



Boulder

Generative Networks; Conditional GANs



[YouTube Playlist](#)

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu



Boulder

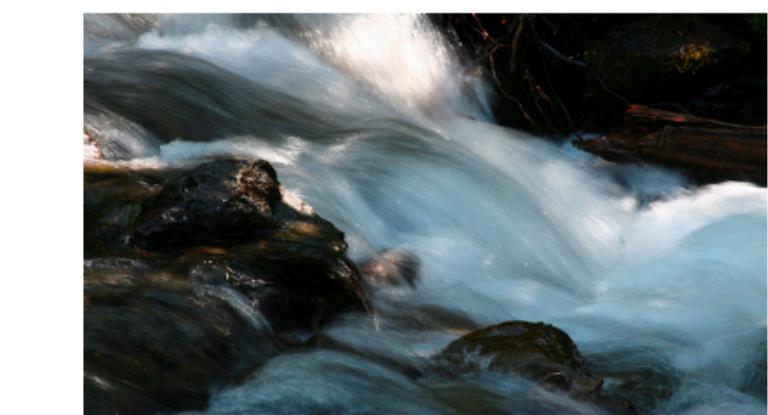
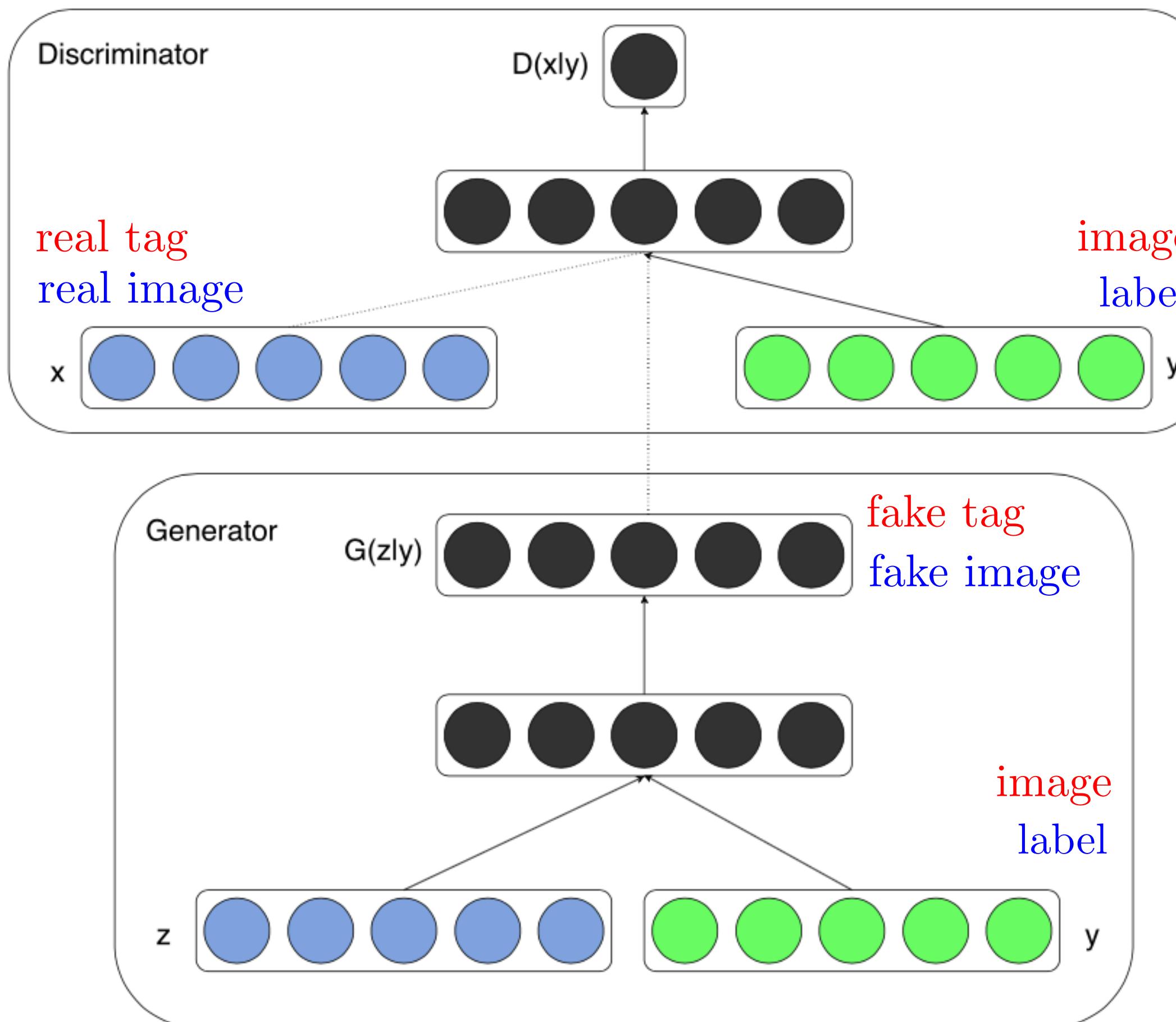
Conditional Generative Adversarial Nets



YouTube Video

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

One-To-Many Mapping



User tags + annotations	Generated tags
montanha, trem, inverno, frio, people, male, plant life, tree, structures, transport, car	<p style="color: red;">cosine similarity</p> <p>taxi, passenger, line, transportation, railway station, passengers, railways, signals, rail, rails</p>
food, raspberry, delicious, homemade	chicken, fattening, cooked, peanut, cream, cookie, house made, bread, biscuit, bakes
water, river	creek, lake, along, near, river, rocky, treeline, valley, woods, waters
people, portrait, female, baby, indoor	love, people, posing, girl, young, strangers, pretty, women, happy, life



Boulder



[YouTube Video](#)

Context Encoders: Feature Learning by Inpainting



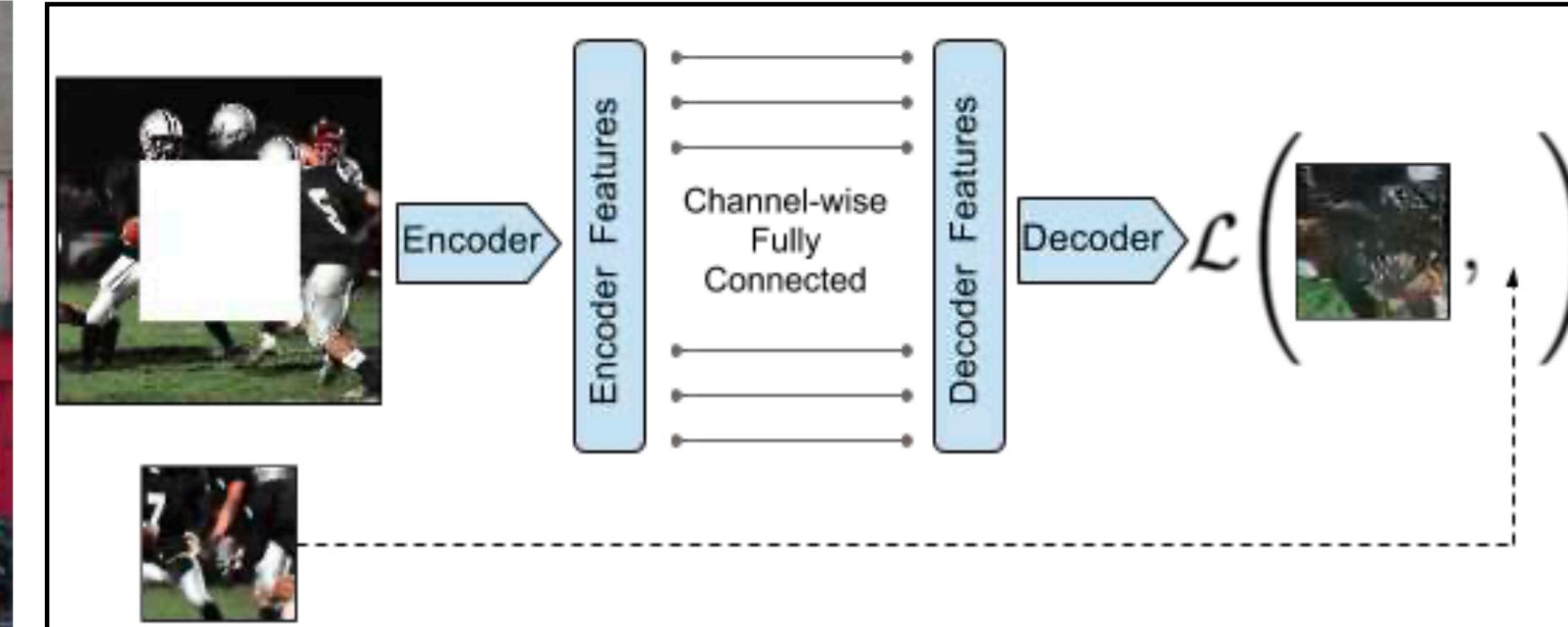
(a) Input context



(c) Context Encoder
(L2 loss)

Loss function
 $x \rightarrow$ ground truth image
 $F \rightarrow$ context encoder

$F(x) \rightarrow$ output
 $\hat{M} \rightarrow$ binary mask (dropped image region)
 1 whenever a pixel is dropped and 0 for input pixels



Encoder
 $\mathbb{R}^{227 \times 227 \times 3} \ni I \triangleright \underbrace{\text{AlexNet(pool5)}}_{\substack{5 \text{ conv layers} \\ 6 \times 6 \times 256 = 9,216 \text{ from scratch}} \in \mathbb{R}^{6 \times 6 \times 256}}$

Channel-wise fully-connected layer

input: m feature maps of size $n \times n$

output: m feature maps of size $n \times n$

$$Y_j = \underbrace{W_j}_{n^2 \times 1} \underbrace{X_j}_{n^2 \times n^2} \in \mathbb{R}^{n^2 \times 1}$$

$mn^4 \rightarrow$ number of parameters (rather than m^2n^4)

Decoder

5 up-conv layers

$$\mathcal{L}_{rec}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2$$

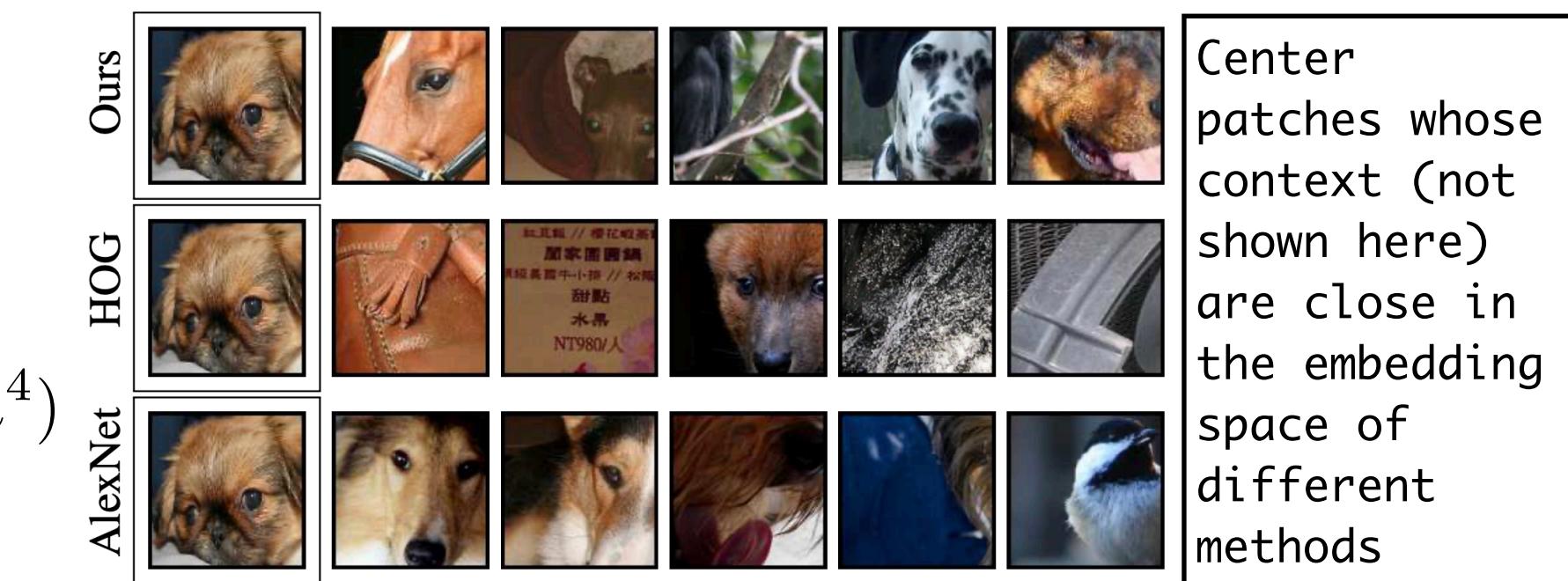
$$\mathcal{L}_{adv} = \max_D \mathbb{E}_{x \in \mathcal{X}} [\log(D(x))]$$

$$+ \log(1 - D(F((1 - \hat{M}) \odot x)))]$$

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv}$$



Method	Mean L1 Loss	Mean L2 Loss	PSNR (higher better)
NN-inpainting (HOG features)	19.92%	6.92%	12.79 dB
NN-inpainting (our features)	15.10%	4.30%	14.70 dB
Our Reconstruction (joint)	10.33%	2.35%	17.59 dB



Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	78.2%	56.8%	48.0%
Random Gaussian	initialization	< 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal <i>et al.</i> [1]	egomotion	10 hours	52.9%	41.8%	-
Doersh <i>et al.</i> [7]	context	4 weeks	55.3%	46.6%	-
Wang <i>et al.</i> [39]	motion	1 week	58.4%	44.0%	-
Ours	context	14 hours	56.5%	44.5%	29.7%



Boulder

Conditional Image Synthesis with Auxiliary Classifier GANs

Generative Adversarial Network (GAN)

$$X_{fake} = G(z)$$

$$P(S | X) = D(X)$$

$\mathcal{L}_{\text{source}}$

$$L = E[\log P(S = \text{real} | X_{real})] + E[\log P(S = \text{fake} | X_{fake})]$$

Auxiliary Classifier GAN (AC-GAN)

$$X_{fake} = G(c, z)$$

class label \mathcal{L} noise

$$P(S | X), P(C | X) = D(X)$$

$$L_S = E[\log P(S = \text{real} | X_{real})] + E[\log P(S = \text{fake} | X_{fake})]$$

$$L_C = E[\log P(C = c | X_{real})] + E[\log P(C = c | X_{fake})]$$

$L_S \rightarrow$ log-likelihood of the correct source

$L_C \rightarrow$ log-likelihood of the correct class

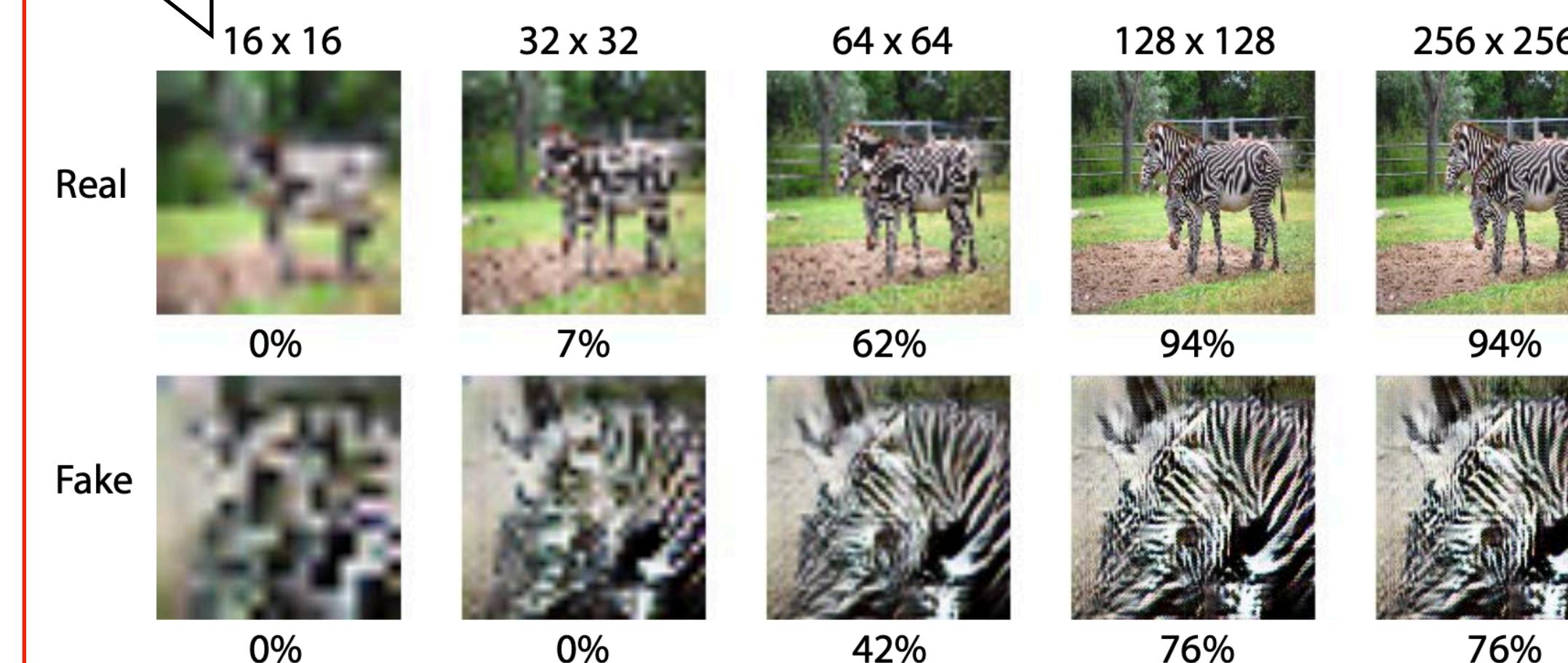
$$D^* = \arg \max_D L_S + L_C$$

$$G^* = \arg \max_G L_C - L_S$$

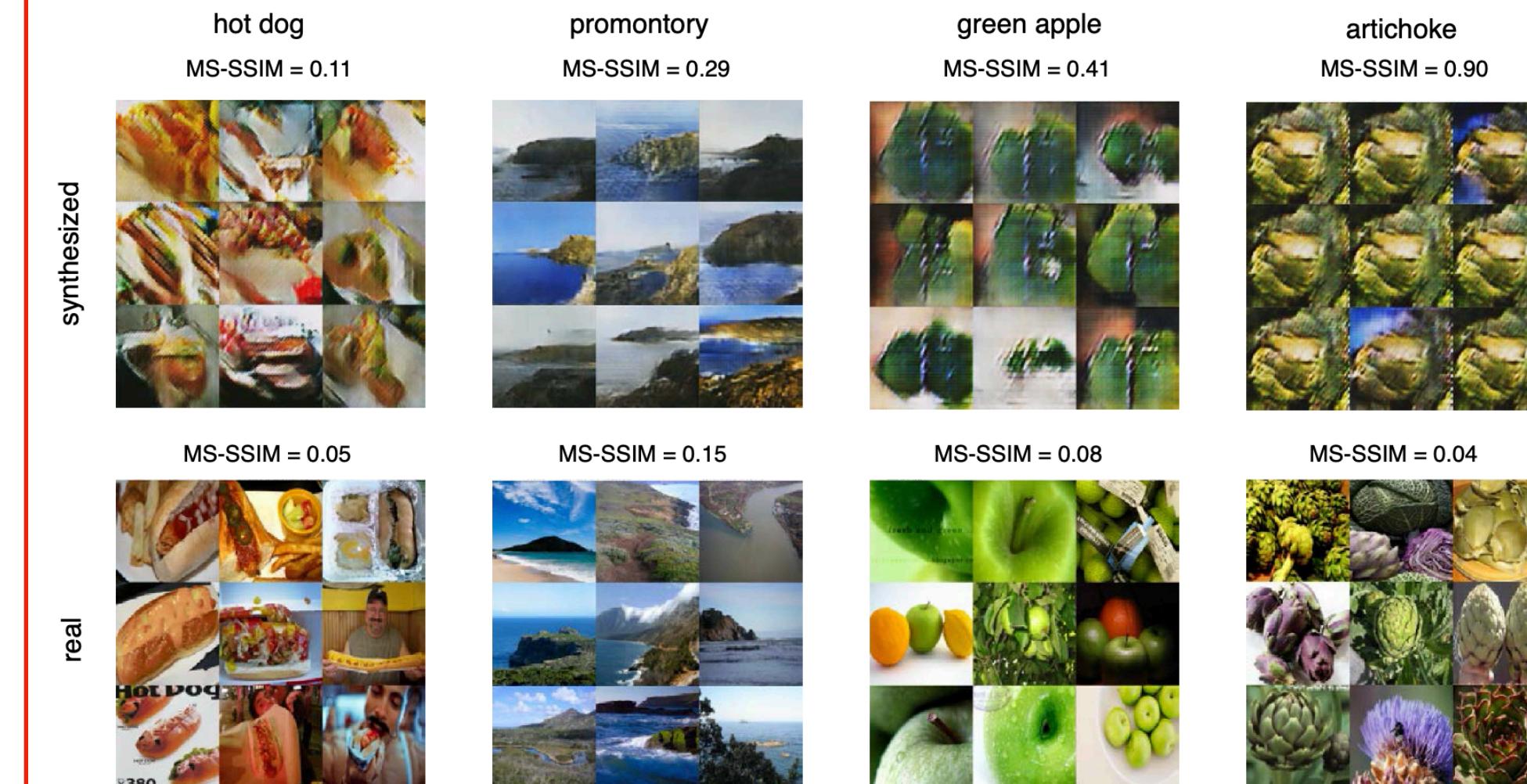
All ImageNet experiments are conducted using an ensemble of 100 AC-GANs, each trained on a 10-class split.

Generating High Resolution Images Improves Discriminability

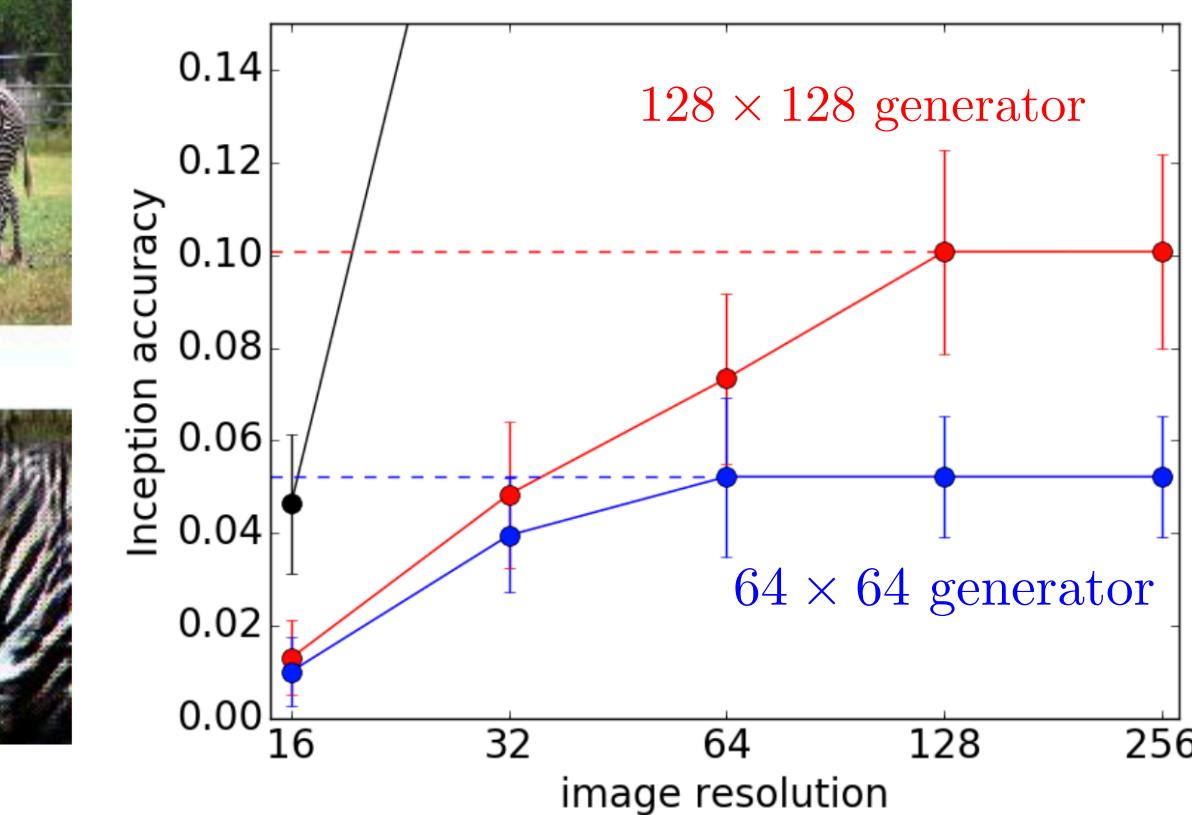
spatial resolution artificially decreased by bilinear interpolation



Measuring the Diversity of Generated Images



Synthesized Images \triangleright Inception Model fraction of images for which the Inception network assigns the correct label



Multi-Scale Structural Similarity (MS-SSIM) attempts to discount aspects of an image that are not important for human perception. MS-SSIM values range between 0.0 and 1.0; higher MS-SSIM values correspond to perceptually more similar images.

As a proxy for image diversity, measure the MS-SSIM scores between 100 randomly chosen pairs of images within a given class. Samples from classes that have higher diversity result in lower mean MS-SSIM scores.

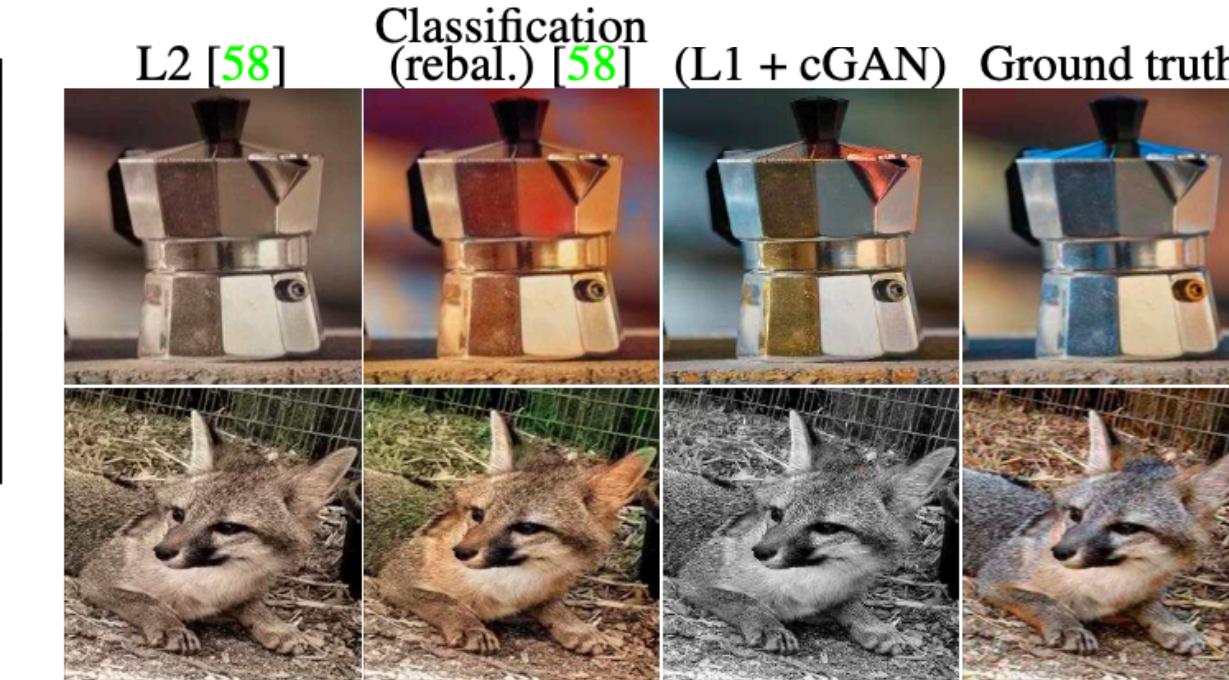
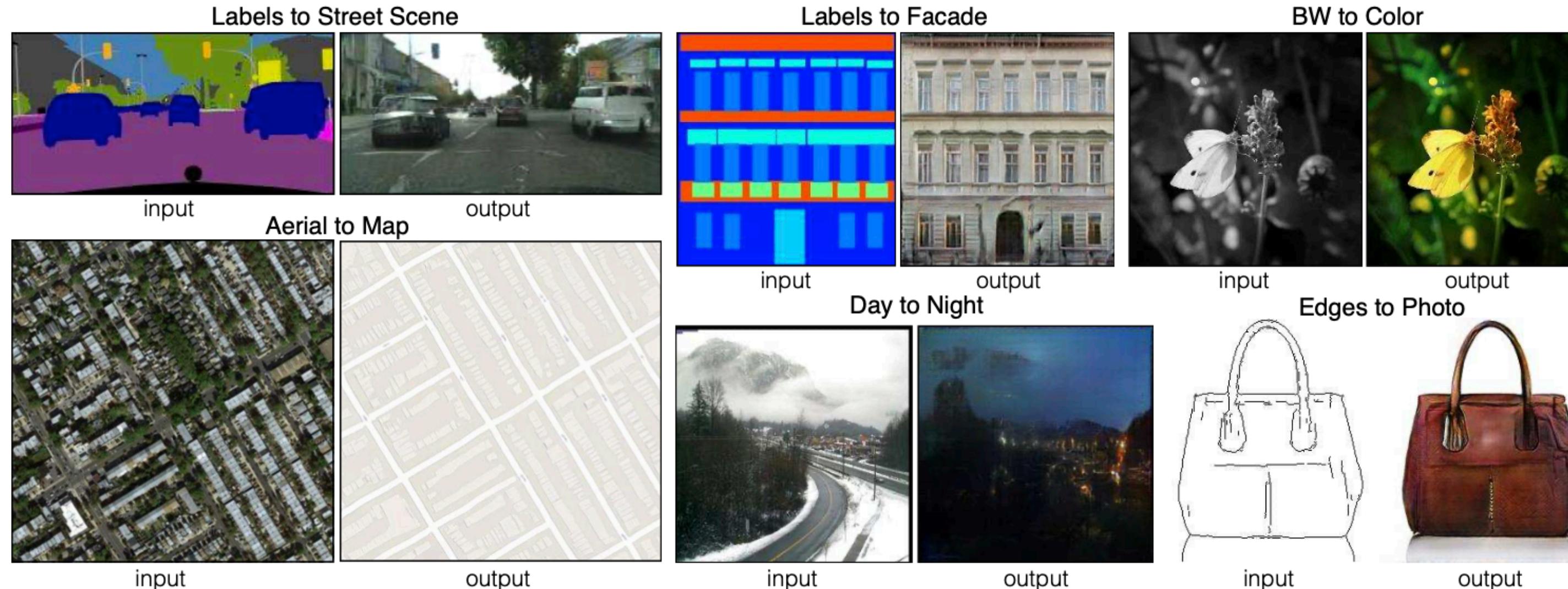


Boulder

Image-to-Image Translation with Conditional Adversarial Networks



[YouTube Video](#)

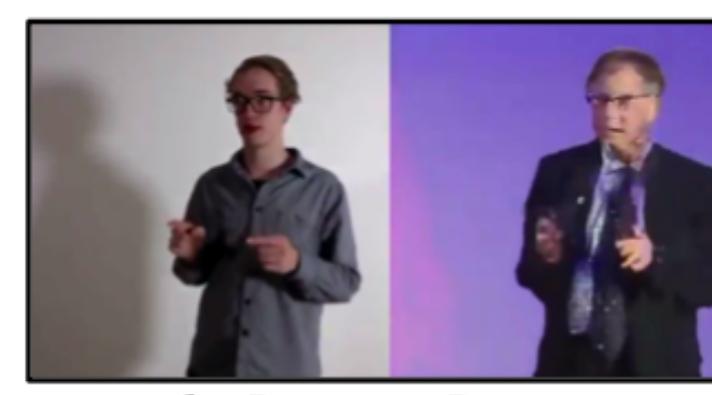


Background removal



by Kaihu Chen

“Do as I do”



by Brannon Dorsey

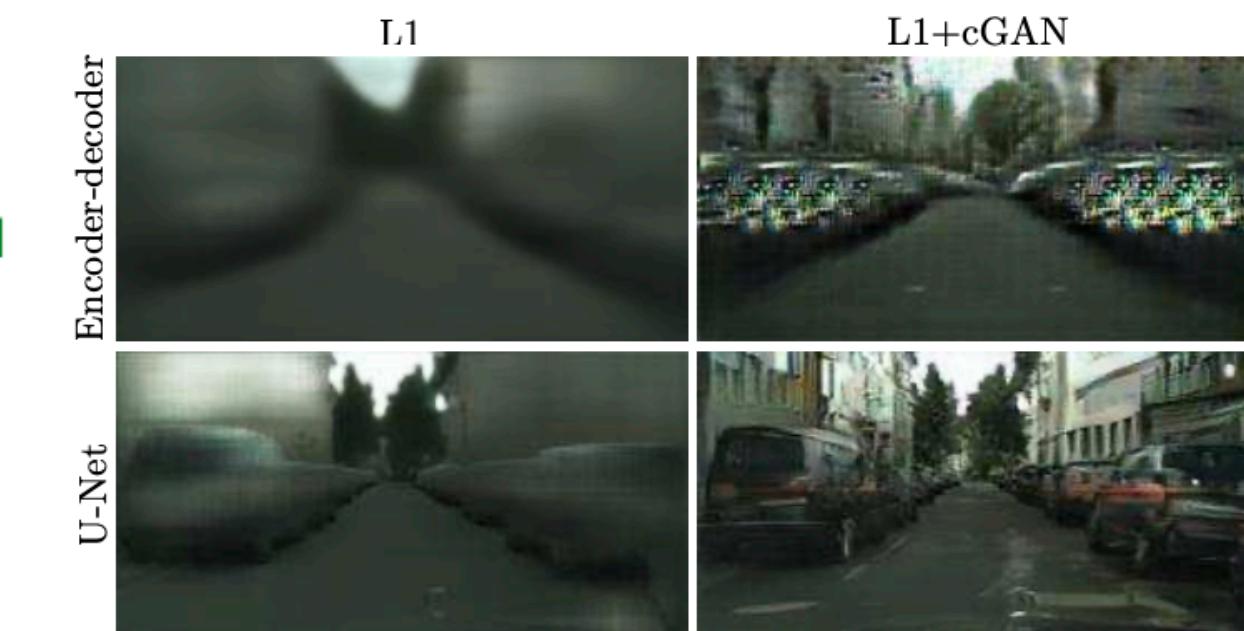
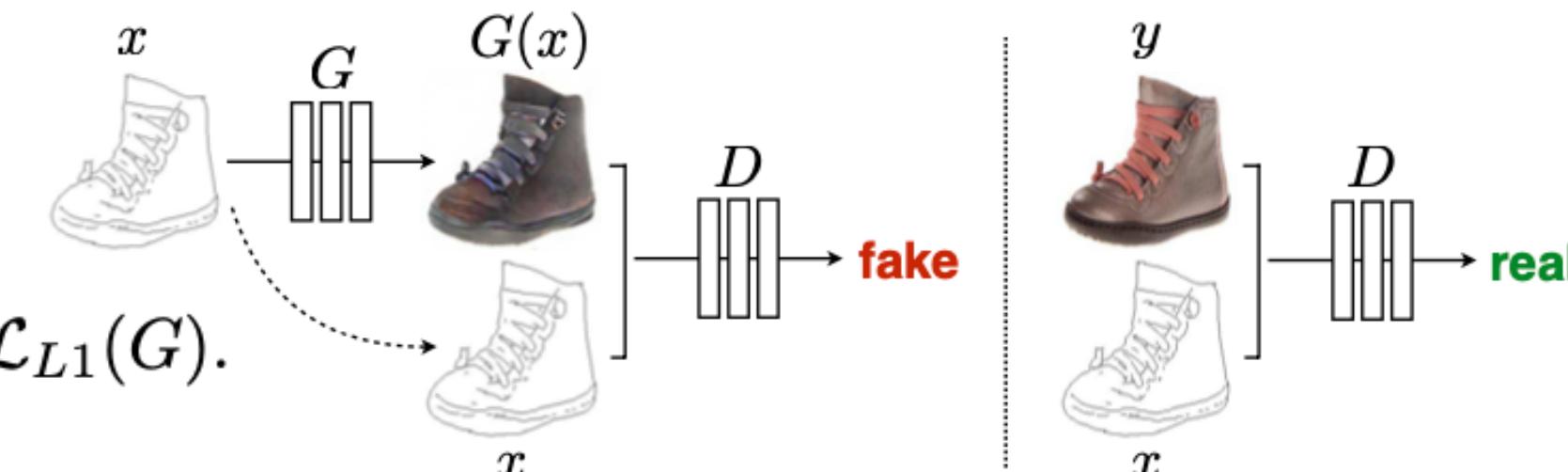
$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))] \rightarrow \text{objective of a conditional GAN}$$

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D)$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1].$$

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_y[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x, z)))] \rightarrow \text{an unconditional variant}$$



Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

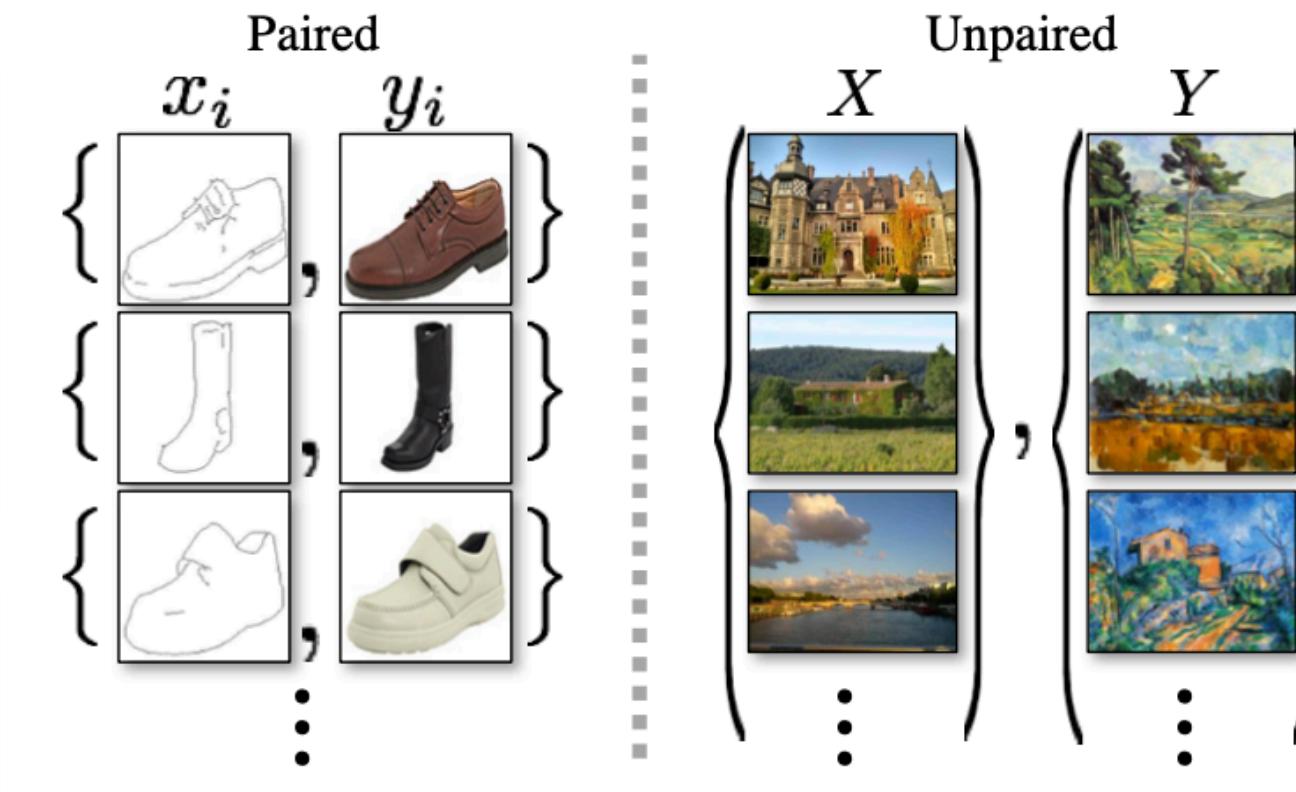
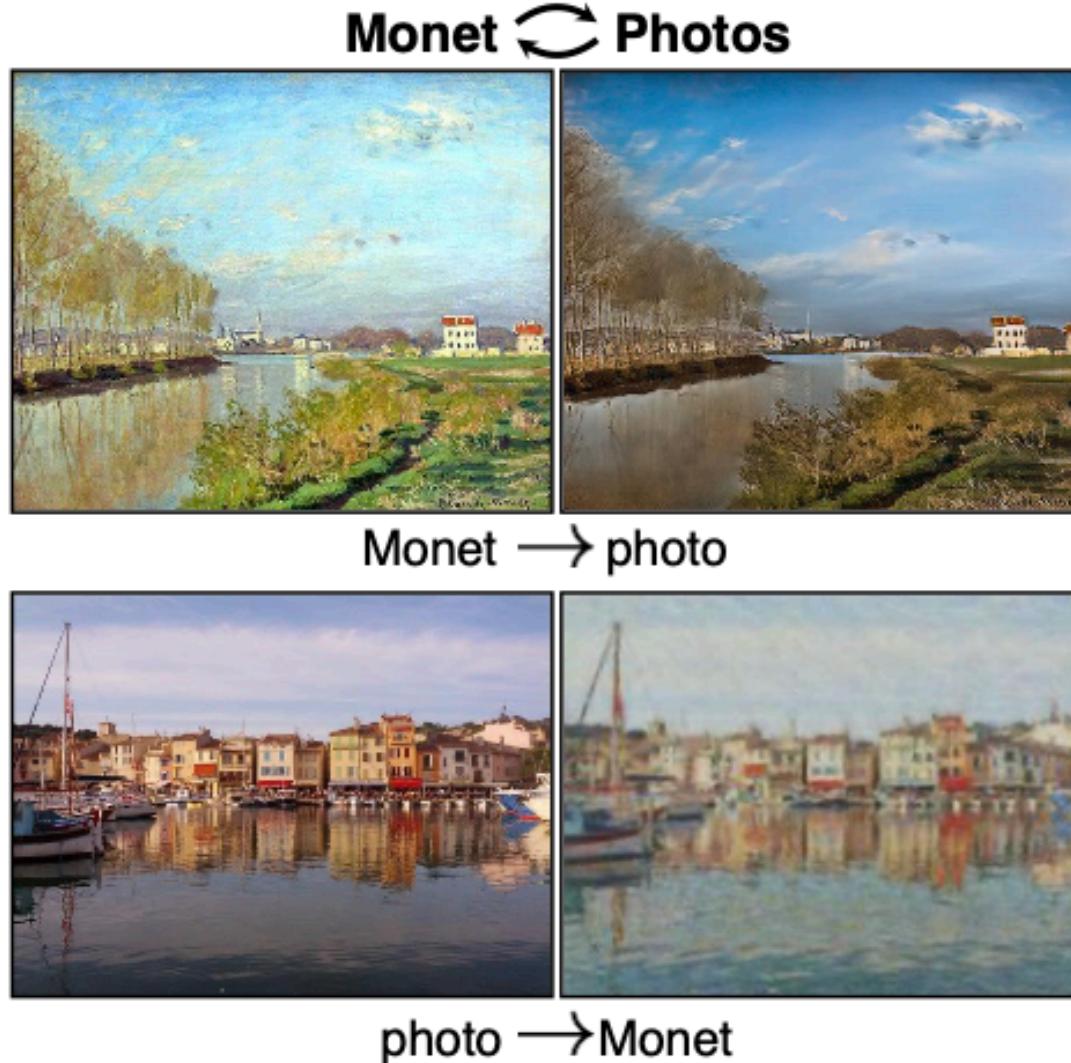


Boulder

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks



[YouTube Video](#)



$$G : X \rightarrow Y$$

$$F : Y \rightarrow X$$

$$F(G(x)) \approx x \text{ and } G(F(y)) \approx y.$$

D_X → aims to discriminate between $\{x\}$ and $\{F(y)\}$

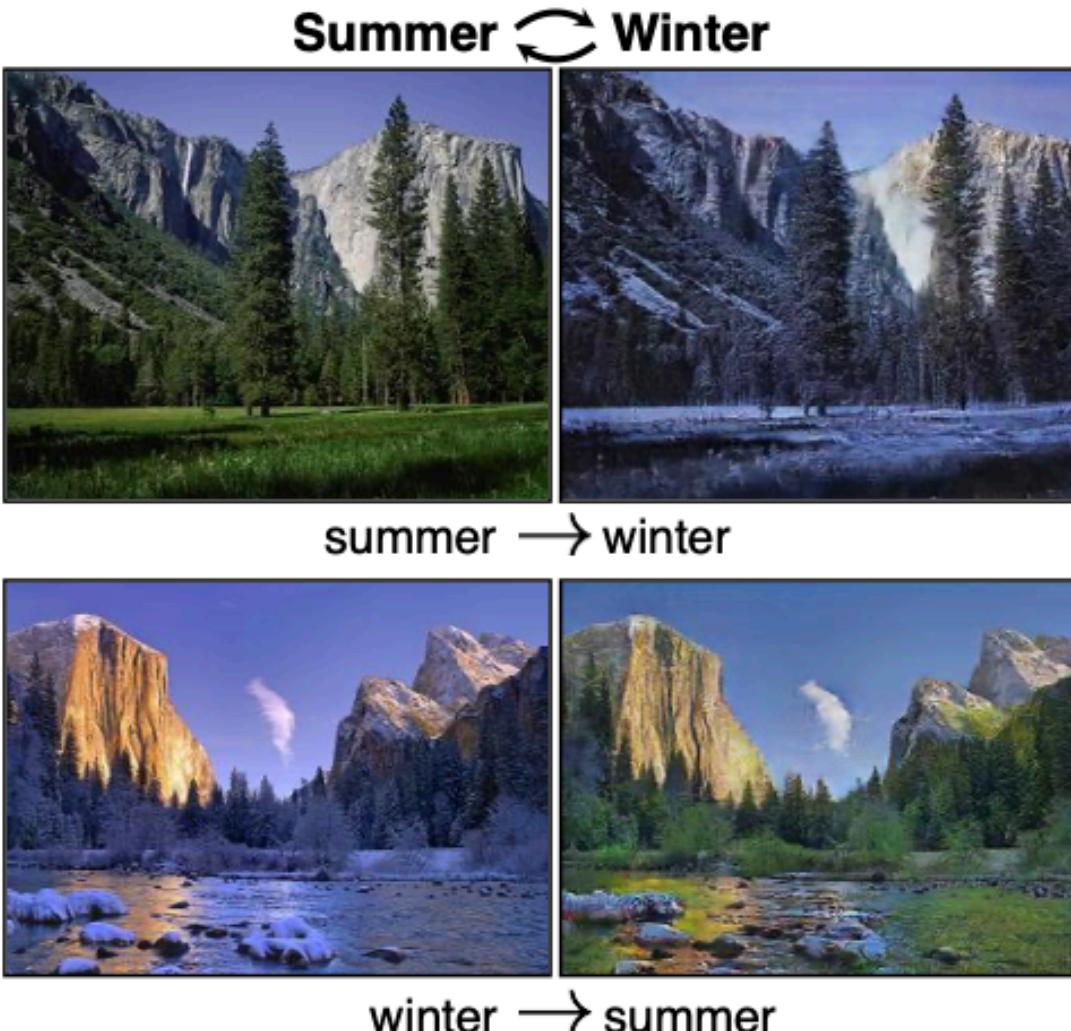
D_Y → aims to discriminate between $\{y\}$ and $\{G(x)\}$

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))]$$

$$\min_G \max_{D_Y} \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y)$$

$$\min_F \max_{D_X} \mathcal{L}_{\text{GAN}}(F, D_X, Y, X).$$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].$$



$$\begin{aligned} \mathcal{L}(G, F, D_X, D_Y) &= \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ &\quad + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &\quad + \lambda \mathcal{L}_{\text{cyc}}(G, F), \end{aligned}$$

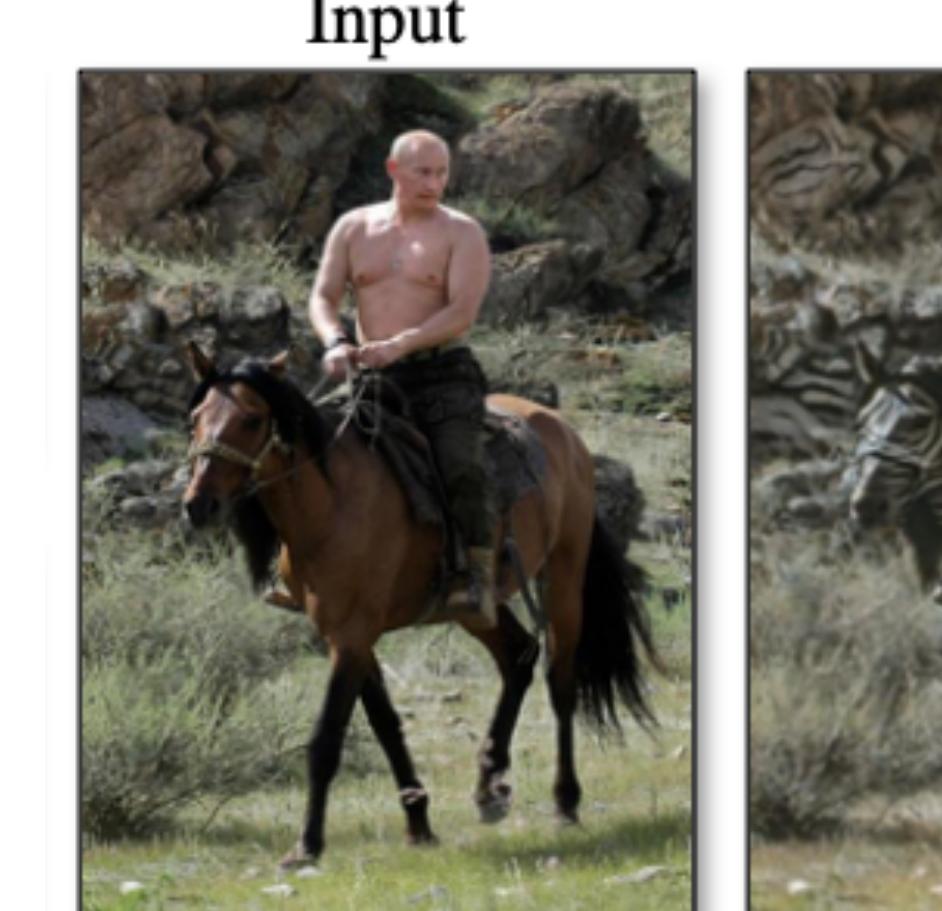
$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y).$$

Least-squares Loss

$$\min_G \mathbb{E}_{x \sim p_{\text{data}}(x)}[(D(G(x)) - 1)^2]$$

$$\min_D \mathbb{E}_{y \sim p_{\text{data}}(y)}[(D(y) - 1)^2] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[D(G(x))^2]$$

update the discriminators using a history of generated images



horse → zebra



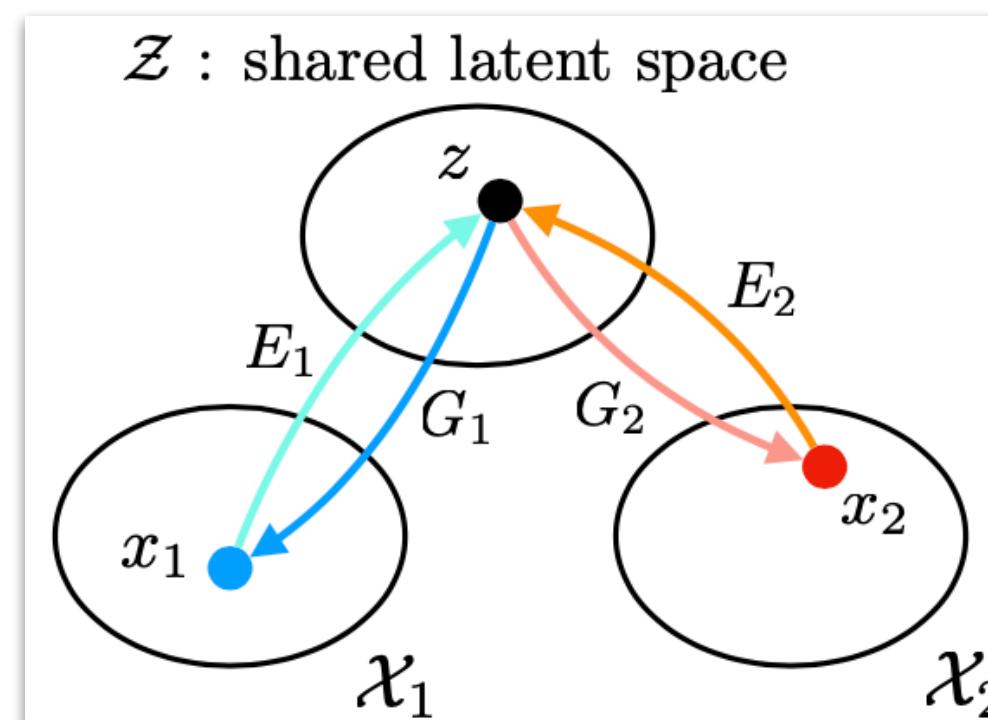
Boulder

Unsupervised Image-to-Image Translation Networks

UNIT: UNsupervised Image-to-image Translation Networks

Goal: To learn a joint distribution of images in different domains by using images from the marginal distributions in individual domains.
Since an infinite set of possible joint distributions can yield the given marginal distributions, we could infer nothing about the joint distribution from the marginal samples without additional assumptions.

Shared Latent Space Assumption



$$z = E_1(x_1) = E_2(x_2)$$

$$x_1 = G_1(z) \quad x_2 = G_2(z)$$

Shared latent space constraint implies cycle-consistency!

$$F^{1 \rightarrow 2} := G_2 \circ E_1 \quad F^{2 \rightarrow 1} := G_1 \circ E_2$$

$$x_1 = F^{2 \rightarrow 1}(F^{1 \rightarrow 2}(x_1))$$

$$x_2 = F^{1 \rightarrow 2}(F^{2 \rightarrow 1}(x_2))$$

$\{E_1, E_2\} \rightarrow$ encoders, $\{G_1, G_2\} \rightarrow$ generators, $\{D_1, D_2\} \rightarrow$ discriminators

Variational Autoencoders (VAEs)

$\{E_1, G_1\} \rightarrow$ VAE for the \mathcal{X}_1 domain (VAE_1)

$$q_1(z_1|x_1) := \mathcal{N}(z_1|E_1(x_1), I)$$

$x_1^{1 \rightarrow 1} = G_1(\underbrace{z_1 \sim q_1(z_1|x_1)}_{\text{re-parametrization trick}}) \rightarrow$ reconstructed image

$$\text{re-parametrization trick } z_1 = E_1(x_1) + \eta, \eta \sim \mathcal{N}(0, I)$$

$$\mathcal{L}_{\text{VAE}_1}(E_1, G_1) = \lambda_1 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_1 \sim q_1(z_1|x_1)} [\log p_{G_1}(x_1|z_1)]$$

$$p_\eta(z) = \mathcal{N}(z|0, I)$$

$p_{G_1} \rightarrow$ Laplace distribution (equivalent to minimizing the absolute distance)

Weight Sharing

$$E_1 = E_H \circ E_{L,1}, E_2 = E_H \circ E_{L,2}$$

$H, L \rightarrow$ High, Low level features

$$G_1 = G_{L,1} \circ G_H, G_2 = G_{L,2} \circ G_H$$

Note that the weight-sharing constraint alone does not guarantee that corresponding images in two domains will have the same latent code.

Generative Adversarial Networks (GANs)

$$\{G_1, D_1\} \rightarrow \text{GAN}_1$$

G_1 can generate two types of images:

- reconstruction stream (supervised training): $x_1^{1 \rightarrow 1} = G_1(z_1 \sim q_1(z_1|x_1))$
- translation stream (adversarial training): $x_2^{2 \rightarrow 1} = G_1(z_2 \sim q_2(z_2|x_2))$

$$\mathcal{L}_{\text{GAN}_1}(E_1, G_1, D_1) = \lambda_0 \mathbb{E}_{x_1 \sim P_{\mathcal{X}_1}} [\log D_1(x_1)] + \lambda_0 \mathbb{E}_{z_2 \sim q_2(z_2|x_2)} [\log(1 - D_1(G_1(z_2)))]$$

Cycle-consistency (CC)

$$\mathcal{L}_{\text{CC}_1}(E_1, G_1, E_2, G_2) = \lambda_3 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) + \lambda_3 \text{KL}(q_2(z_2|x_1^{1 \rightarrow 2}))||p_\eta(z)) - \lambda_4 \mathbb{E}_{z_2 \sim q_2(z_2|x_1^{1 \rightarrow 2})} [\log p_{G_1}(x_1|z_2)]$$

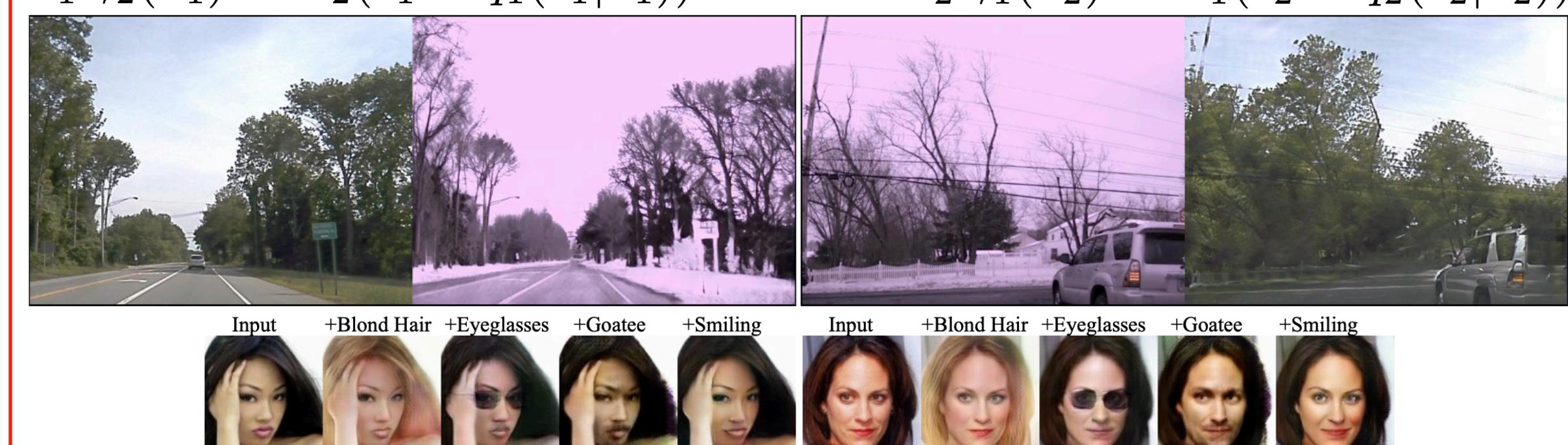
Learning

$$\min_{E_1, E_2, G_1, G_2} \max_{D_1, D_2} \mathcal{L}_{\text{VAE}_1}(E_1, G_1) + \mathcal{L}_{\text{GAN}_1}(E_1, G_1, D_1) + \mathcal{L}_{\text{CC}_1}(E_1, G_1, E_2, G_2)$$

Translation

$$F_{1 \rightarrow 2}(x_1) = G_2(z_1 \sim q_1(z_1|x_1))$$

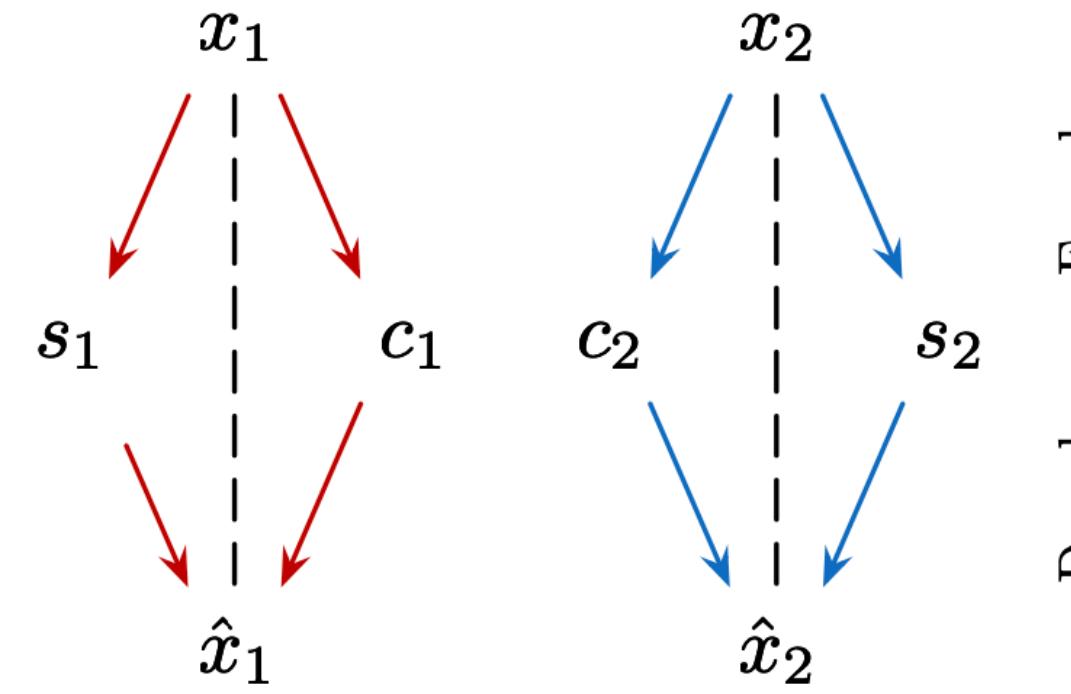
$$F_{2 \rightarrow 1}(x_2) = G_1(z_2 \sim q_2(z_2|x_2))$$



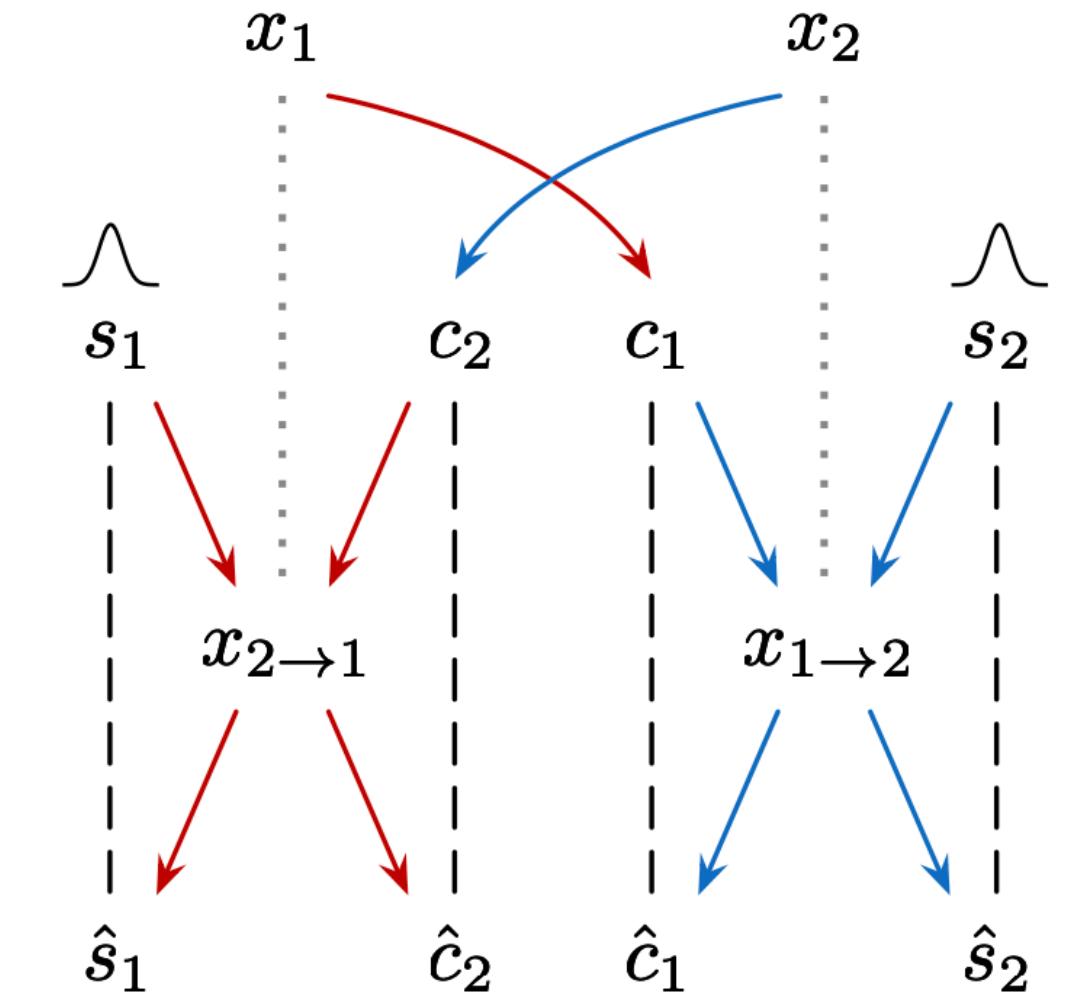


Boulder

Multimodal Unsupervised Image-to-Image Translation



(a) Within-domain reconstruction



(b) Cross-domain translation

E_i and $G_i \rightarrow$ encoder and decoder for each domain \mathcal{X}_i ($i = 1, 2$)

$(c_i, s_i) = (E_i^c(x_i), E_i^s(x_i)) = E_i(x_i) \rightarrow$ content and style codes

Translate image $x_1 \in \mathcal{X}_1$ to \mathcal{X}_2 :

$$c_1 = E_1^c(x_1)$$

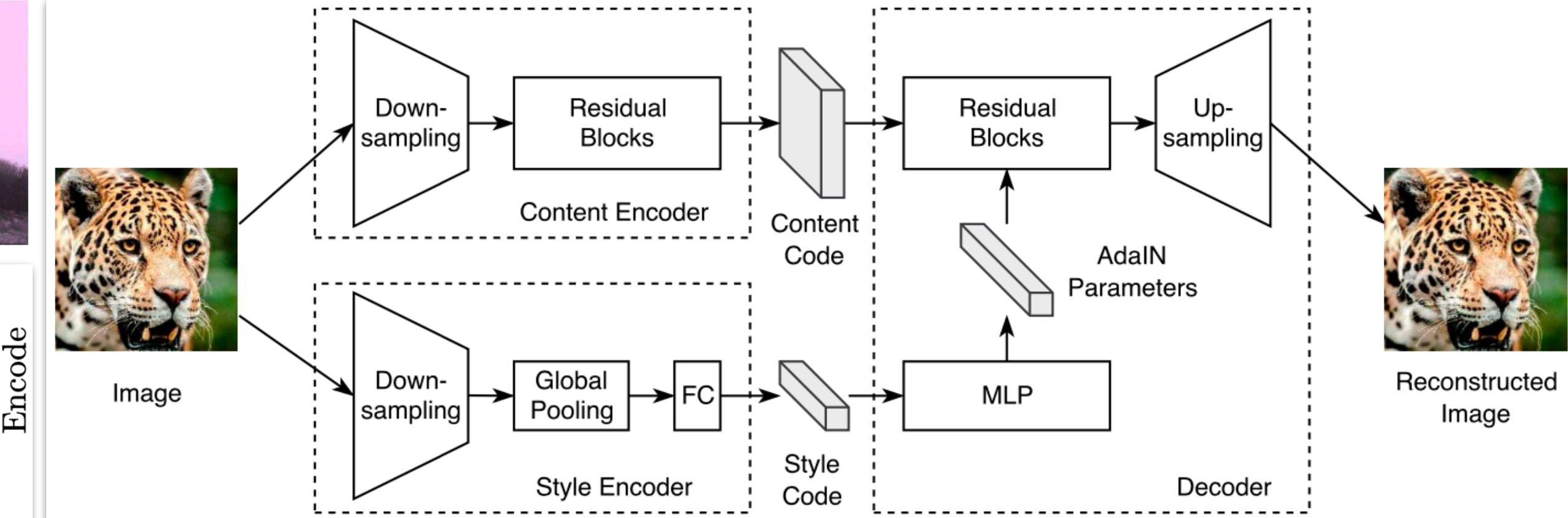
$$s_2 \sim q(s_2) = \mathcal{N}(0, I)$$

$$x_{1 \rightarrow 2} = G_2(c_1, s_2)$$

(Conditional) Inception Score

$$\text{CIS} = \mathbb{E}_{x_1 \sim p(x_1)} [\mathbb{E}_{x_{1 \rightarrow 2} \sim p(x_{2 \rightarrow 1} | x_1)} [\text{KL}(p(y_2 | x_{1 \rightarrow 2}) || p(y_2 | x_1))]]$$

$$\text{IS} = \mathbb{E}_{x_1 \sim p(x_1)} [\mathbb{E}_{x_{1 \rightarrow 2} \sim p(x_{2 \rightarrow 1} | x_1)} [\text{KL}(p(y_2 | x_{1 \rightarrow 2}) || p(y_2))]]$$



$$\text{AdaIN}(z, \gamma, \beta) = \gamma \left(\frac{z - \mu(z)}{\sigma(z)} \right) + \beta \rightarrow \text{Adaptive Instance Normalization}$$

γ and $\beta \rightarrow$ parameters generated by the MLP

Loss

$$\min_{E_1, E_2, G_1, G_2} \max_{D_1, D_2} \mathcal{L}(E_1, E_2, G_1, G_2, D_1, D_2) = \mathcal{L}_{\text{GAN}}^{x_1} + \mathcal{L}_{\text{GAN}}^{x_2} + \lambda_x (\mathcal{L}_{\text{recon}}^{x_1} + \mathcal{L}_{\text{recon}}^{x_2}) + \lambda_c (\mathcal{L}_{\text{recon}}^{c_1} + \mathcal{L}_{\text{recon}}^{c_2}) + \lambda_s (\mathcal{L}_{\text{recon}}^{s_1} + \mathcal{L}_{\text{recon}}^{s_2})$$

$$\mathcal{L}_{\text{recon}}^{x_1} = \mathbb{E}_{x_1 \sim p(x_1)} [||G_1(E_1^c(x_1), E_1^s(x_1)) - x_1||_1]$$

$$\mathcal{L}_{\text{recon}}^{c_1} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)} [||E_2^c(G_2(c_1, s_2)) - c_1||_1]$$

$$\mathcal{L}_{\text{recon}}^{s_2} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)} [||E_2^s(G_2(c_1, s_2)) - s_2||_1]$$

$p(c_1)$ is given by $c_1 = E_1^c(x_1)$ and $x_1 \sim p(x_1)$

$$\mathcal{L}_{\text{GAN}}^{x_2} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)} [\log(1 - D_2(G_2(c_1, s_2)))] + \mathbb{E}_{x_2 \sim p(x_2)} [\log D_2(x_2)]$$



Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network


[YouTube Video](#)


I^{SR} → high-resolution, super-resolved image

I^{LR} → low-resolution input image

I^{HR} → high-resolution counterpart of I^{LR}

$I^{HR} \triangleright$ Gaussian Filter \triangleright downsampling (downsampling_factor = r)

C → number of color channels

$I^{LR} \in \mathbb{R}^{W \times H \times C}$

$I^{SR}, I^{HR} \in \mathbb{R}^{rW \times rH \times C}$

evaluation metrics → $\begin{cases} \text{PSNR: peak signal-to-noise ratio} \\ \text{MOS: mean opinion score} \\ \text{SSIM: structural similarity} \end{cases}$

$G_{\theta_G} : I^{LR} \mapsto I^{HR}$ → generator

$\hat{\theta}^G = \arg \min_{\theta_G} \frac{1}{N} \sum_{n=1}^N \ell^{SR}(G_{\theta_G}(I_n^{LR}), I_n^{HR})$ → loss function

$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR} \sim p_{\text{train}}(I^{HR})} [\log D_{\theta_D}(I^{HR})] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})} [\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))]$

$$l^{SR} = \underbrace{l_X^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}}$$

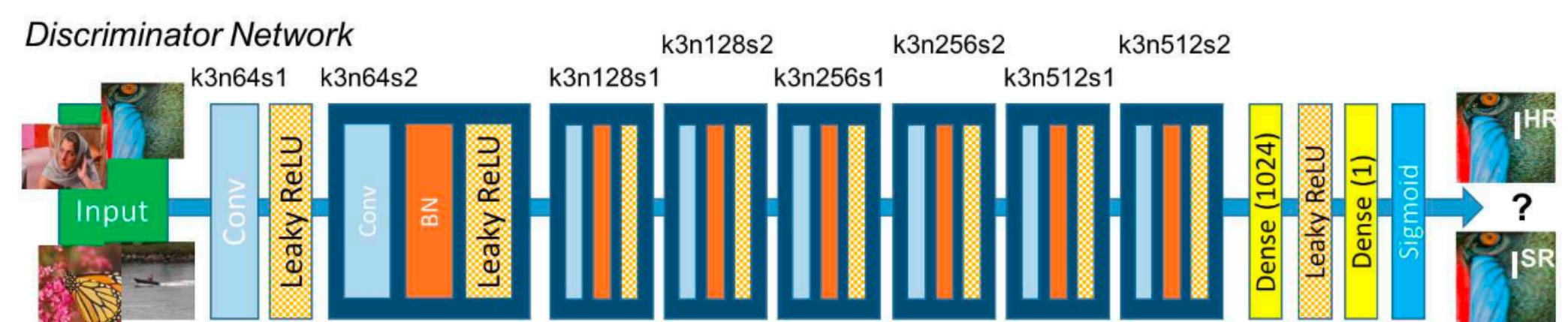
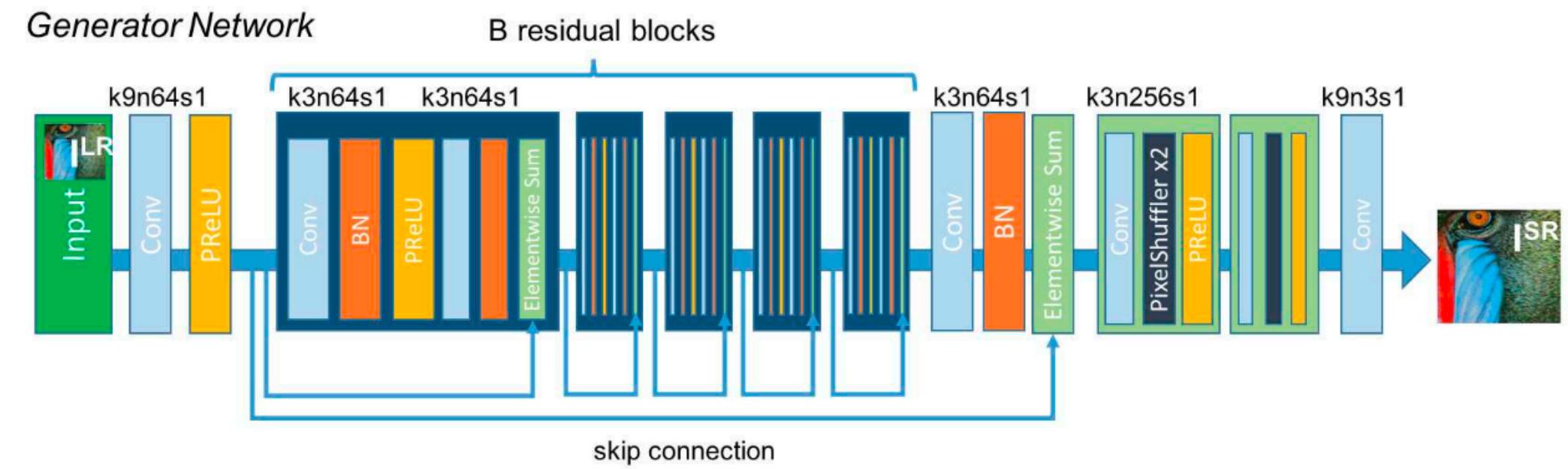
perceptual loss (for VGG based content losses)

$$l_{VGG/i.j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

$$l_{MSE}^{SR} = \frac{1}{r^2 W H} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y})^2$$

$$l_{Gen}^{SR} = \sum_{n=1}^N -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

$\phi_{i,j}$ → feature map obtained by the j -th convolution
(after activation) before the i -th maxpooling layer within the VGG19 network

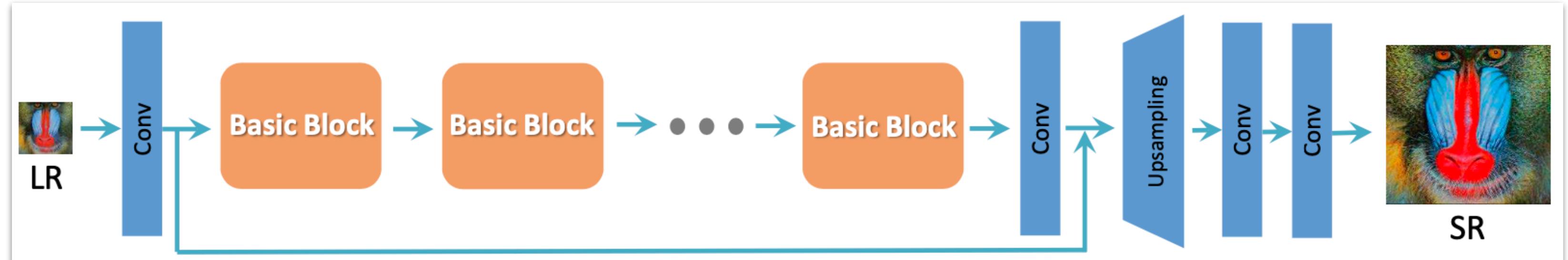




Boulder

ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks

Network Architecture



$D_{Ra} \rightarrow$ relativistic average discriminator (RaD)

$$D_{Ra}(x_r, x_f) = \sigma(C(x_r) - \mathbb{E}_{x_f}[C(x_f)])$$

all fake data in the mini-batch

$$L_D^{Ra} = -\mathbb{E}_{x_r}[\log(D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(1 - D_{Ra}(x_f, x_r))]$$

$$L_G^{Ra} = -\mathbb{E}_{x_r}[\log(1 - D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(D_{Ra}(x_f, x_r))]$$

$x_f = G(x_i)$ and x_i stands for the input LR image

Perceptual Loss

Features before activation rather than after activation!

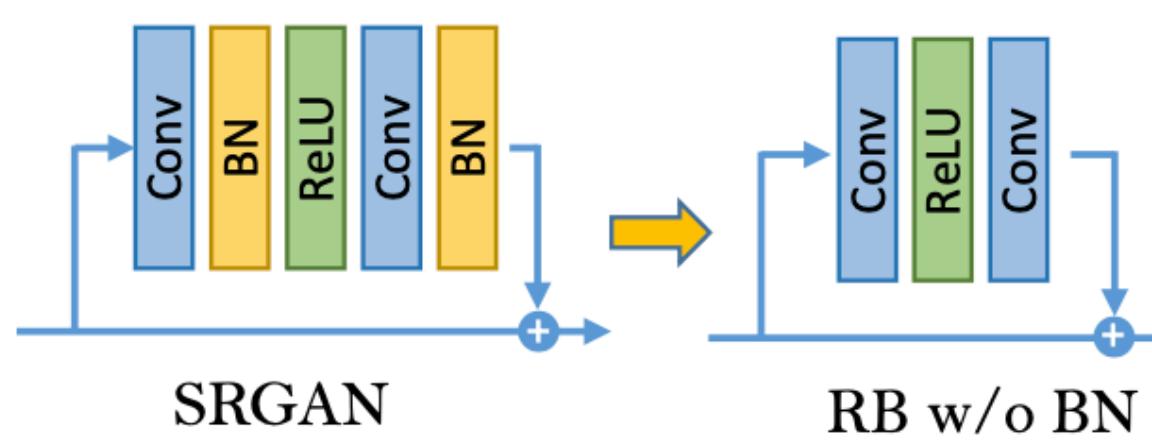
$$L_G = L_{percep} + \lambda L_G^{Ra} + \eta L_1$$

$$L_1 = \mathbb{E}_{x_i} \|G(x_i) - y\|_1 \rightarrow \text{content loss}$$

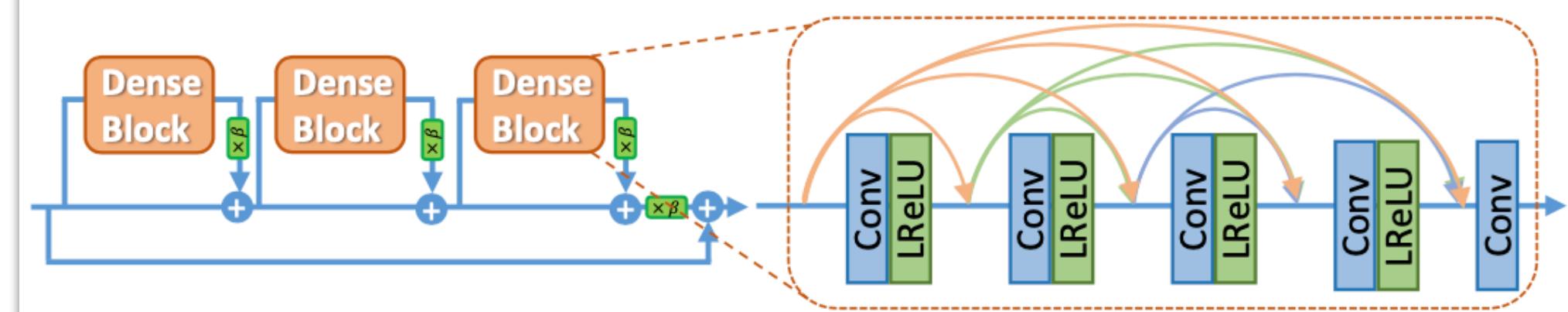
Network Interpolation

$$\theta_G^{\text{INTERP}} = (1 - \alpha) \theta_G^{\text{PSNR}} + \alpha \theta_G^{\text{GAN}}$$

Residual Block (RB)



Residual in Residual Dense Block (RRDB)



Relativistic Discriminator

$$D(x_r) = \sigma(C(\text{Real})) \rightarrow 1 \quad \text{Real?}$$

$$D_{Ra}(x_r, x_f) = \sigma(C(\text{Real}) - \mathbb{E}[C(\text{Fake})]) \rightarrow 1 \quad \text{More realistic than fake data?}$$

$$D(x_f) = \sigma(C(\text{Fake})) \rightarrow 0 \quad \text{Fake?}$$

$$D_{Ra}(x_f, x_r) = \sigma(C(\text{Fake}) - \mathbb{E}[C(\text{Real})]) \rightarrow 0 \quad \text{Less realistic than real data?}$$

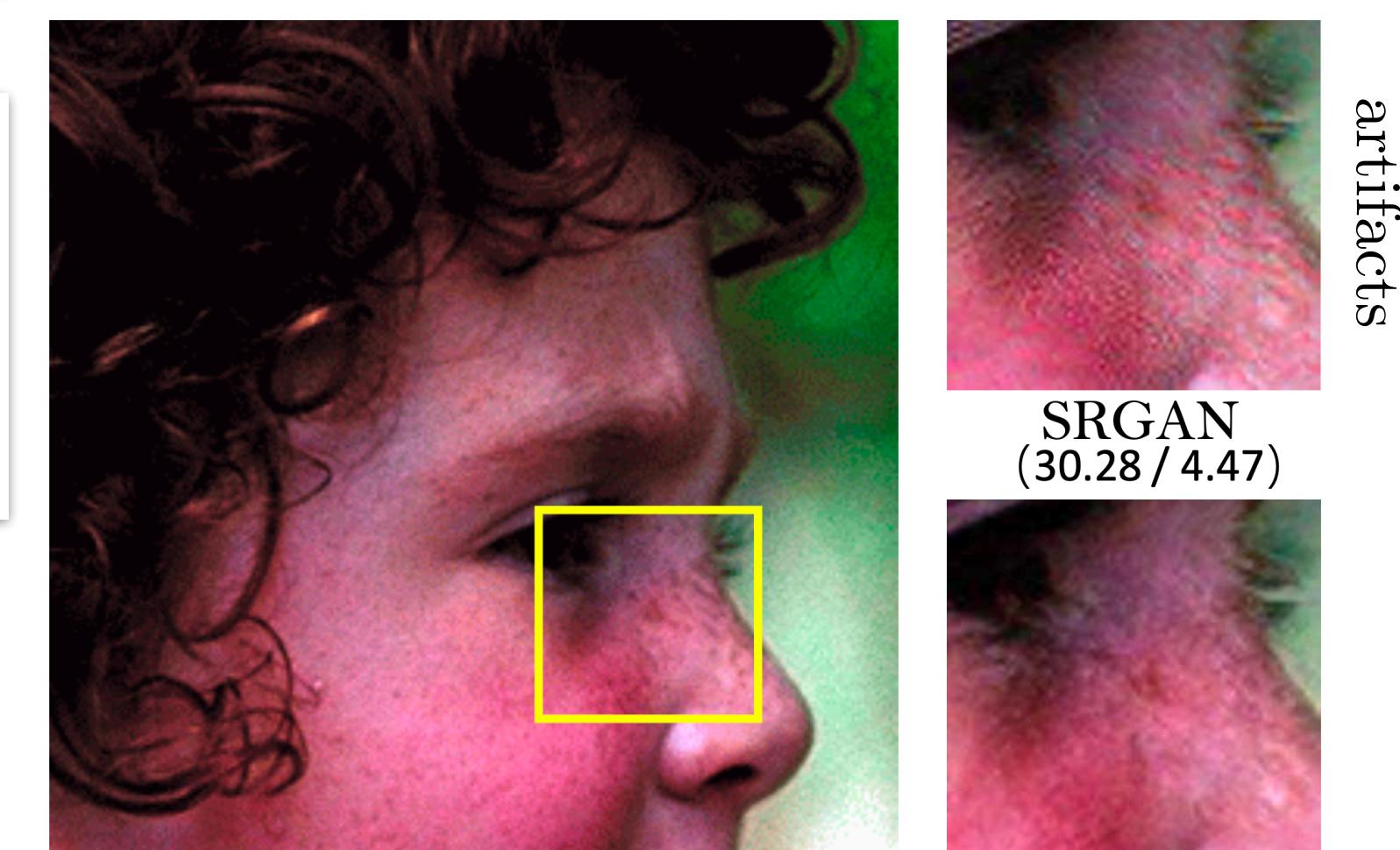
a) Standard GAN

b) Relativistic GAN

Judge “whether one image is more realistic than the other” rather than “whether one image is real or fake”

$D(x) = \sigma(C(x)) \rightarrow$ standard discriminator

$C(x) \rightarrow$ non-transformed discriminator output



face from Set14
(PSNR / Perceptual Index)



Boulder

High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs



[YouTube Video](#)

The pix2pix baseline

$G \rightarrow$ generator

$D \rightarrow$ discriminator

$G \rightarrow$ translate semantic label maps to realistic looking images

$D \rightarrow$ distinguish real images from the translated ones

$\{(s_i, x_i)\} \rightarrow$ training data

$s_i \rightarrow$ semantic label map

$x_i \rightarrow$ corresponding natural photo

$\min_G \max_D \mathcal{L}_{\text{GAN}}(G, D)$

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{(s, x)}[\log D(s, x)] + \mathbb{E}_s[\log(1 - D(s, G(s)))]$$

U-Net as the generator

patch-based fully convolutional network as the discriminator

256×256

Improving photorealism and resolution 2048×1024

- coarse-to-fine generator
- multi-scale discriminator
- improved adversarial loss

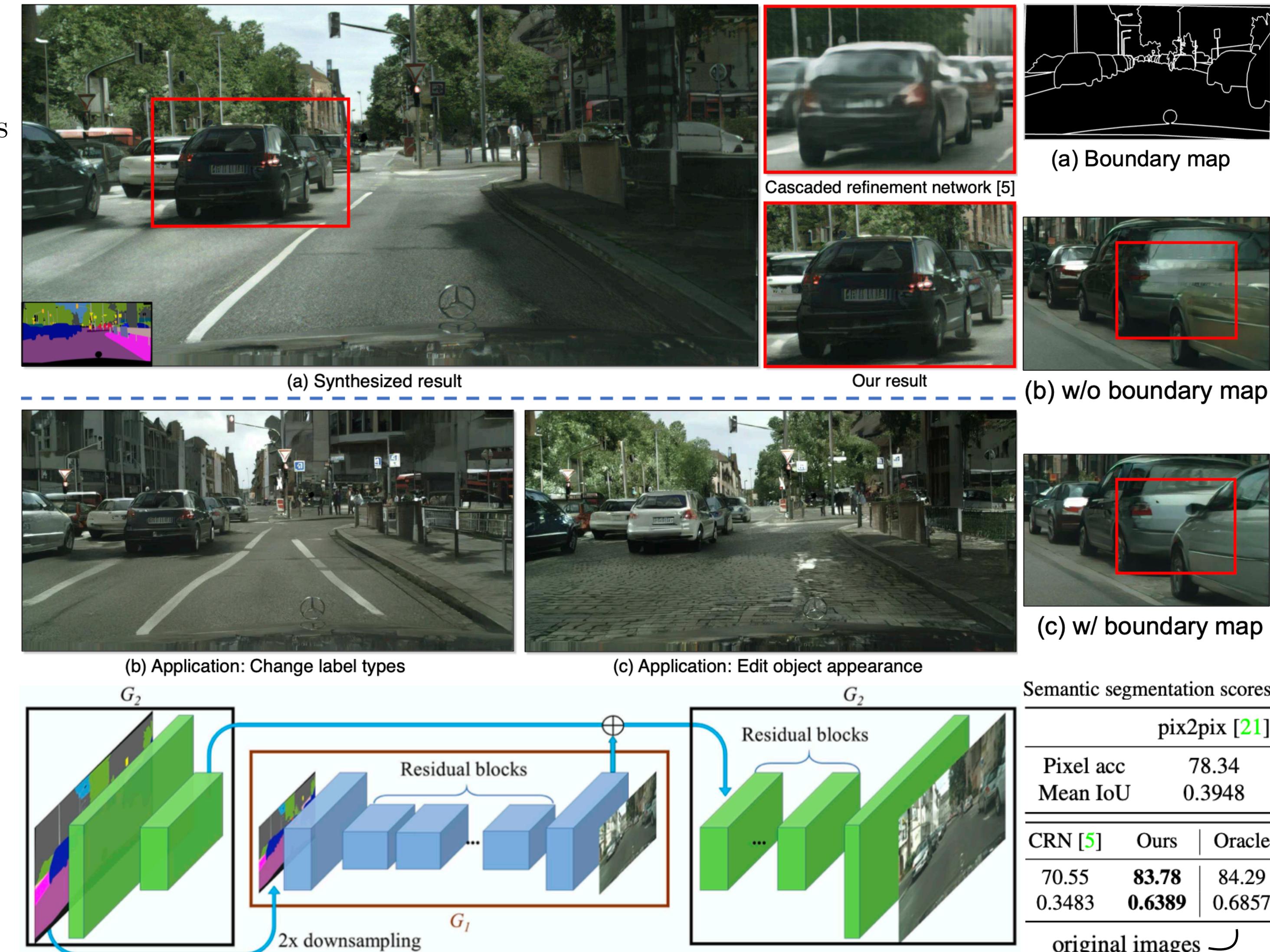
$$\min_G \left(\left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k) \right) + \lambda \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) \right)$$

$$\mathcal{L}_{\text{FM}}(G, D_k) = \mathbb{E}_{(\mathbf{s}, \mathbf{x})} \sum_{i=1}^T \frac{1}{N_i} \| | | D_k^{(i)}(\mathbf{s}, \mathbf{x}) - D_k^{(i)}(\mathbf{s}, G(\mathbf{s})) | |_1 \|$$

$D_k^{(i)} \rightarrow$ i -th layer feature extractor of discriminator D_k

$T \rightarrow$ total number of layers

$N_i \rightarrow$ number of elements in each layer

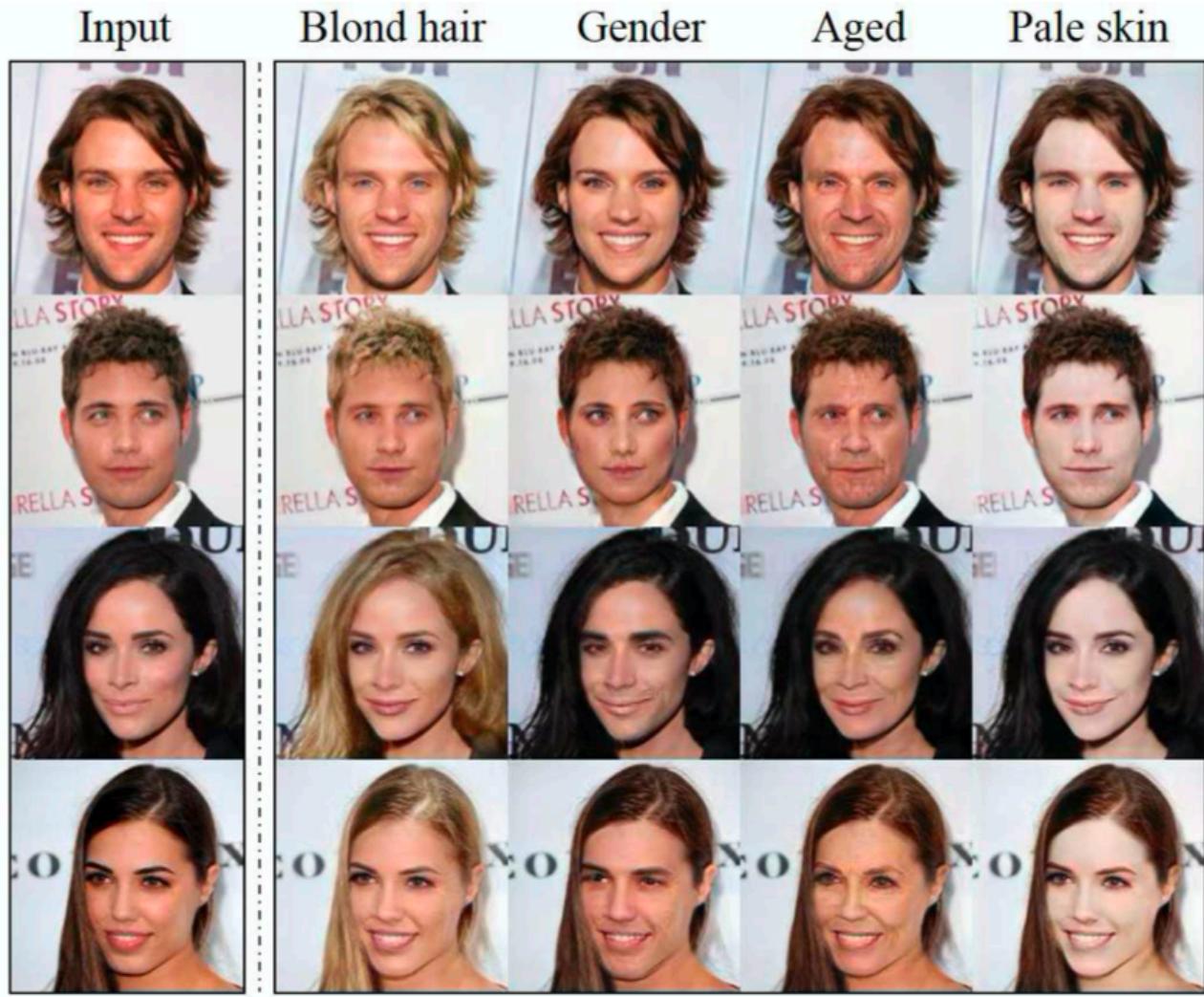




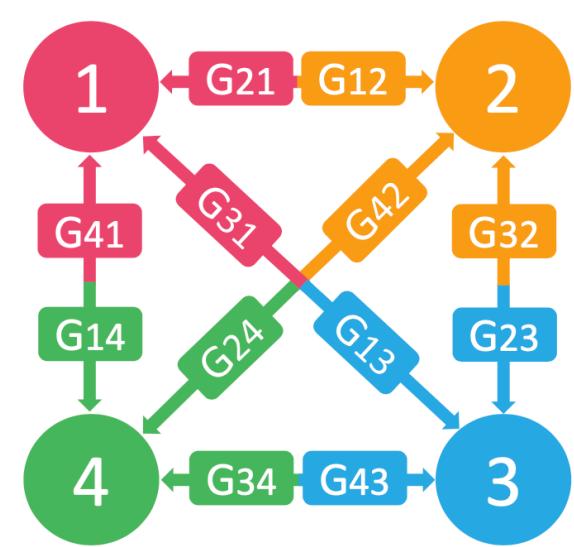
StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation



[YouTube Video](#)



(a) Cross-domain models



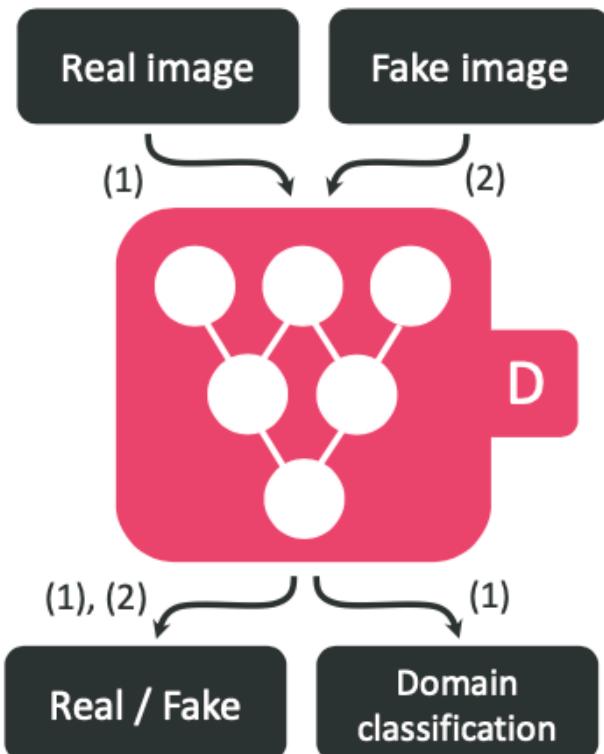
$$\begin{aligned} \mathcal{L}_{adv} &= \mathbb{E}_y[\log D_{src}(y)] + \mathbb{E}_{x,c}[\log(1 - D_{src}(G(x, c)))] \rightarrow \text{adversarial loss} \\ \mathcal{L}_{cls}^r &= \mathbb{E}_{y,c}[-\log D_{cls}(c|y)] \rightarrow \text{domain classification loss of real images} \\ \mathcal{L}_{cls}^f &= \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x, c))] \rightarrow \text{fake images} \\ \mathcal{L}_{rec} &= \mathbb{E}_{x,c,c'}[\|x - G(G(x, c), c')\|_1] \rightarrow \text{reconstruction loss} \\ \lambda_{cls} &= 1 \& \lambda_{rec} = 10 \end{aligned}$$

(b) StarGAN

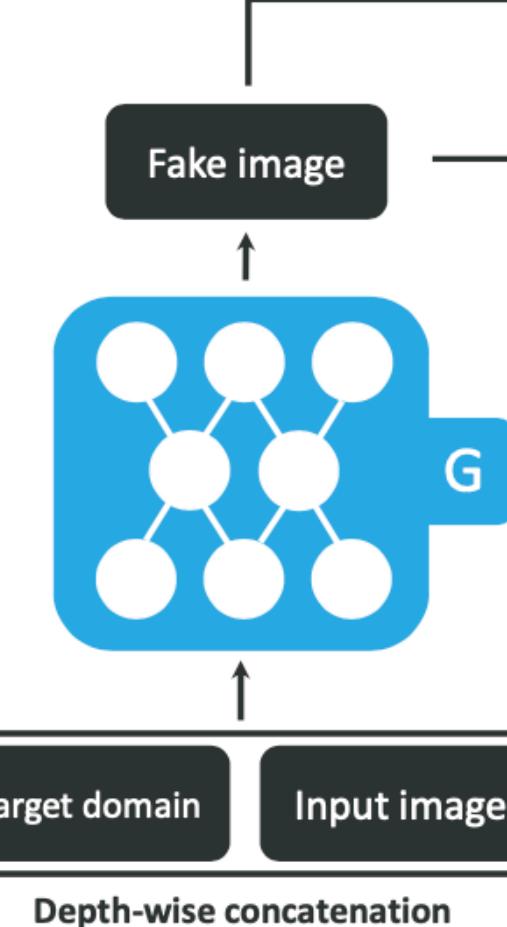


$G \rightarrow$ single generator
 $x \rightarrow$ input image
 $y \rightarrow$ output image
 $c \rightarrow$ target domain label
 $G(x, c) \rightarrow$ generated image
 $D : x \mapsto \{D_{src}(x), D_{cls}(x)\}$
 $\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^r$
 $\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^f + \lambda_{rec}\mathcal{L}_{rec}$

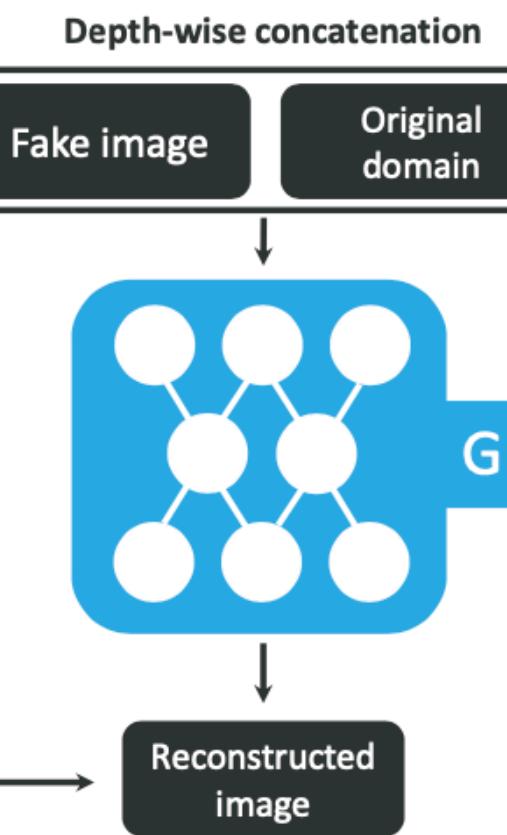
(a) Training the discriminator



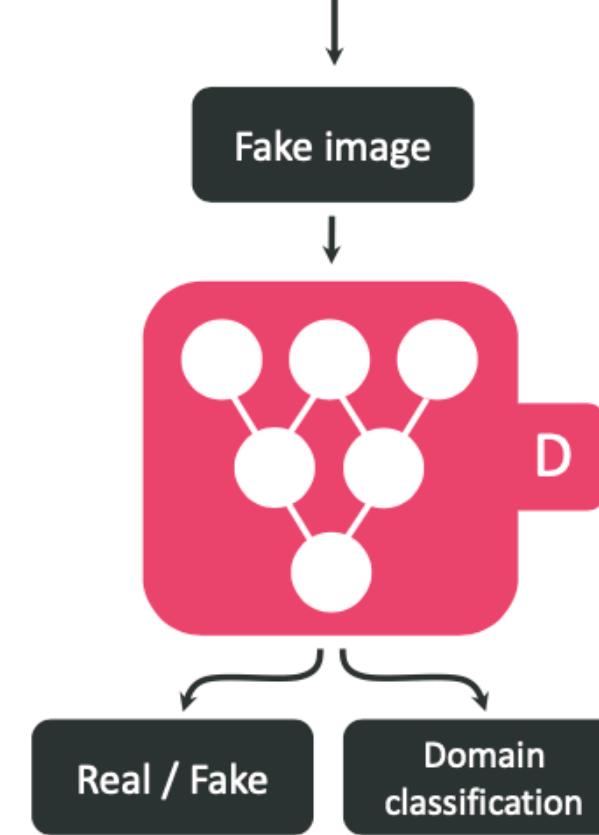
(b) Original-to-target domain



(c) Target-to-original domain



(d) Fooling the discriminator



Training with multiple datasets

$n \rightarrow$ number of datasets (e.g., $n = 2$)

CelebA and RaFD

$m \rightarrow$ mask vector (n-dimensional one-hot vector)

$\tilde{c} := [c_1, \dots, c_n, m] \rightarrow$ concatenate

$c_i \rightarrow$ vector for labels of the i -th dataset
assign zero values to the remaining $n - 1$ unknown labels

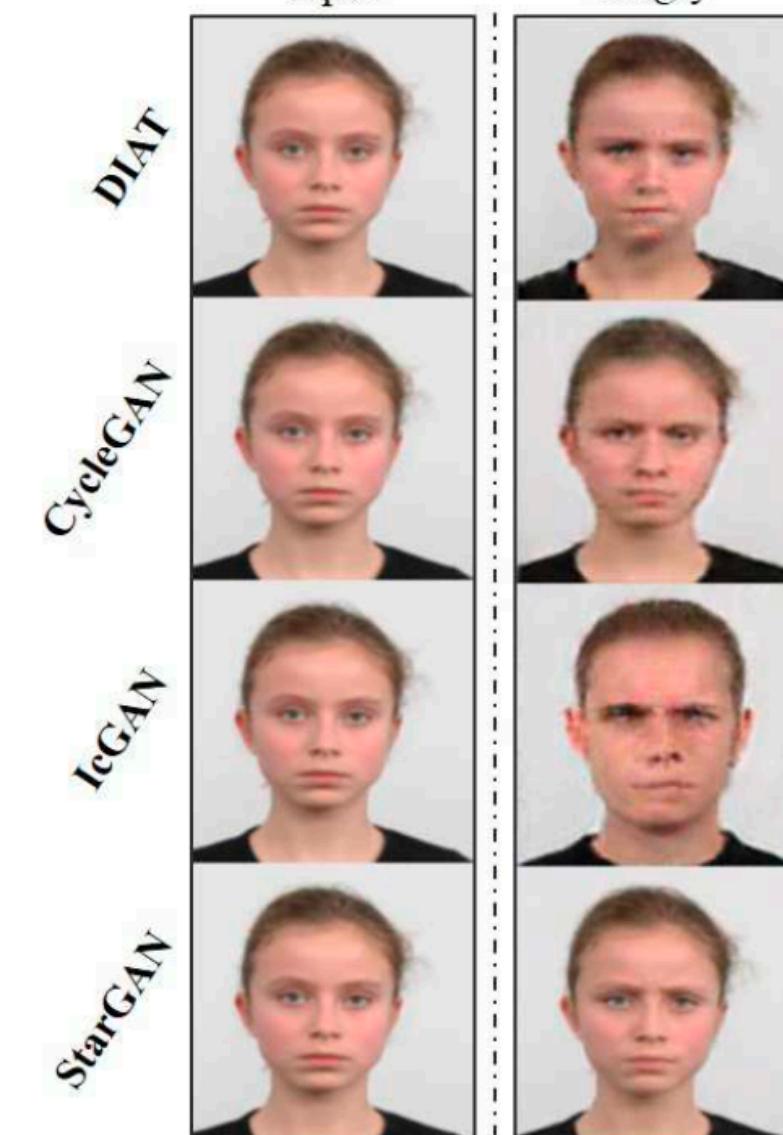
Improved GAN training

$$\mathcal{L}_{adv} = \mathbb{E}_y[D_{src}(y)] - \mathbb{E}_{x,c}[D_{src}(G(x, c))]$$

$$- \lambda_{gp} \mathbb{E}_{\hat{y}}[(\|\nabla_{\hat{y}} D_{src}(\hat{y})\|_2 - 1)^2]$$

$\hat{y} \rightarrow$ sampled uniformly along a straight line between y and $G(x, c)$

$$\lambda_{gp} = 10 \rightarrow \text{gradient penalty}$$



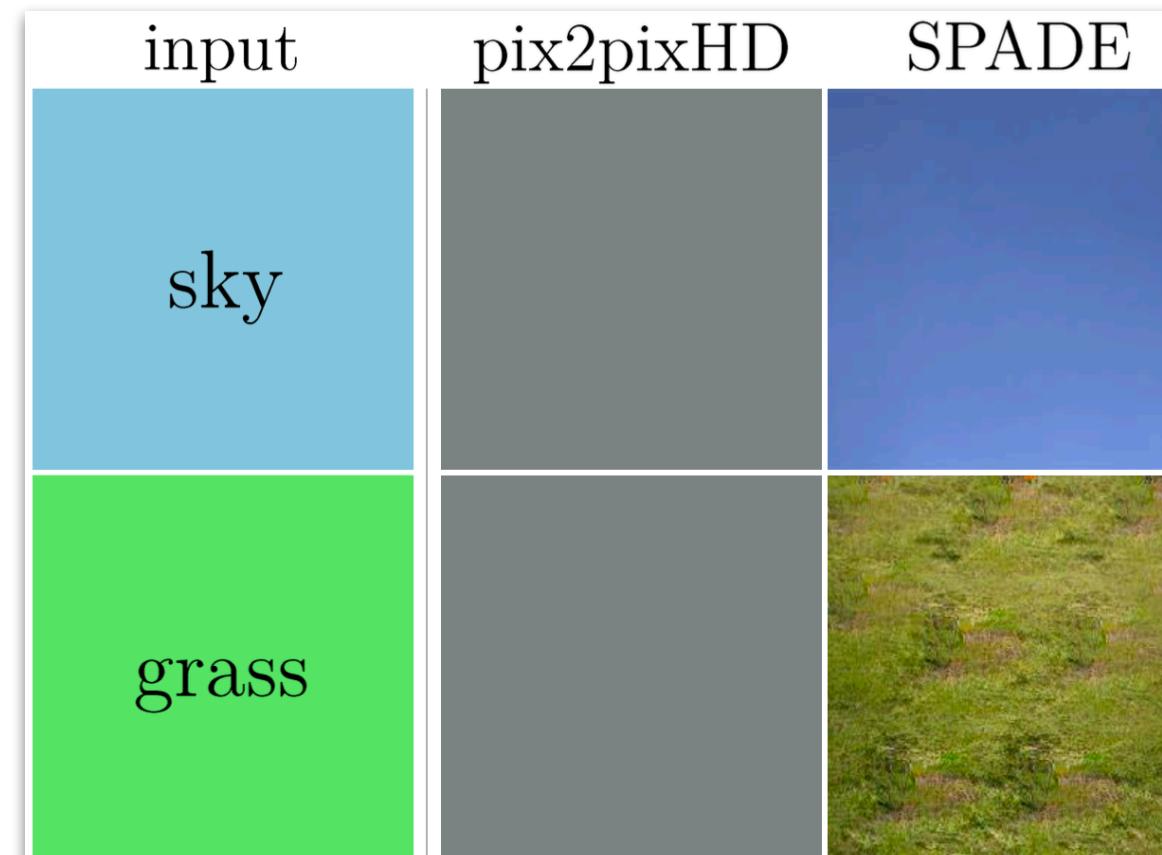


Boulder

Semantic Image Synthesis with Spatially-Adaptive Normalization



SPatially-Adaptive (DE)normalization (SPADE)
Normalization layers tend to “wash away” semantic information!



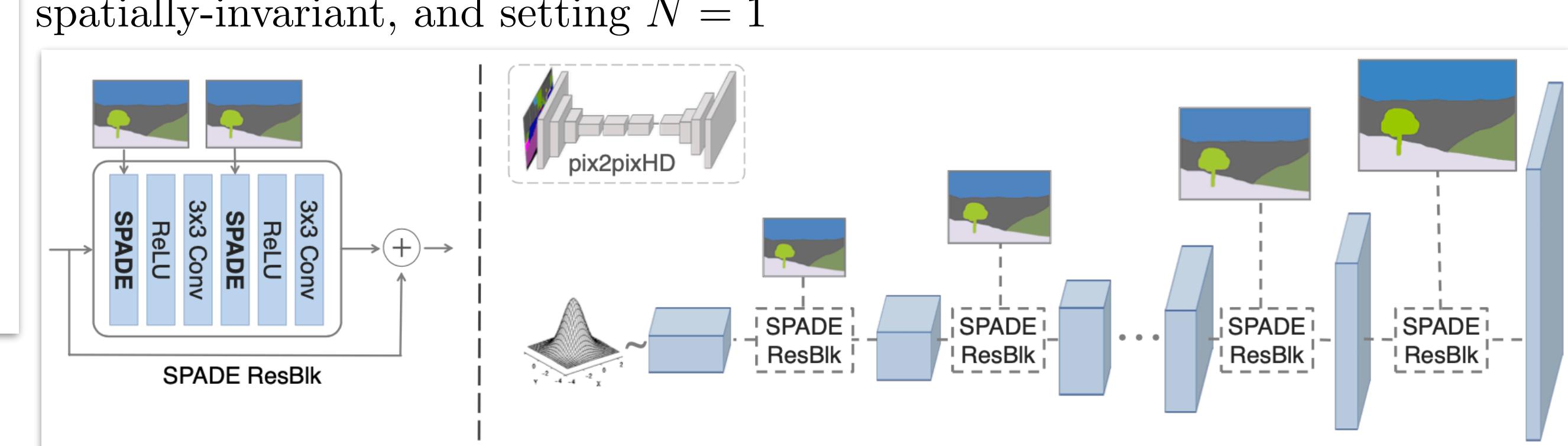
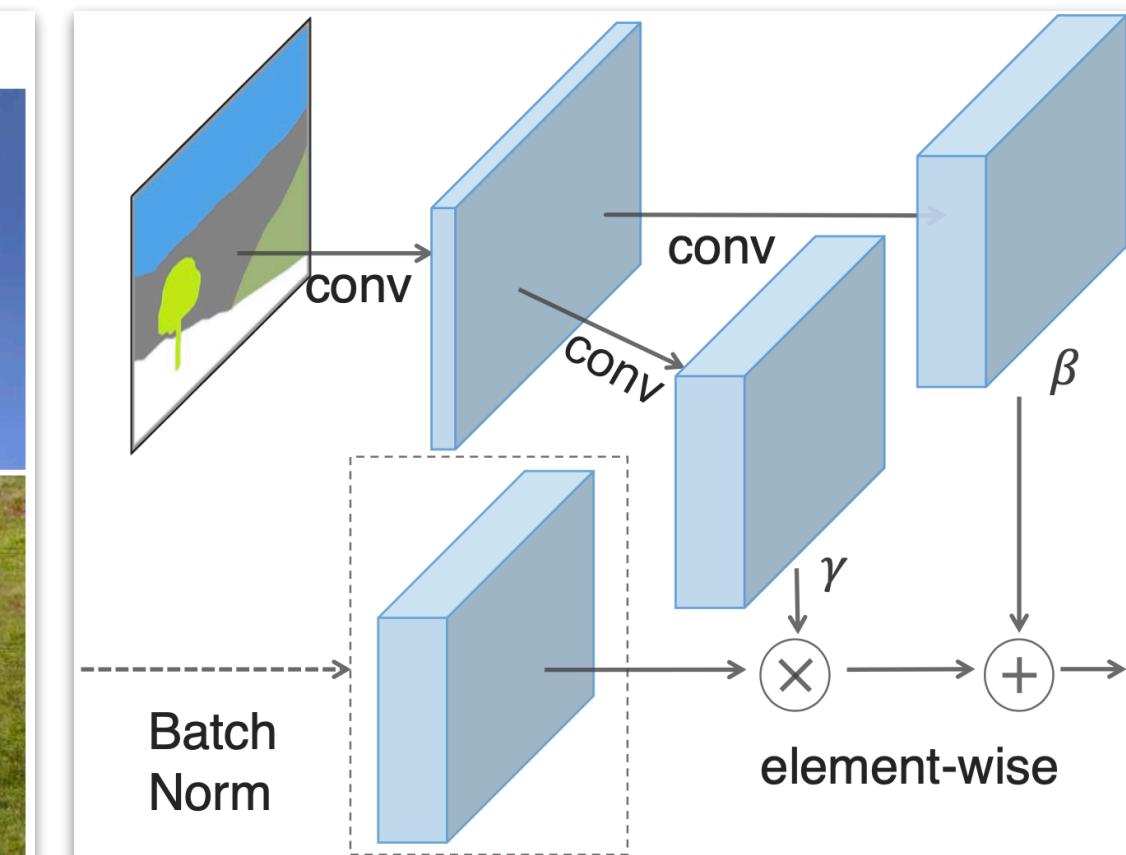
$\mathbf{m} \in \mathbb{L}^{H \times W} \rightarrow$ semantic segmentation mask

$\mathbb{L} \rightarrow$ set of integers denoting the semantic labels

$\mathbf{h}^i \in \mathbb{R}^{N \times C^i \times H^i \times W^i} \rightarrow$ activations of the i -th layer of a deep convolutional network for a batch of N samples

$\gamma_{c,y,x}^i(\mathbf{m}) \frac{h_{n,c,y,x}^i - \mu_c^i}{\sigma_c^i} + \beta_{c,y,x}^i(\mathbf{m}) \rightarrow$ activation value at site ($n \in N, c \in C^i, y \in H^i, x \in W^i$)

$h_{n,c,y,x}^i \rightarrow$ activation at the site before normalization



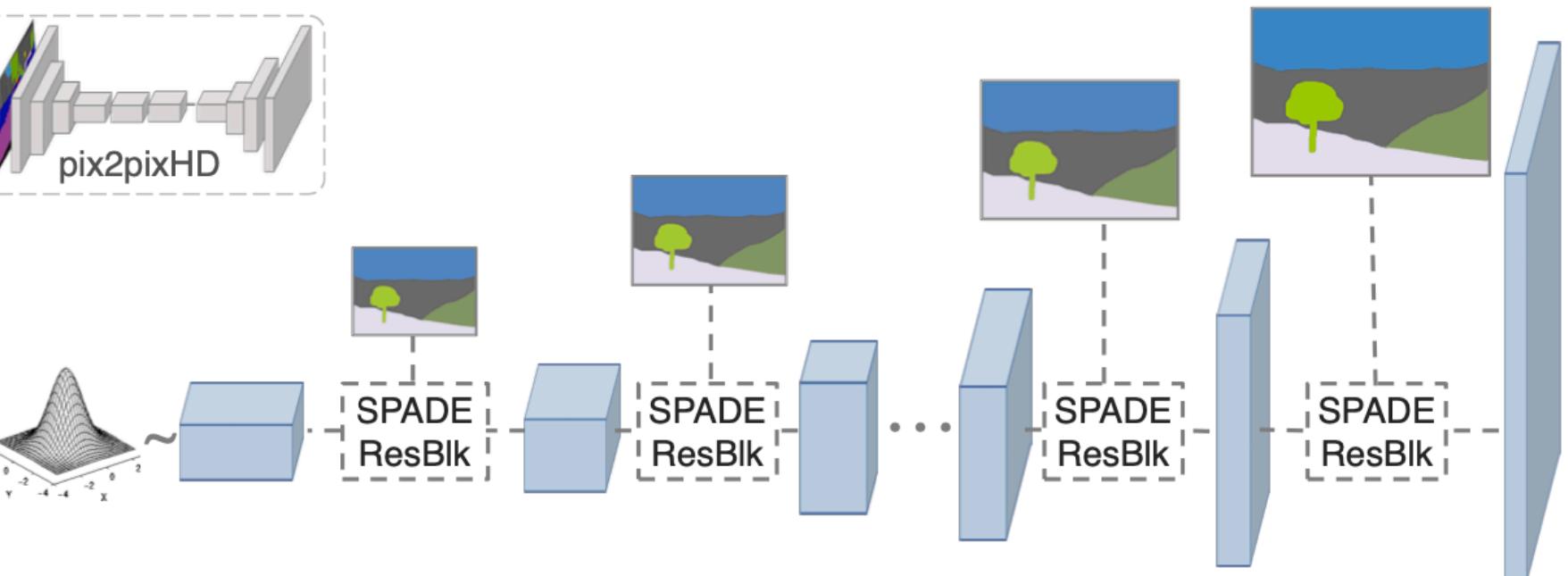
$$\mu_c^i = \frac{1}{NH^iW^i} \sum_{n,y,x} h_{n,c,y,x}^i$$

$$\sigma_c^i = \sqrt{\frac{1}{NH^iW^i} \sum_{n,y,x} ((h_{n,c,y,x}^i)^2 - (\mu_c^i)^2)}$$

$\gamma_{c,y,x}^i(\mathbf{m})$ and $\beta_{c,y,x}^i(\mathbf{m}) \rightarrow$ learned modulation parameters of the normalization layer (functions of \mathbf{m}, y, x)

Conditional BatchNorm: Replacing the segmentation mask \mathbf{m} with the image class label and making the modulation parameters spatially-invariant

AdaIN: Replacing \mathbf{m} with a real image, making the modulation parameters spatially-invariant, and setting $N = 1$



Same multi-scale discriminator and loss function used in pix2pixHD except replace the least squared loss term with the hinge loss term

Multi-modal synthesis

style \triangleright encoder \rightarrow random vector

For training, add a KL-Divergence loss term.



Boulder



Questions?

[YouTube Playlist](#)
