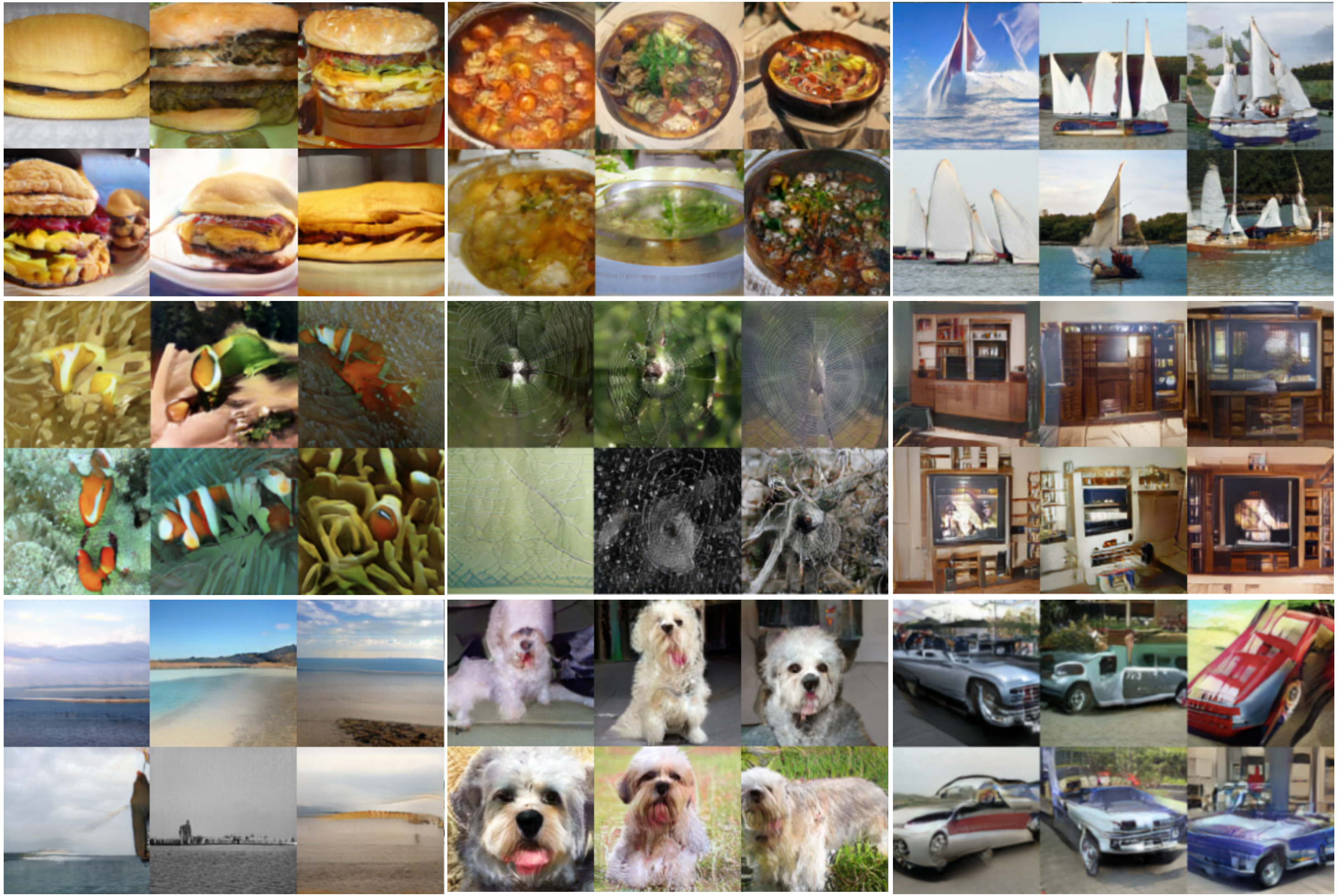
Spectral Normalization for Generative Adversarial Networks

Takeru Miyato,
Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida

Preferred Networks
RITSUMEIKAN UNIVERSITY
NII

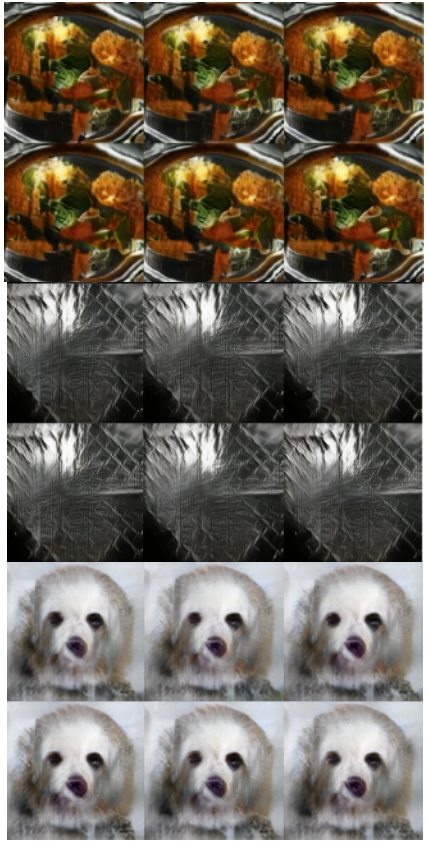


Generated images on ImageNet

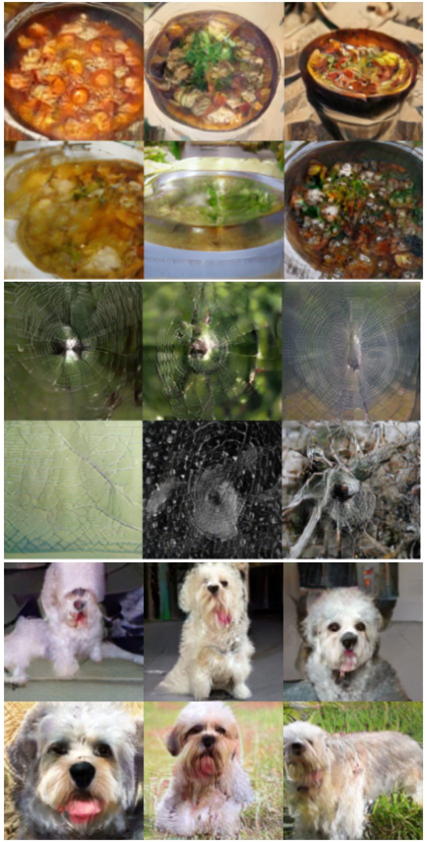
hot pot

spider's web

Dandie Dinmont



AC-GANs
(Odena et al., 2017)



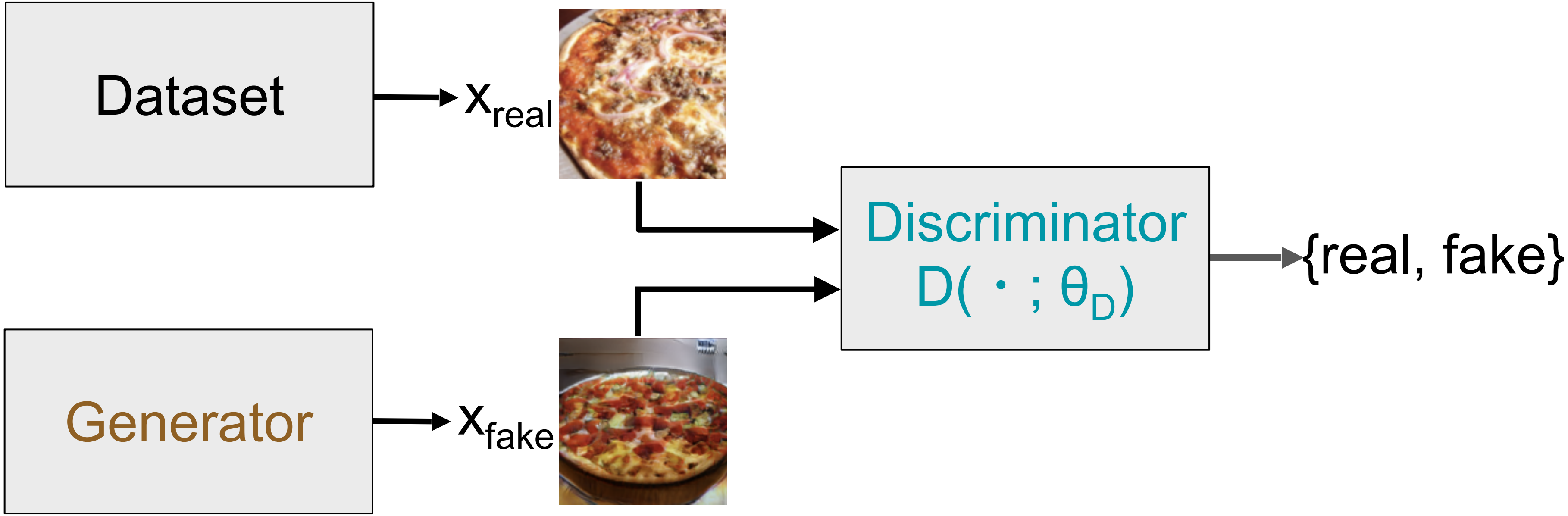
Spectrally normalized GANs(Ours)

cardoon

Agenda

- Method of spectral normalization
- Experimental results
- Some “new” results not shown in the paper

Generative adversarial networks (GANs)



Training the discriminator and generator

Alternately iterate:

- Discriminator update

$$\theta_D \leftarrow \theta_D + \alpha \left\{ \nabla_{\theta_D} \mathrm{E}_{q(x)}[\log D(x;\theta_D)] + \mathrm{E}_{p_{\mathrm{G}}(x)}[\log(1 - D(x;\theta_D))] \right\}$$

$q(x)$: dataset $p_G(x)$: generator

- Generator update

$$\theta_G \leftarrow \theta_G + \beta \nabla_{\theta_G} \mathrm{E}_{p_{\mathrm{G}}(x)}[\log D(x;\theta_D)]$$

Problem: Instability of training

The optimal solution of D given the current G:

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

$$\nabla_x f^*(x) = \underline{\frac{1}{q(x)}}\nabla q(x) + \underline{\frac{1}{p_G(x)}}\nabla p_G(x)$$

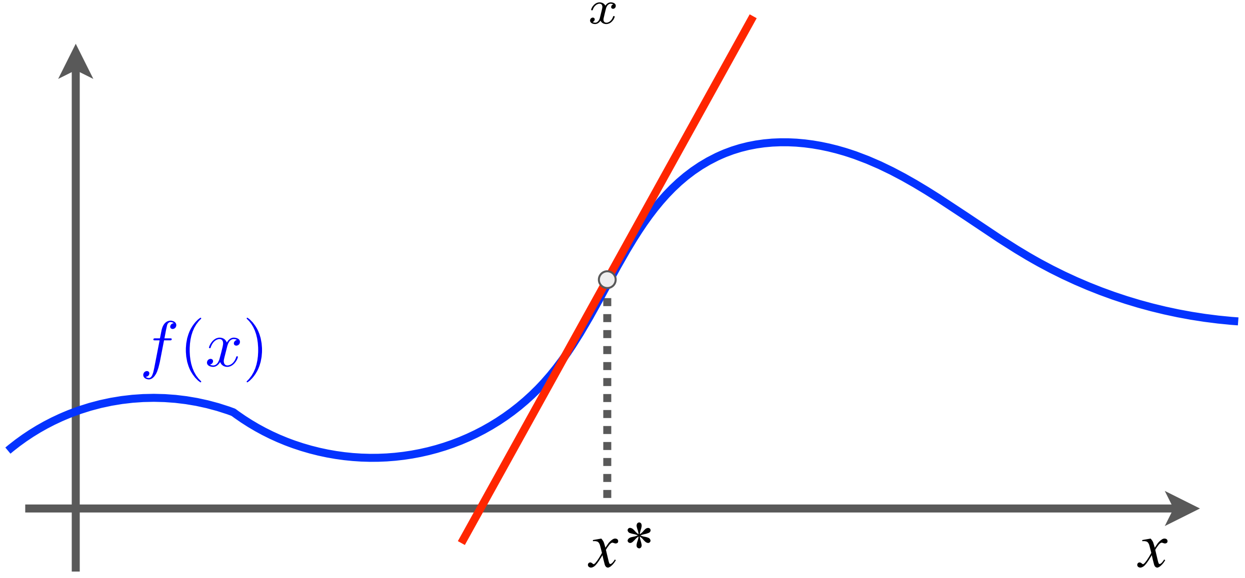
Unbounded when the either denominator is 0

Need to impose Lipschitz constraint on f(x)

Lipschitz constant $\|f\|_{\mathrm{Lip}}$

$$\|f\|_{\mathrm{Lip}} := \max_x \|\nabla f(x)\|$$

$$x^* := \arg\max_x \|\nabla f(x)\|$$



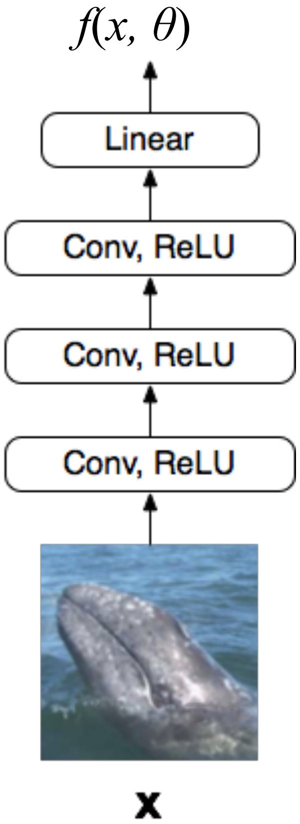
Formulation of the discriminator

Discriminator

$$D(x,\theta) := \mathcal{A}(f(x,\theta))$$

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

- x : input
- $\mathcal{A}$: top activation function (e.g. sigmoid)
- $\theta = \{W^1, \dots, W^{L+1}\}$: weight matrices
- $a_1, \dots, a_L$: activation functions (e.g. ReLU)



Impose Lipschitz constraint

Lipschitz constant of $f(x,\theta)$ is bounded by the product of the **spectral norm of** W (=the largest singular value of W).

$$\|f\|_{\mathrm{Lip}} \leq \prod_{l=1}^{L+1} \sigma(W^l) \quad (\text{Assume } \|a_l\|_{\mathrm{Lip}} = 1, \text{for } l = \{1, ..., L\})$$

Spectral norm : $\sigma(A) := \max_{\boldsymbol{h}:\boldsymbol{h}\neq\boldsymbol{0}} \frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2} = \max_{\|\boldsymbol{h}\|_2\leq 1} \|A\boldsymbol{h}\|_2$

Spectral normalization (SN)

Normalize the weight by its **spectral norm** $\sigma(W)$:

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

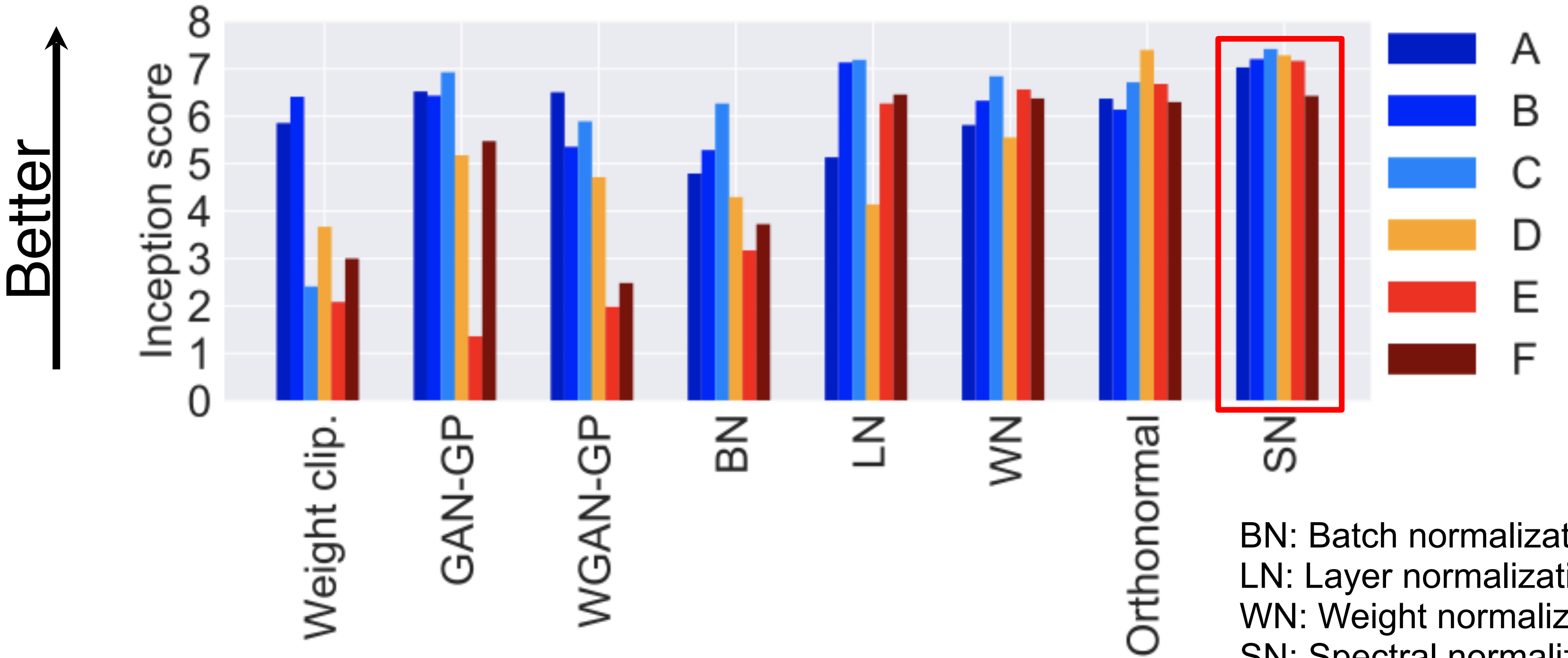
$$(\Rightarrow \sigma(\bar{W}_{\mathrm{SN}}) = 1)$$

Advantages of spectral normalization (SN)

- Easy to incorporate SN into existing discriminator model
  Just replace the standard layer with the spectrally normalized layer

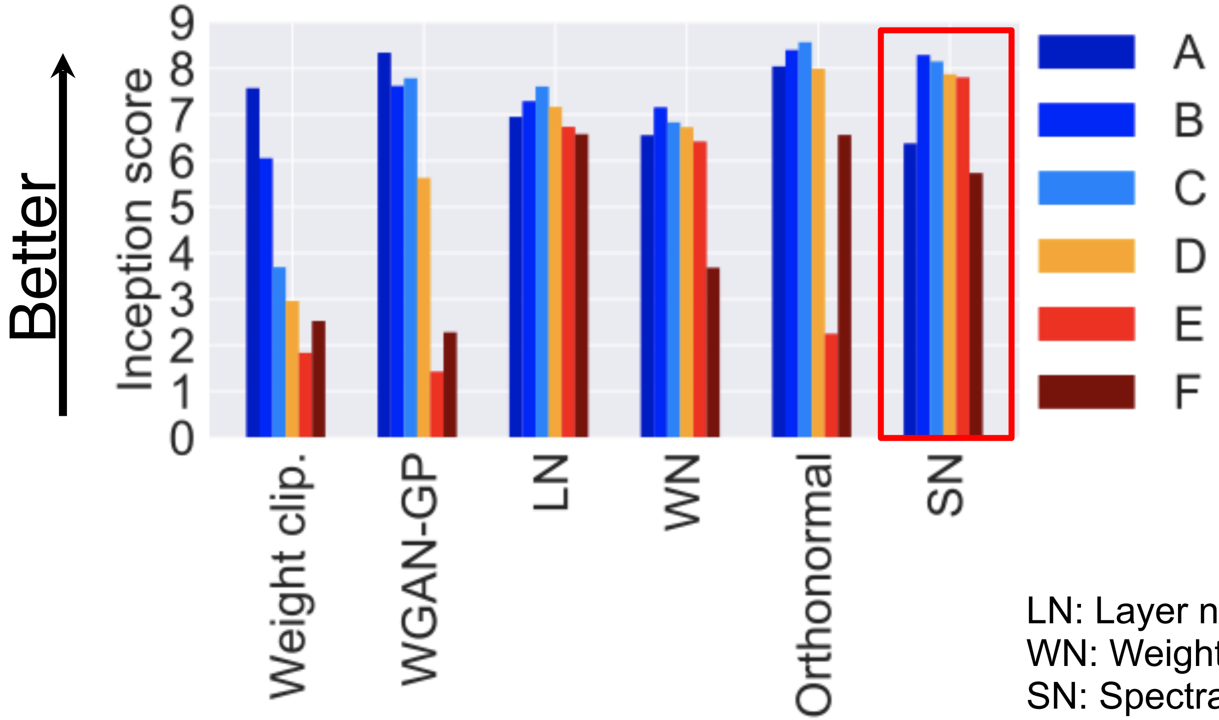
- Computationally light
  $\sigma(W)$ can be approximated easily with small computational cost

Results (CIFAR-10)



BN: Batch normalization
LN: Layer normalization
WN: Weight normalization
SN: Spectral normalization

Results (STL-10)



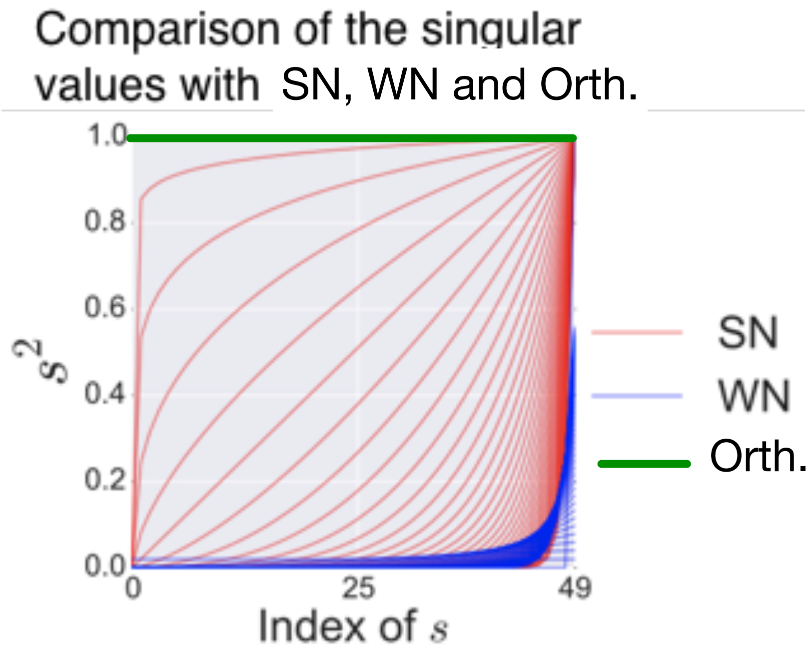
LN: Layer normalization
WN: Weight normalization
SN: Spectral normalization

Spectral Normalization explained from “Spectral” perspective

SN: spectral normalization
WN: weight normalization
Orth.: orthonormal regularization

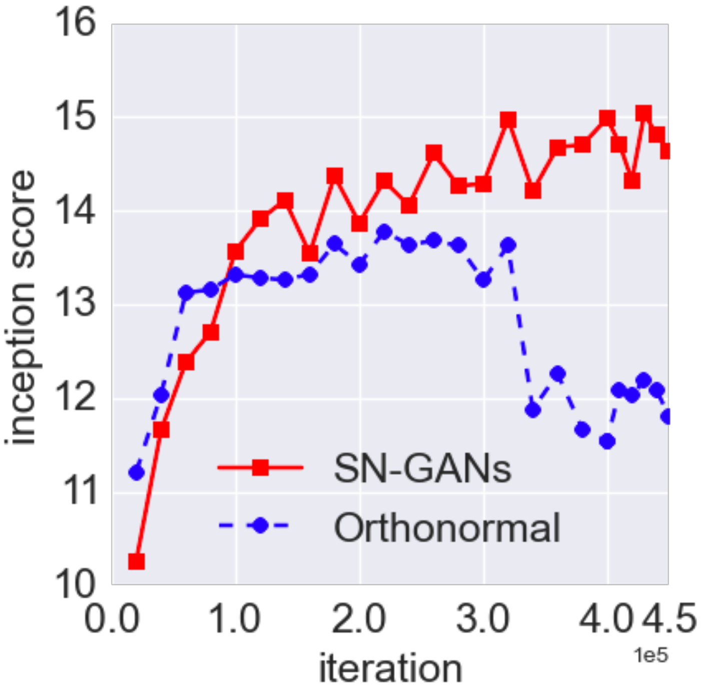
WN requires “the **sum** of singular values of each weight matrix to be some **constant**”.

This will inadvertently encourage the weight matrix to have low rank
= The discriminator will be looking at very few features of the images.



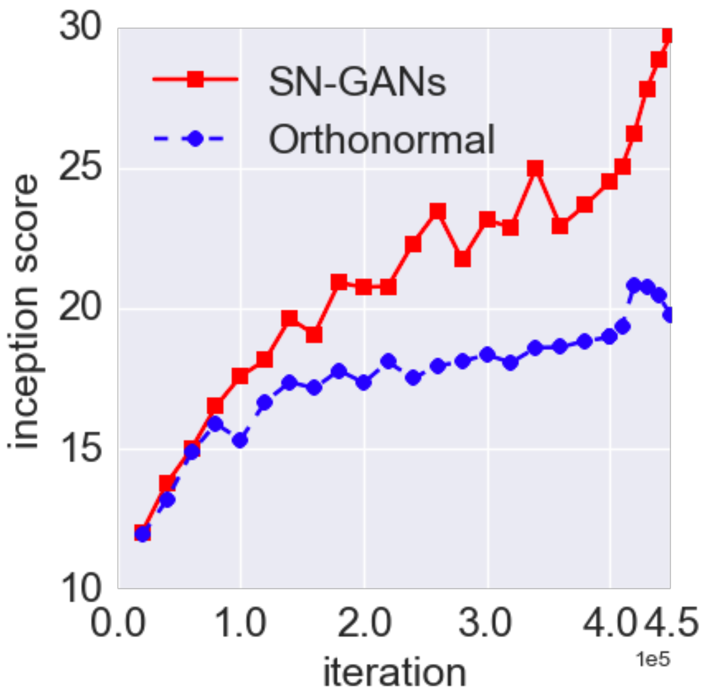
(s: singular value)

Results (ImageNet, 128px)



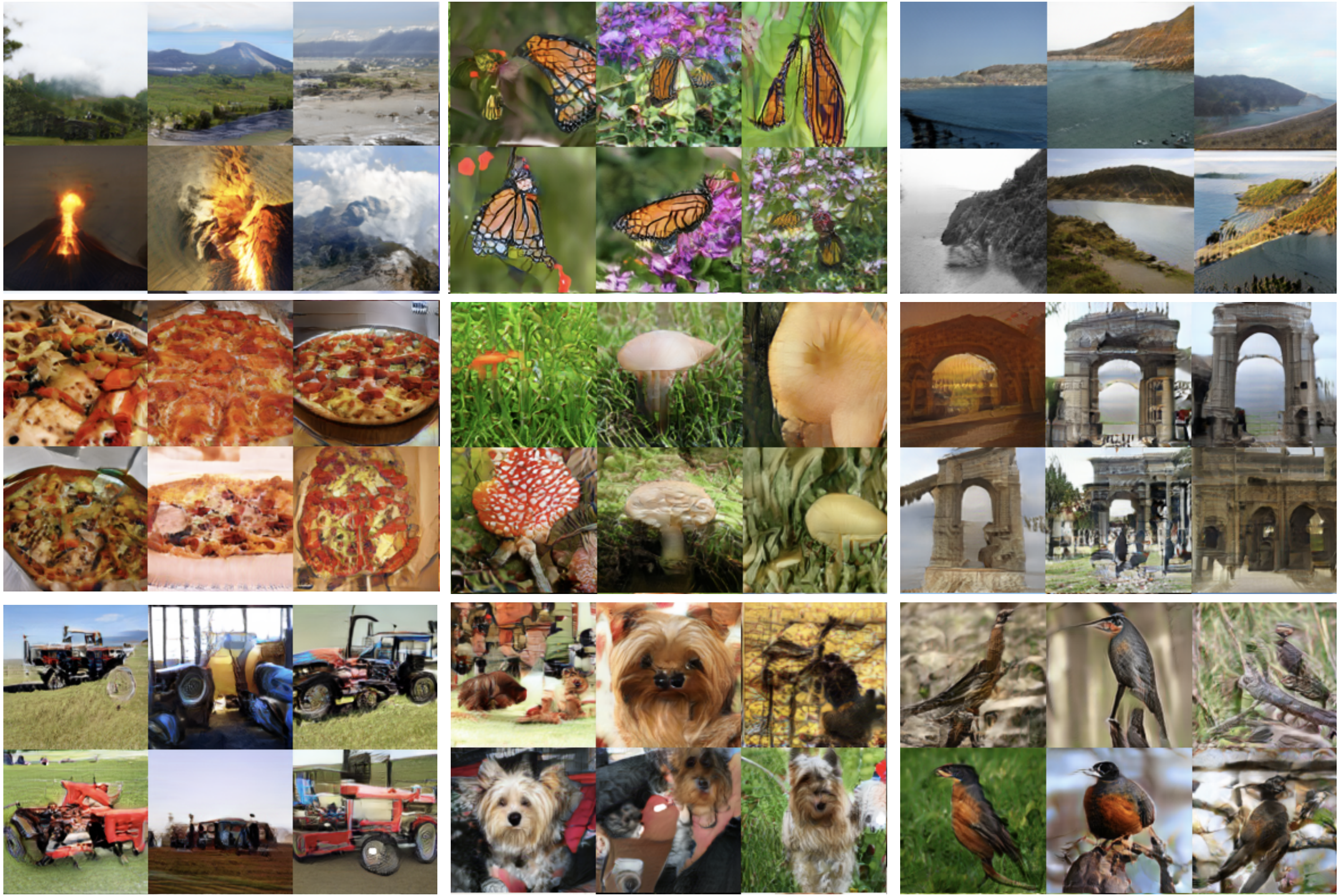
Unconditional GANs

Results (ImageNet, 128px)



Conditional GANs with projection discriminator (Miyato and Koyama, ICLR2018)

Generated images (128px, ImageNet)



Generated Images (256px):
(trained only on dog and cat images)

English springer

Pekinese

Tiger cat

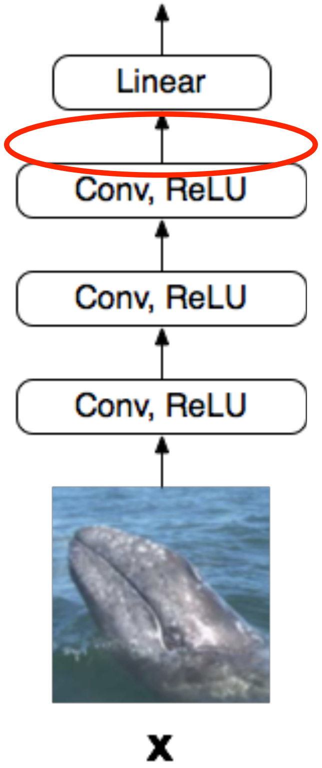
Linx

Nearest neighbors in the discriminator feature space

We used the discriminator trained
"without category label supervision"

$$f(\boldsymbol{x}, \theta) := \boldsymbol{w}^{\mathrm{T}} \phi(\boldsymbol{x})$$

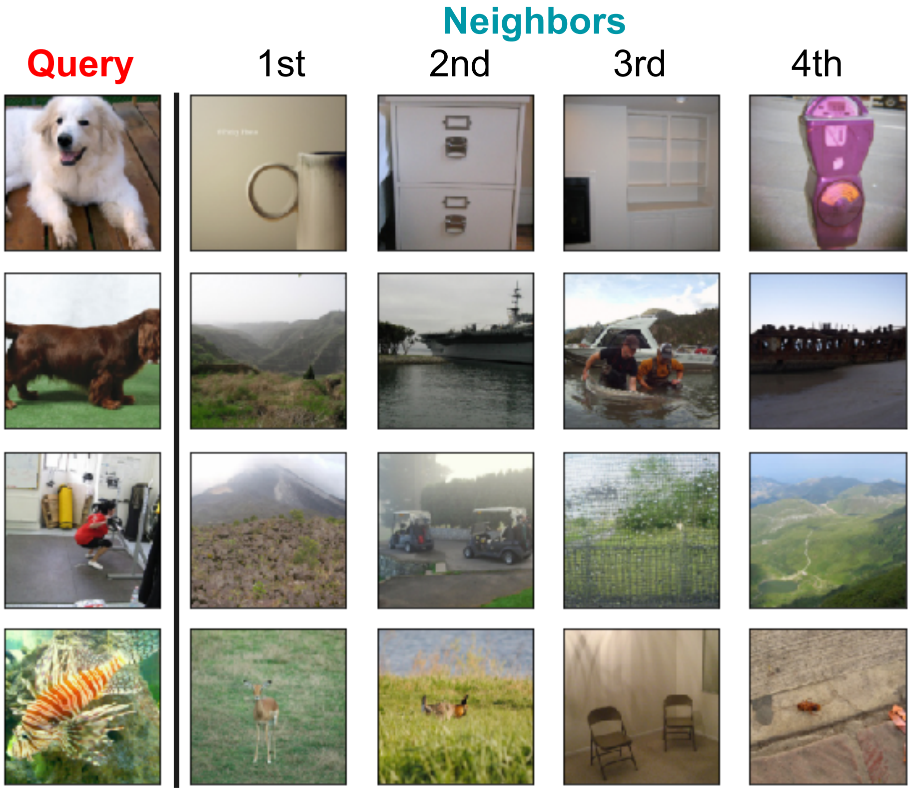
$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) := |\phi(\boldsymbol{x}_1) - \phi(\boldsymbol{x}_2)|$$



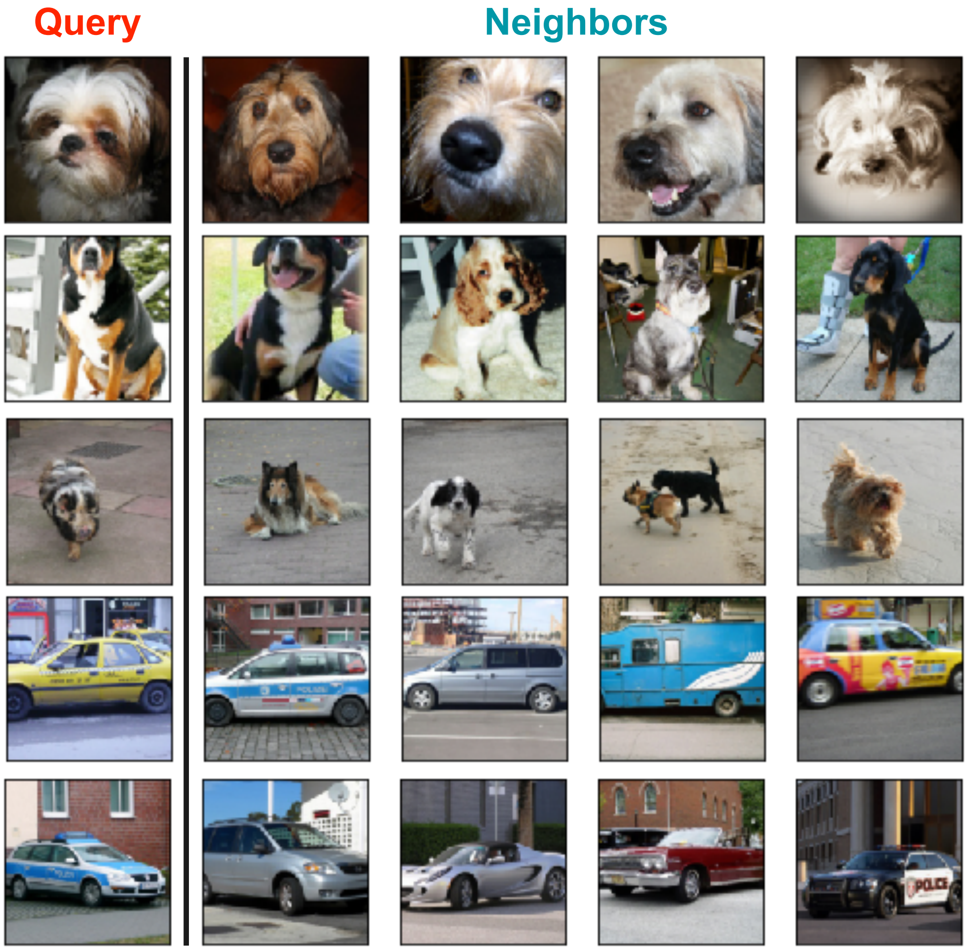
Nearest neighbors with L1 distance

(a) in the **discriminator feature** space

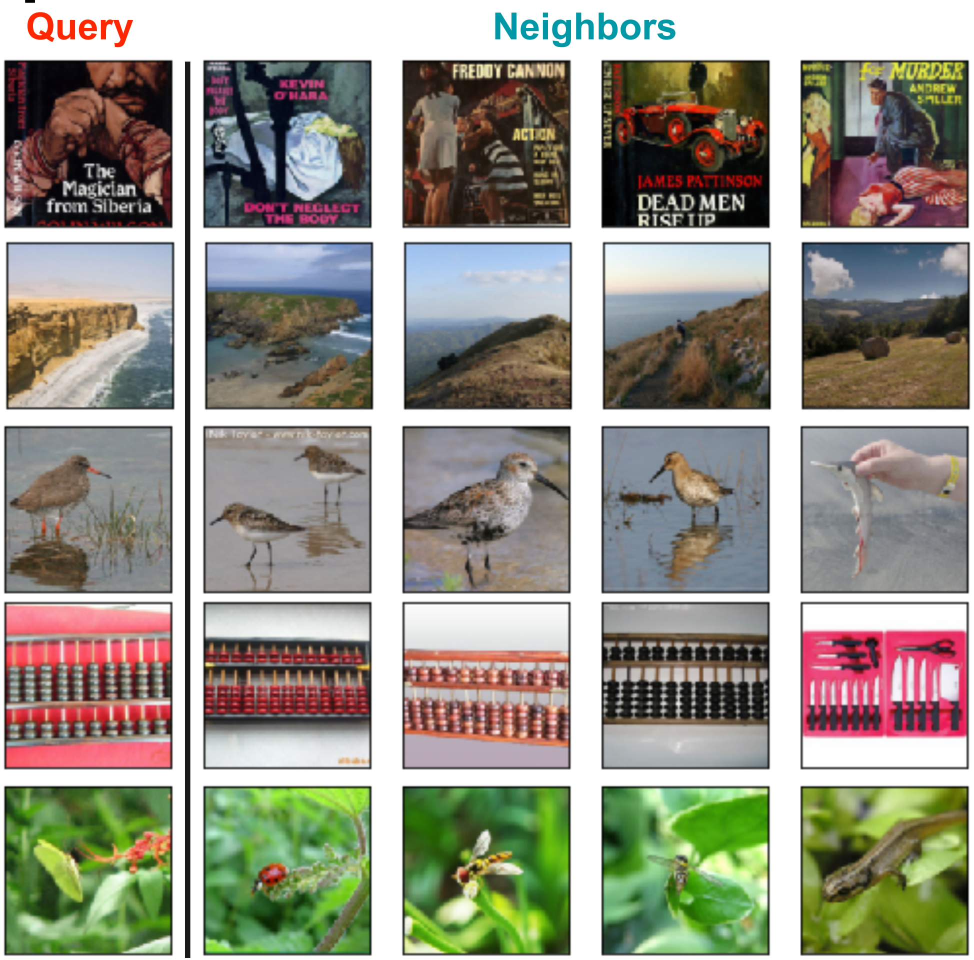
(b) in **raw pixel** space



Other examples



Other examples



Thank you!

- Code, generated images, and pretrained models are available at: https://github.com/pfnet-research/sngan_projection
- Chainer-GAN-lib: https://github.com/pfnet-research/chainer-gan-lib