(x) = \log p_{\text{data}}(x) - \log p_g(x)$$

$$\nabla_x f^*(x) = \frac{1}{p_{\text{data}}(x)}\nabla_x p_{\text{data}}(x) - \frac{1}{p_g(x)}\nabla_x p_g(x)$$

can be unbounded or even incomputable

$$\arg\max_{\|f\|_{\text{Lip}}\le K} V(G,D),$$

Spectral Normalization

$g : \boldsymbol{h}_{in} \mapsto \boldsymbol{h}_{out}$ $\rightarrow$ each layer

$$\|g\|_{\text{Lip}} = \sup_h \underbrace{\sigma(\nabla g(h))}_{\text{spectral norm}}$$

$$\sigma(A) := \max_{\boldsymbol{h}:\boldsymbol{h}\neq \boldsymbol{0}} \frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2} = \max_{\|\boldsymbol{h}\|_2\le 1}\|A\boldsymbol{h}\|_2,$$

$$f(\boldsymbol{x},\theta) = W^{L+1}a_L(W^L(a_{L-1}(W^{L-1}(\dots a_1(W^1\boldsymbol{x})\dots)))),$$

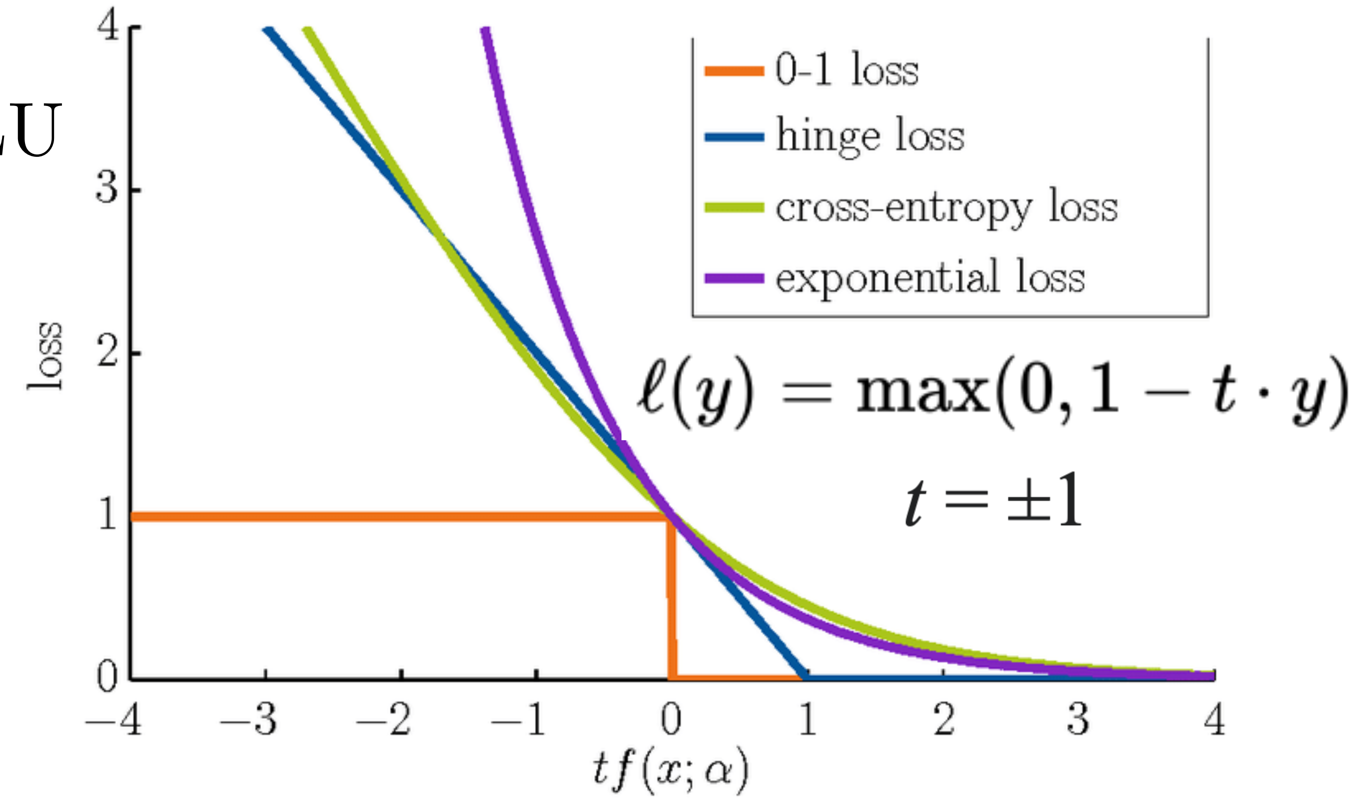
$$\|f\|_{\text{Lip}} \le \|(\boldsymbol{h}_L \mapsto W^{L+1}\boldsymbol{h}_L)\|_{\text{Lip}}\cdot\|a_L\|_{\text{Lip}}\cdot\|(\boldsymbol{h}_{L-1}\mapsto W^L\boldsymbol{h}_{L-1})\|_{\text{Lip}}$$

$$\cdots\|a_1\|_{\text{Lip}}\cdot\|(\boldsymbol{h}_0\mapsto W^1\boldsymbol{h}_0)\|_{\text{Lip}} = \prod_{l=1}^{L+1}\|(\boldsymbol{h}_{l-1}\mapsto W^l\boldsymbol{h}_{l-1})\|_{\text{Lip}} = \prod_{l=1}^{L+1}\sigma(W^l).$$

$$\|g_1\circ g_2\|_{\text{Lip}} \le \|g_1\|_{\text{Lip}}\cdot\|g_2\|_{\text{Lip}}$$

$\|a_l\|_{\text{Lip}}$ is equal to 1 $\rightarrow$ ReLU & Leaky ReLU

$$\bar{W}_{\text{SN}}(W) := W/\sigma(W).$$



Power Iteration Method

$$\tilde{\boldsymbol{v}} \leftarrow W^{\text{T}}\tilde{\boldsymbol{u}}/\|W^{\text{T}}\tilde{\boldsymbol{u}}\|_2,\ \tilde{\boldsymbol{u}} \leftarrow W\tilde{\boldsymbol{v}}/\|W\tilde{\boldsymbol{v}}\|_2.$$

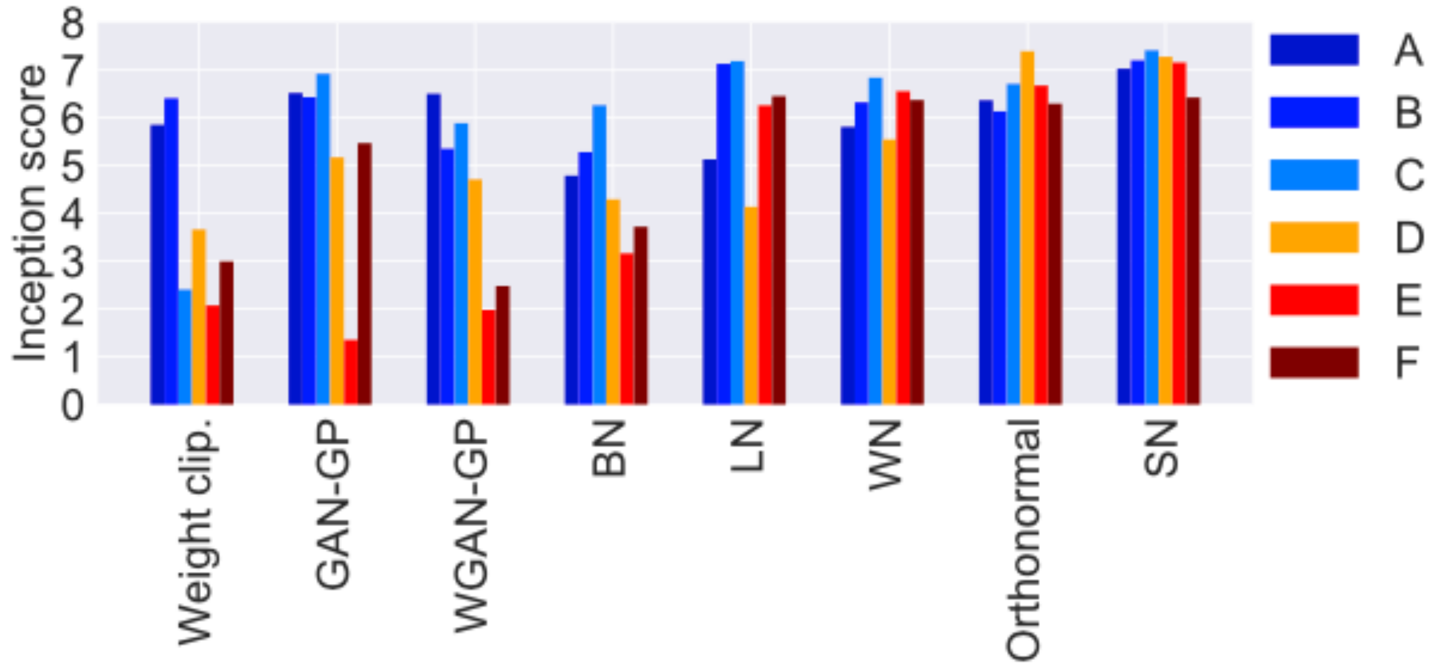
$$\sigma(W) \approx \tilde{\boldsymbol{u}}^{\text{T}}W\tilde{\boldsymbol{v}}.$$

Hinge Loss

$$V_D(\hat{G},D) = \mathop{\text{E}}_{\boldsymbol{x}\sim q_{\text{data}}(\boldsymbol{x})}[\min(0,-1+D(\boldsymbol{x}))] + \mathop{\text{E}}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\min\left(0,-1-D\left(\hat{G}(\boldsymbol{z})\right)\right)\right]$$

$$V_G(G,\hat{D}) = -\mathop{\text{E}}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\hat{D}\left(G(\boldsymbol{z})\right)\right],$$

Setting	α	β_1	β_2	n_{dis}
A†	0.0001	0.5	0.9	5
B‡	0.0001	0.5	0.999	1
C*	0.0002	0.5	0.999	1
D	0.001	0.5	0.9	5
E	0.001	0.5	0.999	5
F	0.001	0.9	0.999	5



Miyato, Takeru, et al. "Spectral normalization for generative adversarial networks." *arXiv preprint arXiv:1802.05957* (2018).

Large Scale GAN Training for High Fidelity Natural Image Synthesis

Boulder

YouTube Video

Background

$G \rightarrow$ generator network

$D \rightarrow$ discriminator network

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))]$$

$z \in \mathbb{R}^{d_z} \rightarrow$ latent variable

$z \sim p(z) = \mathcal{N}(0, I)$ or $\mathcal{U}[-1, 1]$

BigGANs

– Larger Models

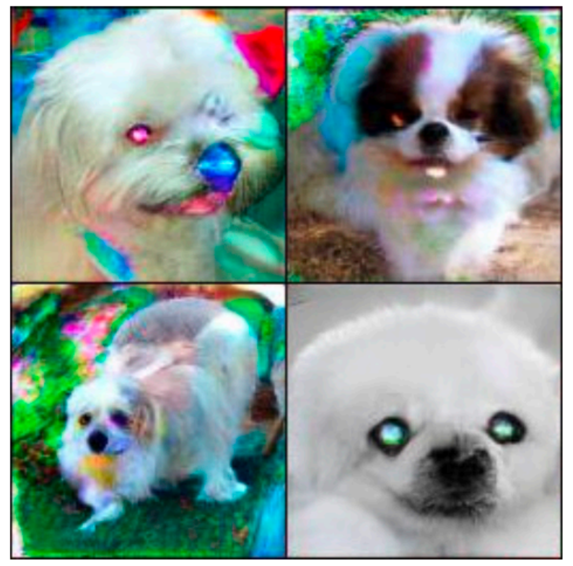
– Larger Batches

1) SA-GAN (Self-Attention Generative Adversarial Networks) architecture
2) Hinge Loss
3) Provide class information to G with class-conditional BatchNorm
4) Provide class information to to D with projection
5) Spectral Normalization in G
6) For evaluation, employ moving averages of G's weights
7) Orthogonal Initialization

Truncation Trick

1) Train using $z \sim \mathcal{N}(0, I)$
2) Sample using truncated normal distribution

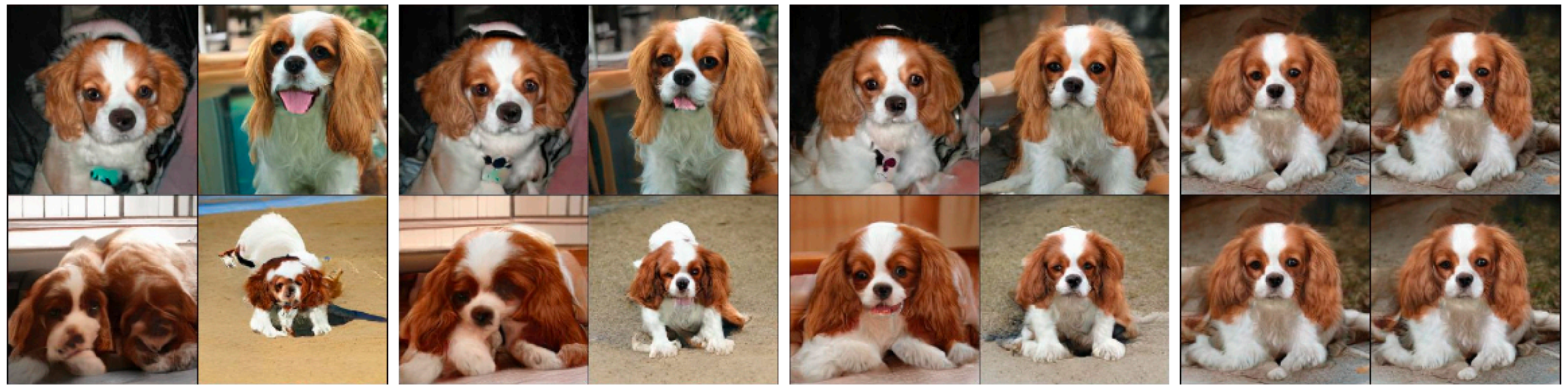
Improvement in individual sample quality at the cost of reduction in overall sample variety!



Saturation artifacts from applying truncation to a poorly conditioned model!

Orthogonal Regularization

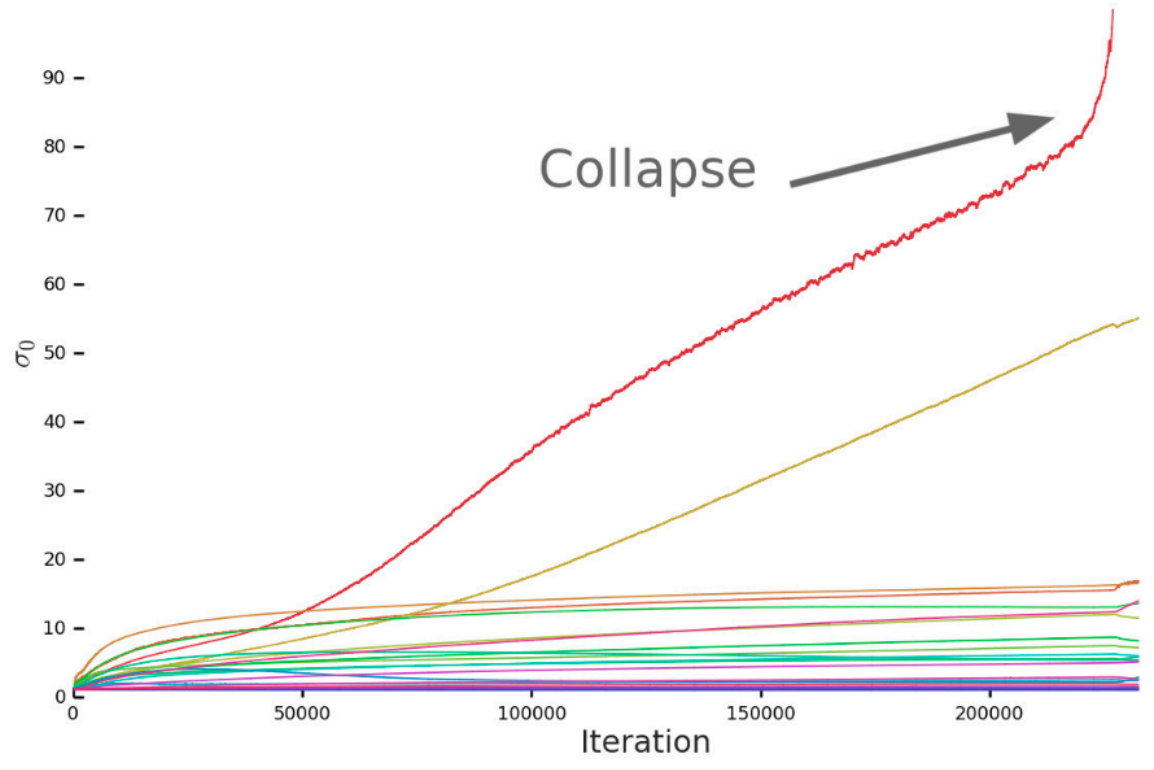
$$R_\beta(W) = \beta \|W^T W \odot (1 - I)\|_F^2 \qquad \beta = 10^{-4}$$



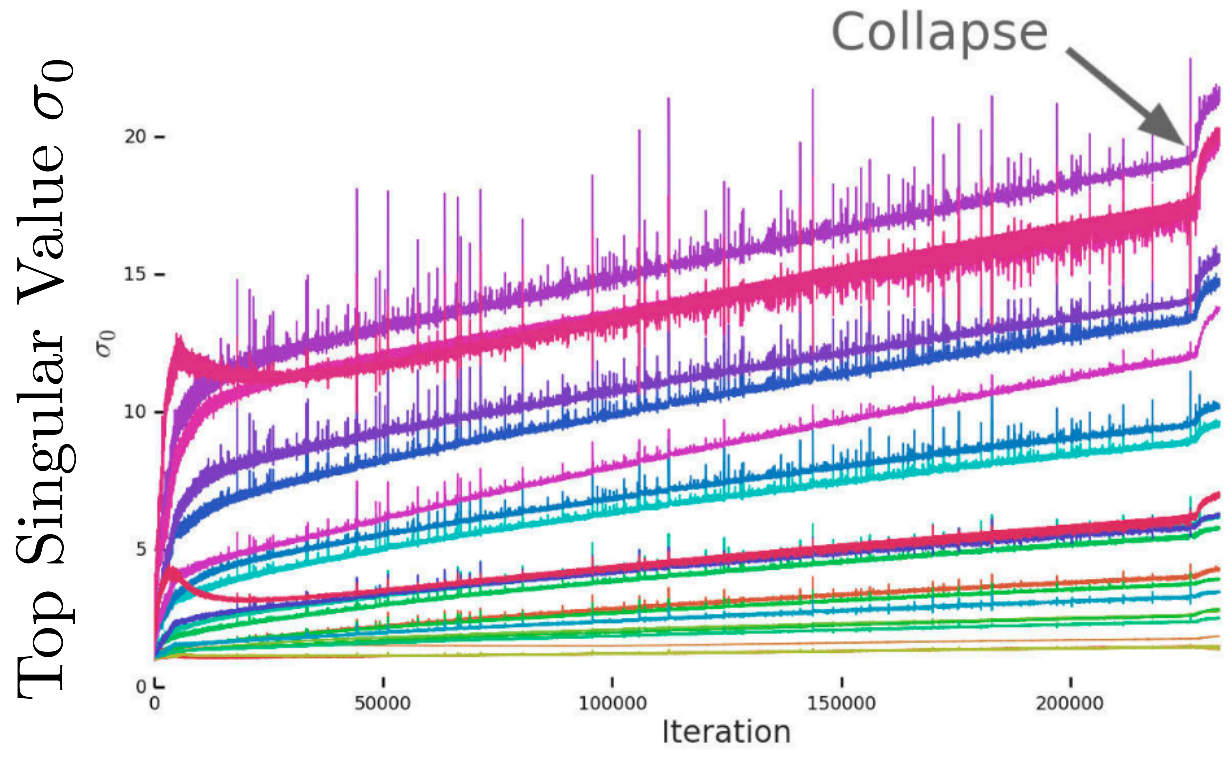
The effects of increasing truncation. From left to right, the threshold is set to 2, 1, 0.5, 0.04.

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)

Fréchet Inception Distance (FID, lower is better) and Inception Score (IS, higher is better)



(a) G



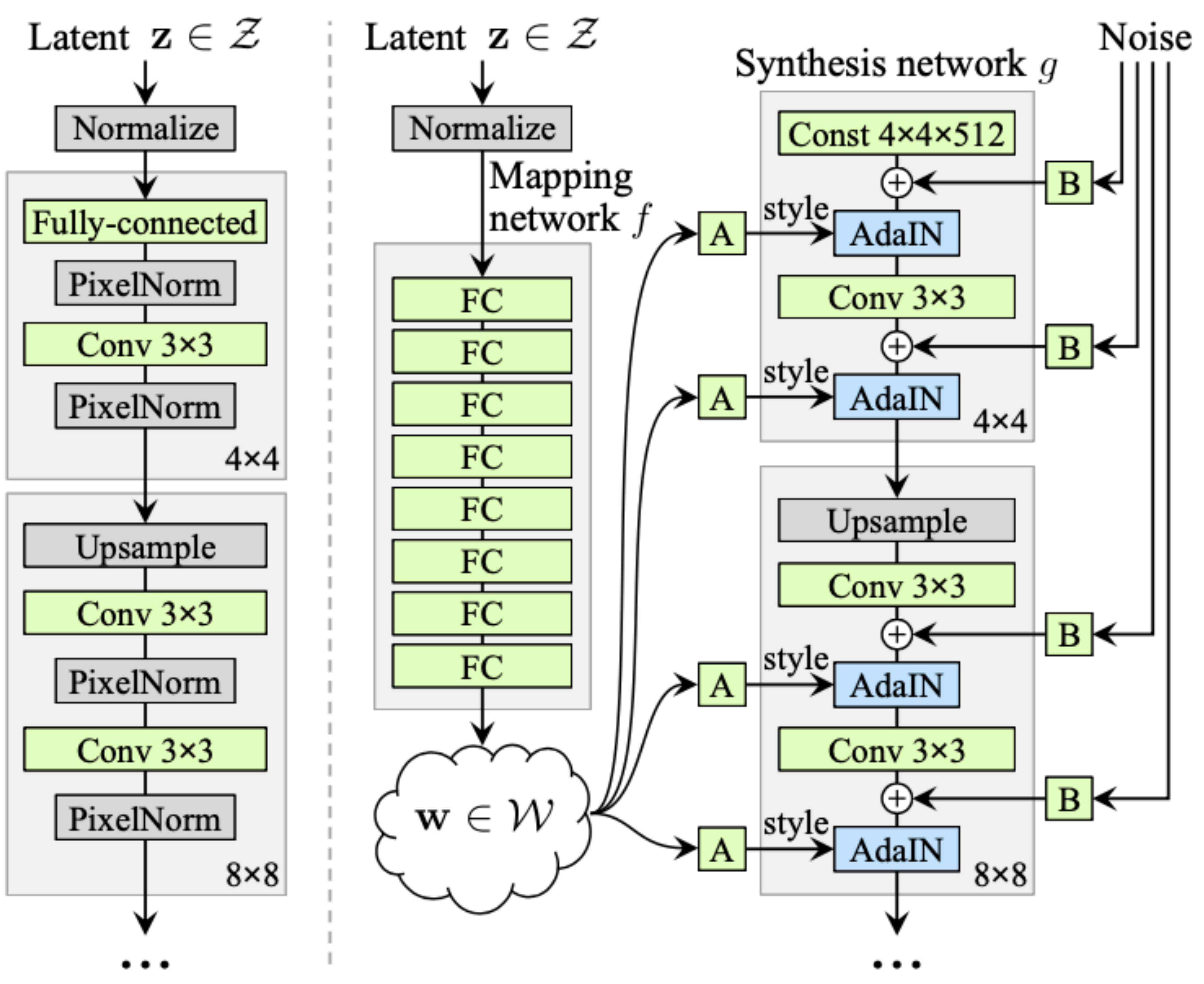
(b) D

Brock, Andrew, Jeff Donahue, and Karen Simonyan. "Large scale gan training for high fidelity natural image synthesis." *arXiv preprint arXiv:1809.11096* (2018).

A Style-Based Generator Architecture for Generative Adversarial Networks

Boulder

[YouTube Video](#)



(a) Traditional (b) Style-based generator

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

$X \in \mathbb{R}^{C \times H \times W}$ adaptive instance normalization

$\mathbf{x}_i = X(i) \in \mathbb{R}^{H \times W}$

$$\mu(\mathbf{x}_i) = \frac{1}{HW} \sum_{h,w} X(i,h,w) \in \mathbb{R}$$

$$\sigma(\mathbf{x}_i) = \sqrt{\frac{1}{HW} \sum_{h,w} (X(i,h,w) - \mu(\mathbf{x}_i))^2} \in \mathbb{R}$$

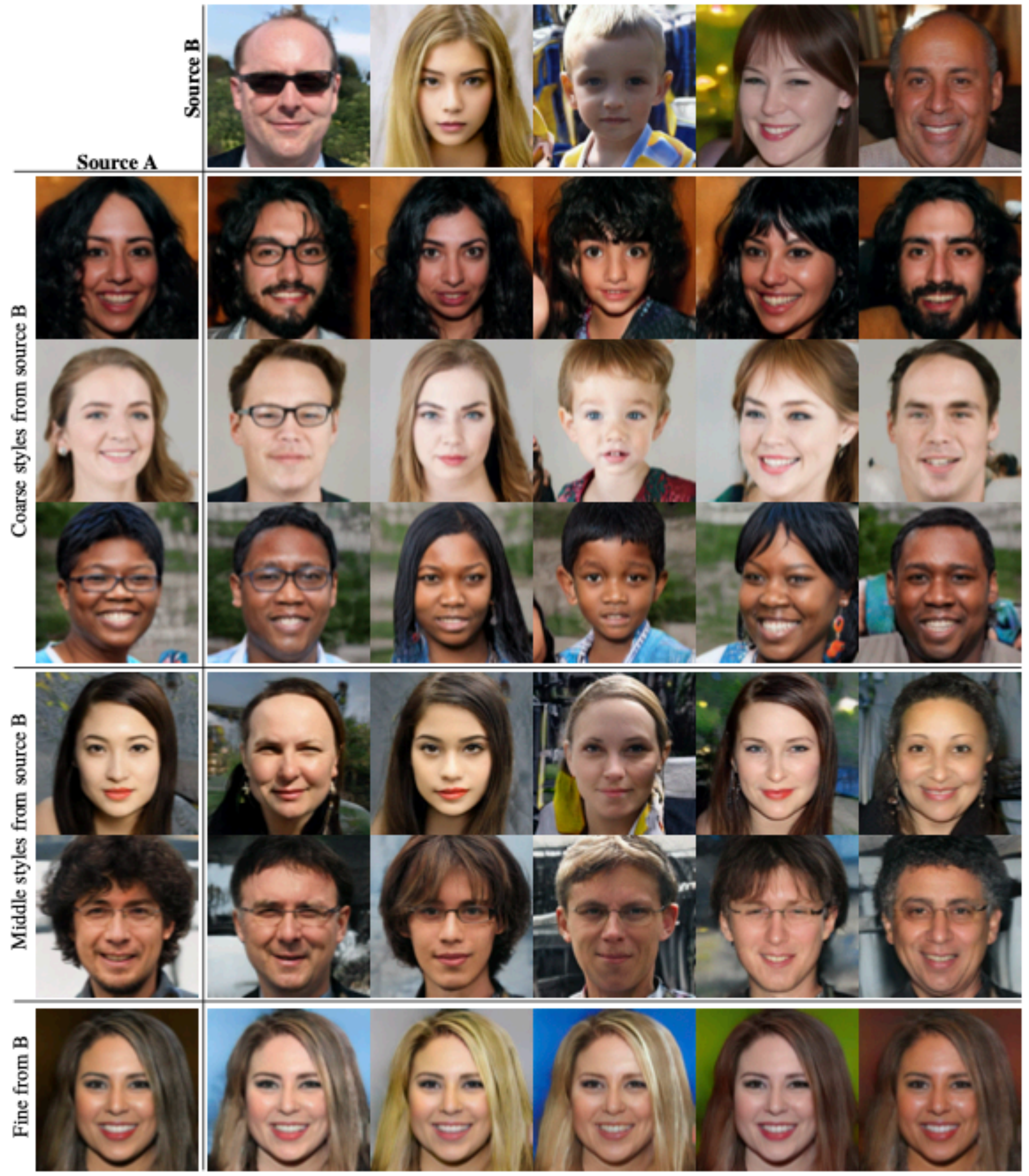
$$\mathbf{w} \xmapsto[\text{affine}]{A} (\mathbf{y}_s, \mathbf{y}_b) \in \mathbb{R}^{2C}$$

$Z \in \mathbb{R}^{H \times W} \to$ Gaussian Noise

$Y(i,h,w) = b_i Z(h,w)$

learned per-feature scaling factors

This generator admits a more linear, less entangled representation of different factors of variation



mixing regularization

$z_1 \mapsto w_1$

$z_2 \mapsto w_2$

at a randomly selected point in the synthesis network switch from w_1 to w_2

Truncation trick

$\bar{\mathbf{w}} = \mathbb{E}_{\mathbf{z} \sim P(\mathbf{z})}[f(\mathbf{z})]$

$\mathbf{w}' = \bar{\mathbf{w}} + \psi(\mathbf{w} - \bar{\mathbf{w}}) \quad \psi < 1$

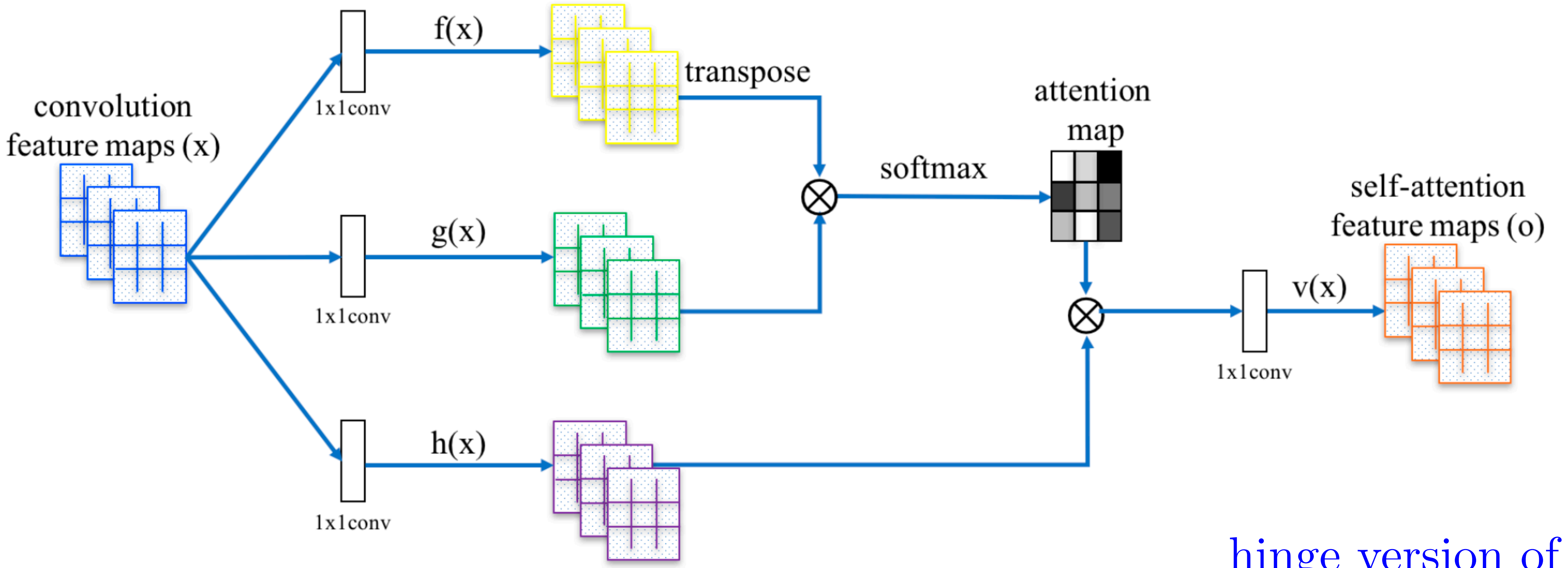
(a) Generated image (b) Stochastic variation (c) Standard deviation

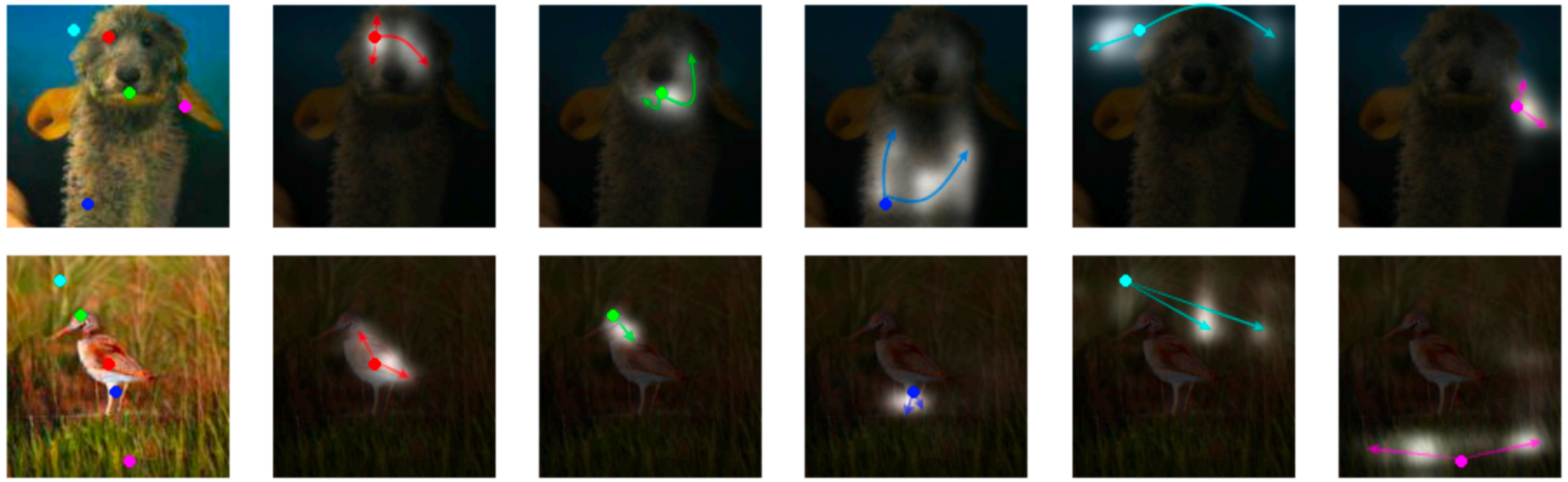
Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019.

Boulder

Self-Attention Generative Adversarial Networks

YouTube Video





$x \in \mathbb{R}^{C \times N}, C \rightarrow \#$ channel, $N \rightarrow \#$ feature locations

$f = f(x) = W_f x \in \mathbb{R}^{\overline{C}}, W_f \in \mathbb{R}^{\overline{C} \times C}$

$g = g(x) = W_g x \in \mathbb{R}^{\overline{C}}, W_g \in \mathbb{R}^{\overline{C} \times C}$ $\qquad \overline{C} = C/8$

$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N} \exp(s_{ij})} \qquad s_{i,j} = f(x_i)^T g(x_j)$$

the extent to which the model attends to the i-th location when synthesizing the j-th region

$o = (o_1, \ldots, o_j, \ldots, o_N) \in \mathbb{R}^{C \times N}$

$$o_j = v\left(\sum_{i=1}^{N} \beta_{j,i} h(x_i)\right), W_h \in \mathbb{R}^{\overline{C} \times C}, h(x) = W_h x$$

$v(x) = W_v x, W_v \in \mathbb{R}^{C \times \overline{C}}$ $\qquad y_i = \gamma o_i + x_i$

hinge version of the adversarial loss

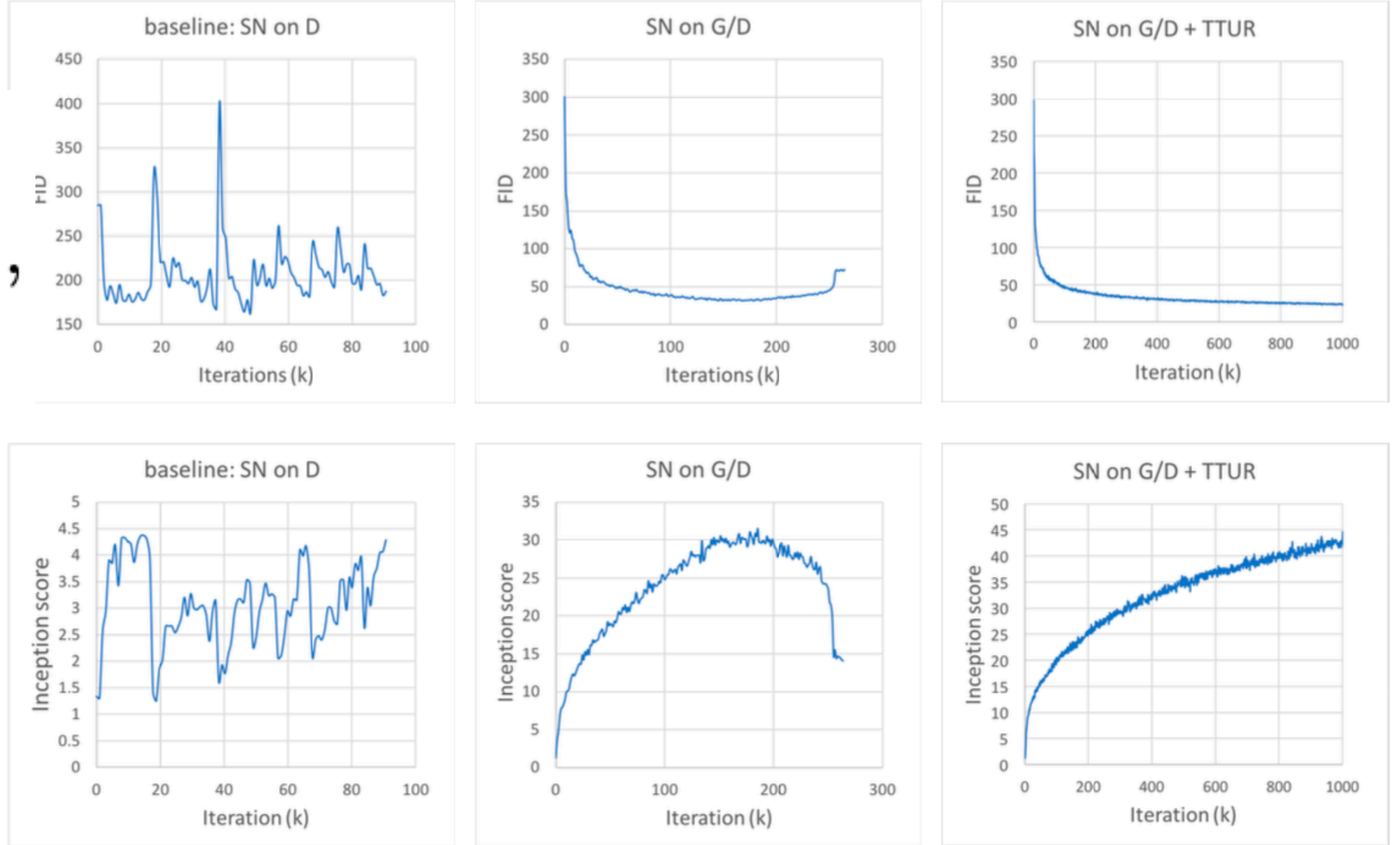
$$\begin{aligned} L_D = & -\mathbb{E}_{(x,y)\sim p_{data}}[\min(0, -1 + D(x,y))] \\ & -\mathbb{E}_{z\sim p_z, y\sim p_{data}}[\min(0, -1 - D(G(z),y))], \\ L_G = & -\mathbb{E}_{z\sim p_z, y\sim p_{data}} D(G(z),y), \end{aligned}$$

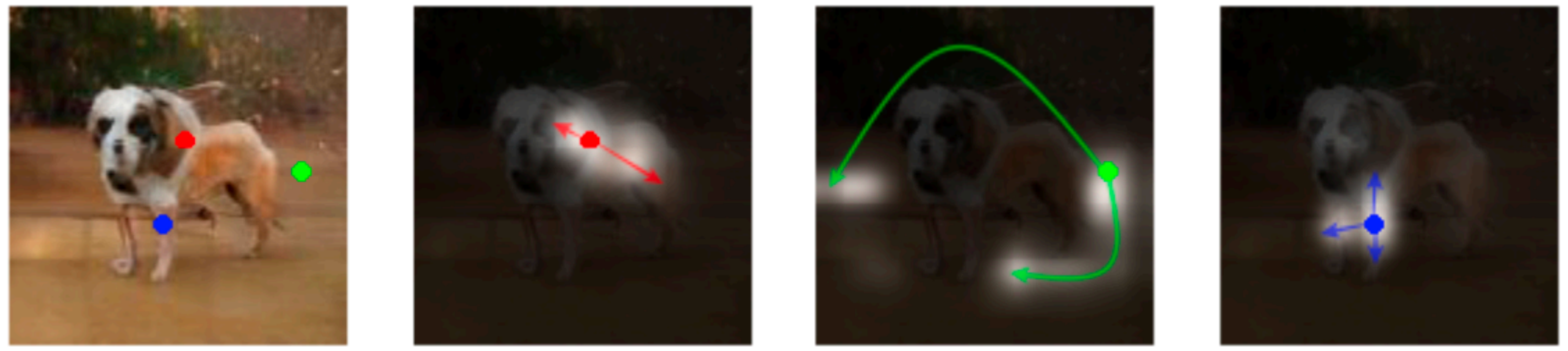
Techniques to stabilize the training of GANs

1) Spectral Normalization
2) Two Time-scale Update Rule (TTUR)

Evaluation Metrics

1) Inception Score
2) Fréchet Inception Distance (FID)





Zhang, Han, et al. "Self-attention generative adversarial networks." *International Conference on Machine Learning*. 2019.

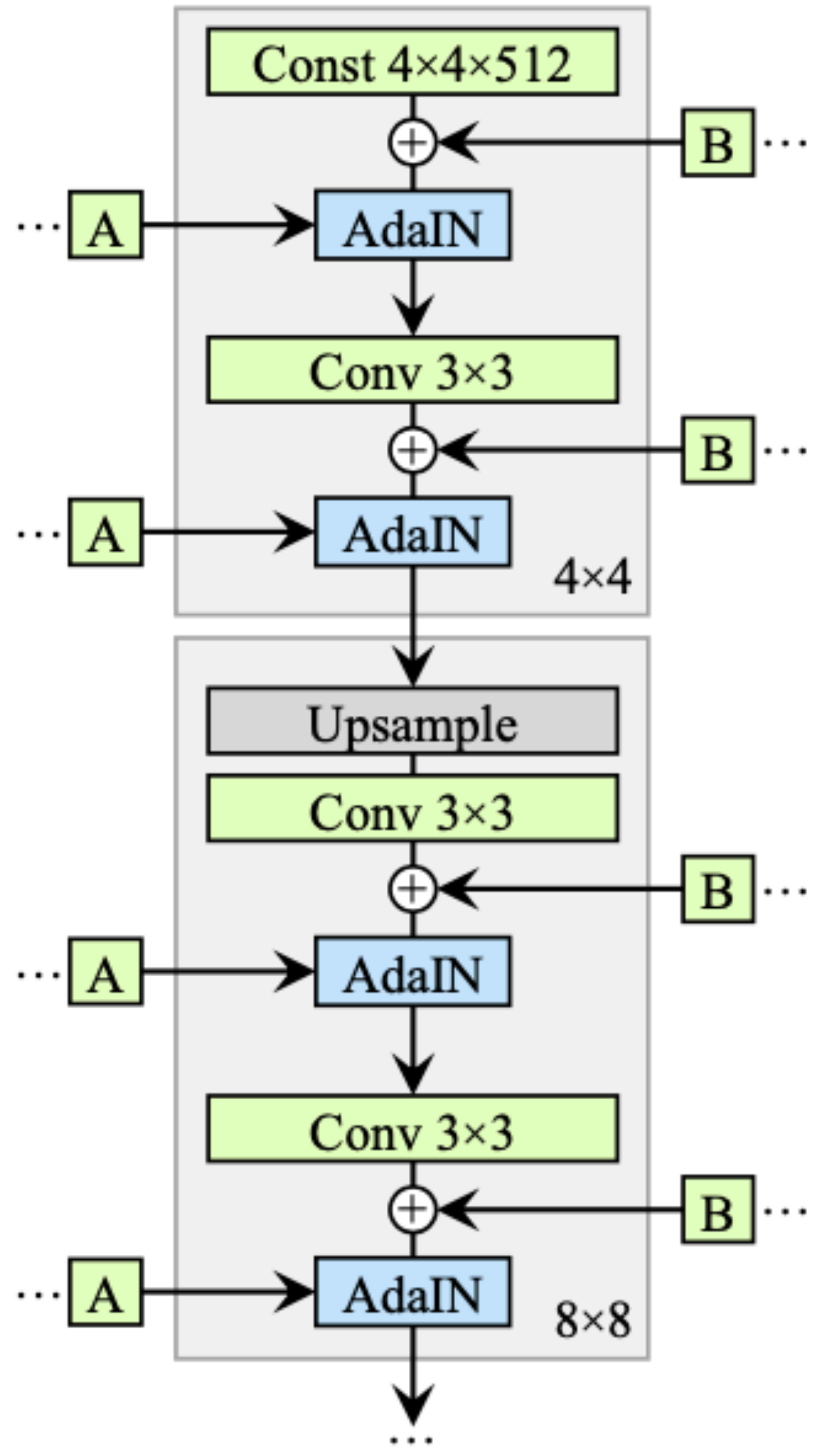
Analyzing and Improving the Image Quality of StyleGAN

$z \in \mathcal{Z} \to$ input latent code
$f \to$ mapping network
$w = f(z) \in \mathcal{W} \to$ intermediate latent code
affine transformations $\to$ styles
$g \to$ synthesis network

$$w'_{ijk} = s_i \cdot w_{ijk} \to \text{modulation}$$

$$w''_{ijk} = w'_{ijk} \Big/ \sqrt{\sum_{i,k} {w'_{ijk}}^2 + \epsilon} \to \text{demodulation}$$

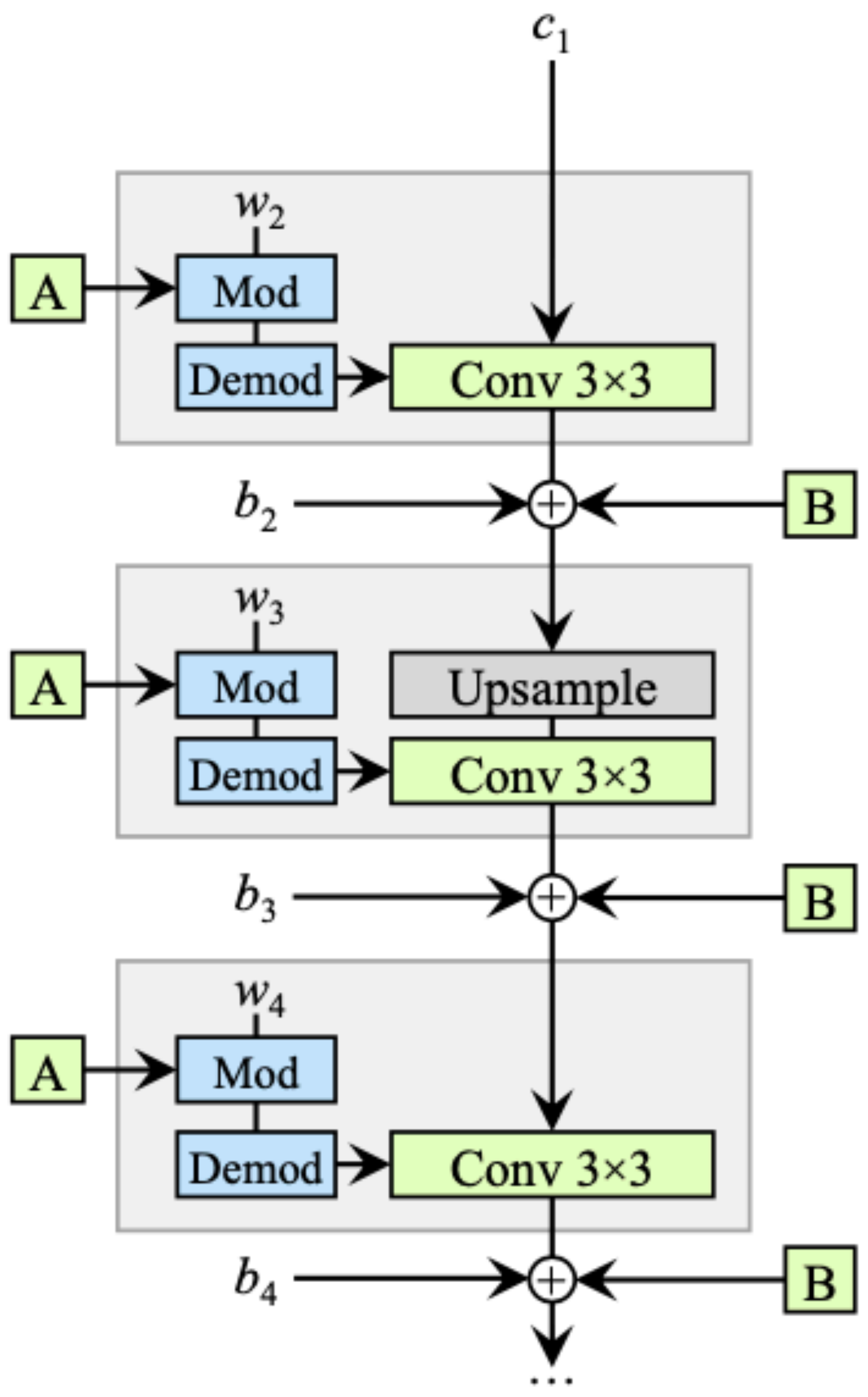
$w, w', w'' \to$ original, modulated, and demodulated weights
$s_i \to$ scale corresponding to i-th input feature map
$j \to$ enumerates the output feature maps
$k \to$ enumerates the spatial footprint of the convolution



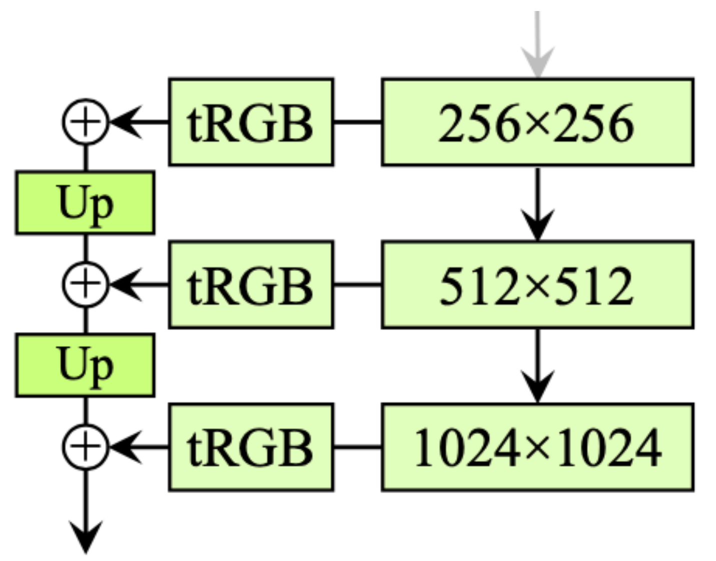
(a) StyleGAN

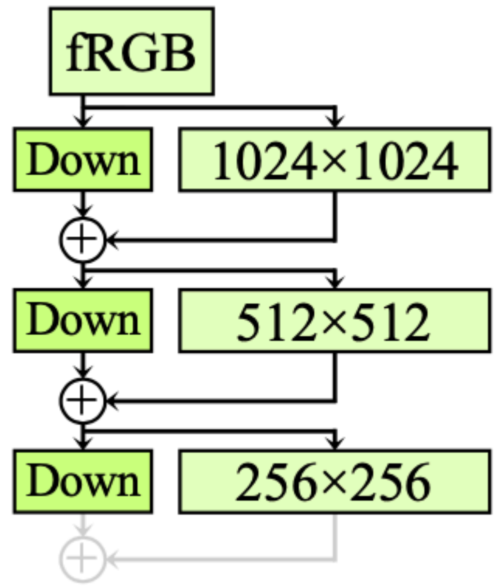
blob-shaped artifacts that resemble water droplets

artifacts related to progressive growing



(b) Input/output skips

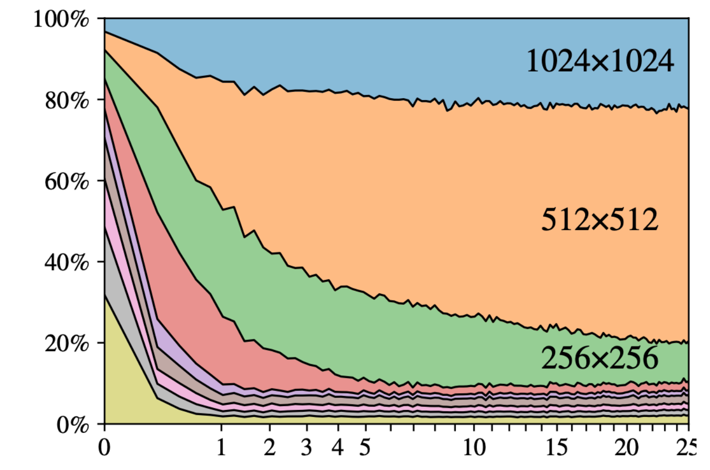




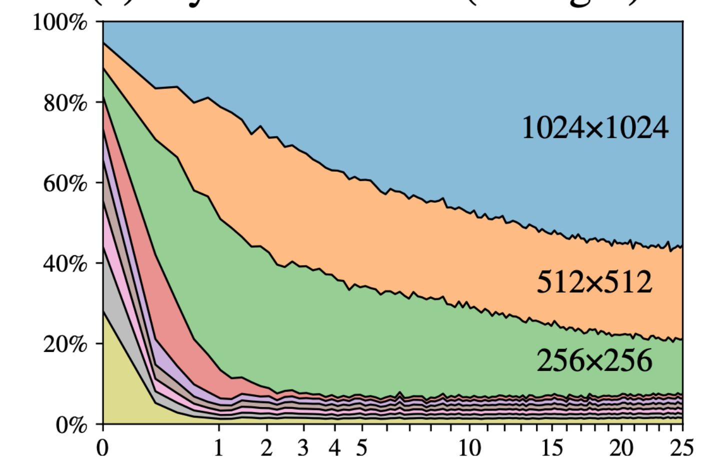
(c) Residual nets

FFHQ, 1024×1024

	Configuration	FID ↓
A	Baseline StyleGAN [21]	4.40
B	+ Weight demodulation	4.39
C	+ Lazy regularization	4.38
D	+ Path length regularization	4.34
E	+ No growing, new G & D arch.	3.31
F	+ Large networks (StyleGAN2)	**2.84**
	Config A with large networks	3.98



(a) StyleGAN-sized (config E)



(b) Large networks (config F)

– Blob-shaped artifacts due to Adaptive Instance Normalization (AdaIN)
– Artifacts related to progressive growing

Path Length Regularization

$a \to$ exponential moving average of $||J_w^T y||_2$

$$\mathbb{E}_{\mathbf{w}}\mathbb{E}_{\mathbf{y}}\left(\left\|\mathbf{J}_{\mathbf{w}}^T\mathbf{y}\right\|_2 - a\right)^2$$

$\mathbf{J_w} = \partial g(\mathbf{w})/\partial \mathbf{w} \to$ Jacobian Matrix

$\mathbf{y} \sim \mathcal{N}(0,\mathbf{I})$ $\quad \mathbf{w} \sim f(\mathbf{z})$

This is minimized when the Jacobian is orthogonal, preserving length and introducing no squeezing along any dimension

Karras, Tero, et al. "Analyzing and improving the image quality of stylegan." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

Boulder

Questions?

YouTube Playlist