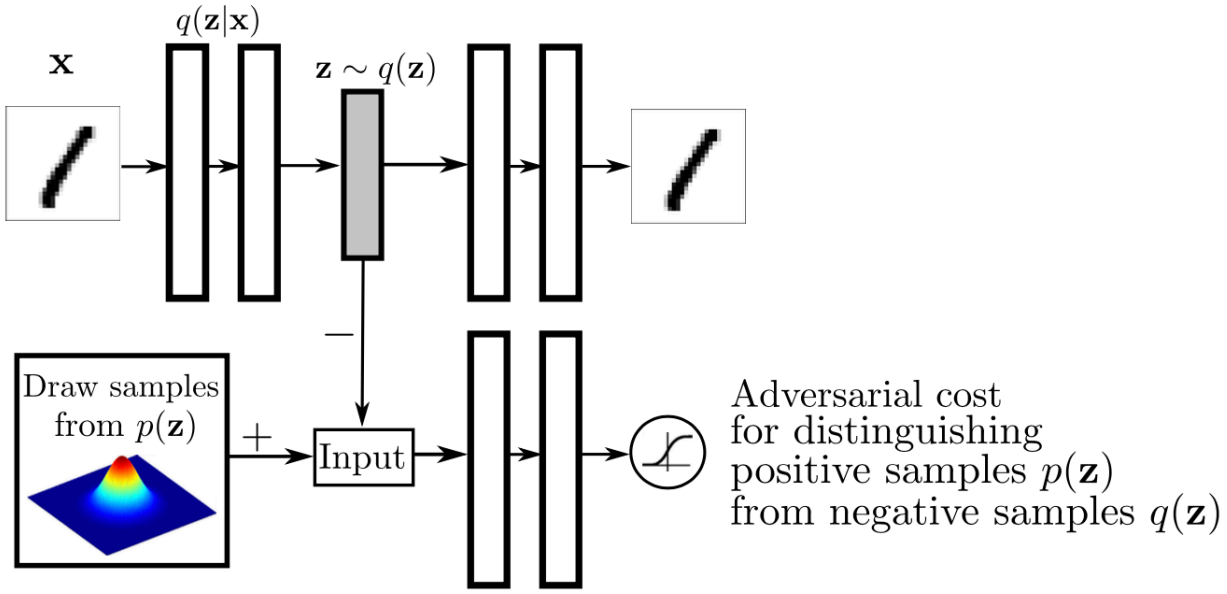
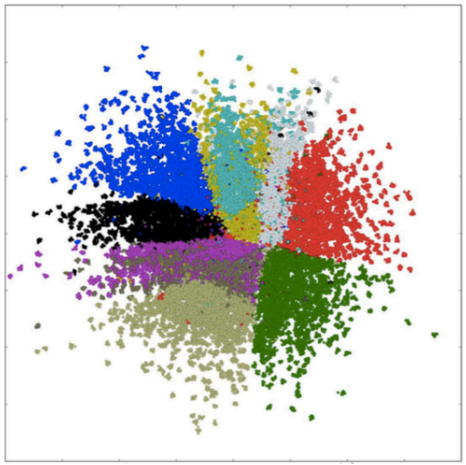
Part II

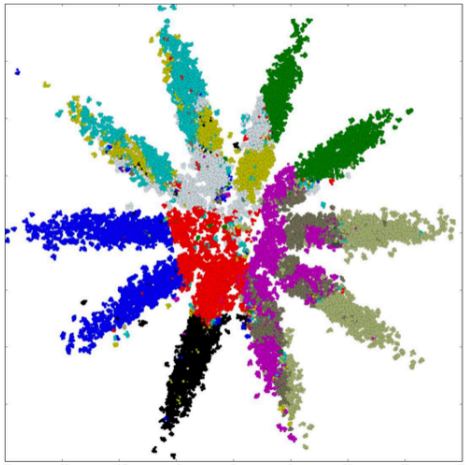
$$q(\mathbf{z}) = \int_{\mathbf{x}} q(\mathbf{z}|\mathbf{x}) p_{\text{data}}(\mathbf{x}) d\mathbf{x}$$



Code Space of MNIST:



Gaussian Prior



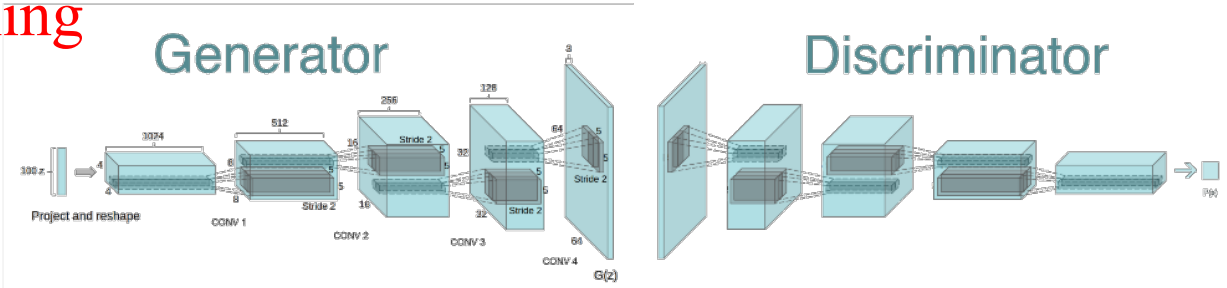
Mixture of Gaussians

Makhzani et al., 2015

Wasserstein GAN

Difficulty of Generative Adversarial Networks

- The gradient vanishing



- The full objective

$$J^{(D)} = -E_{x\sim P_r}[\log D(x)] - E_{x\sim P_g}[\log(1 - D(x))]$$

$$J^{(G)} = E_{x\sim P_g}[\log(1 - D(x))]$$

- At the very early phase of training, D is very easy to be confident in detecting G, so D will output almost always 0

In GAN, better discriminator leads to worse vanishing gradient in its generator!

Proof Sketch: [Martin Arjovsky et al., Wasserstein GAN, 2017]

- Minimizing generator yields minimizing the JS divergence when the discriminator is optimal:

$$J^{(D)} = -E_{x\sim P_r}[\log D(x)] - E_{x\sim P_g}[\log(1 - D(x))]$$

- The optimal $\boldsymbol{D}$ for any P_r and P_g is always: $-P_r(x)\log D(x) - P_g(x)\log[1 - D(x)]$

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)} \qquad -\frac{P_r(x)}{D(x)} + \frac{P_g(x)}{1 - D(x)} = 0$$

- The generative loss (by adding a term independent of P_g) is:

$$J^{(G)} = E_{x\sim P_g}[\log(1 - D(x))]$$

Plug D^* into $J^{(G)}$:

Jensen–Shannon divergence

$$J^{(D^*)} = 2JS(P_r||P_g) - 2\log 2$$

So, when $\boldsymbol{D}$ is ***optimal***, minimizing the loss is equal to minimizing the *JS divergence*.

JS divergence:

$$\mathrm{JS}(p \parallel q) = \frac{1}{2}\,\mathrm{KL}(p \parallel m) + \frac{1}{2}\,\mathrm{KL}(q \parallel m)$$

where

$$m(x) = \frac{1}{2}\,(p(x) + q(x))$$

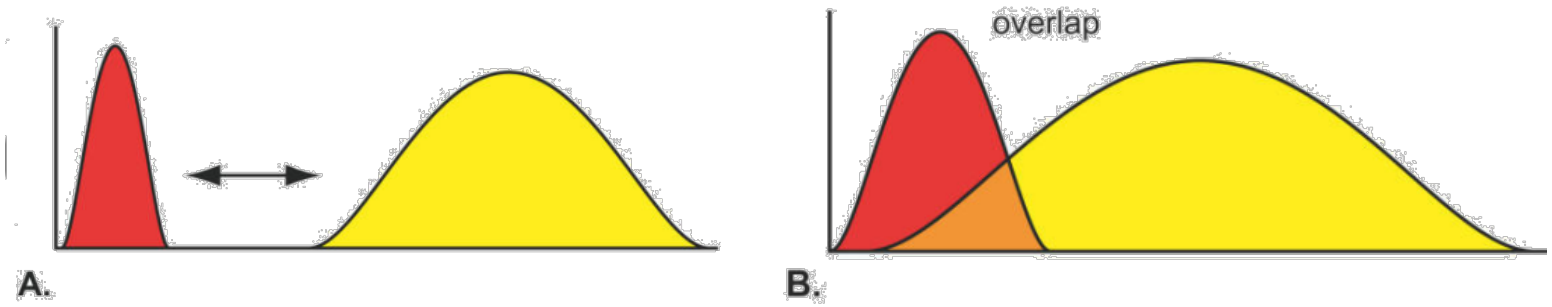
$$\mathrm{KL}(p \parallel q) = -\sum_{x} p(x) \log\left(\frac{q(x)}{p(x)}\right)$$

Proof Sketch:

$$J^{(D^*)} = 2JS(P_r||P_g) - 2\log 2$$

- If the supports (underlying low-dimension manifolds) of P_r and P_g (almost) have no overlap, then $JS(P_r||P_g) = \log 2$, and thus the gradient of $J^{(G)}$ w.r.t. P_g vanishes.

- The probability that the support of P_r and P_g have almost zero overlap is 1.



[Martin Arjovsky et al., Wasserstein GAN, 2017]

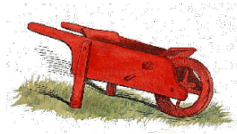
Preliminaries: distance measures for distributions

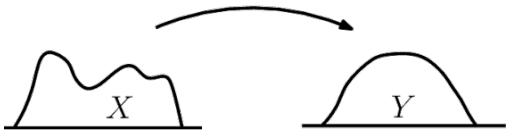
- JS

$$JS(P||Q) = \frac{1}{2}KL(P||\frac{P+Q}{2}) + \frac{1}{2}KL(Q||\frac{P+Q}{2})$$

- Wasserstein

$$W(P||Q) = \inf_{\gamma \in \Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$$





the set of all joint γ distributions whose marginals are P and Q, respectively.

γ indicates a plan to transport "mass" from x to y, when performing P into Q.

The Wasserstein (or Earth-Mover) distance is then the "cost" of the **optimal** transport plan

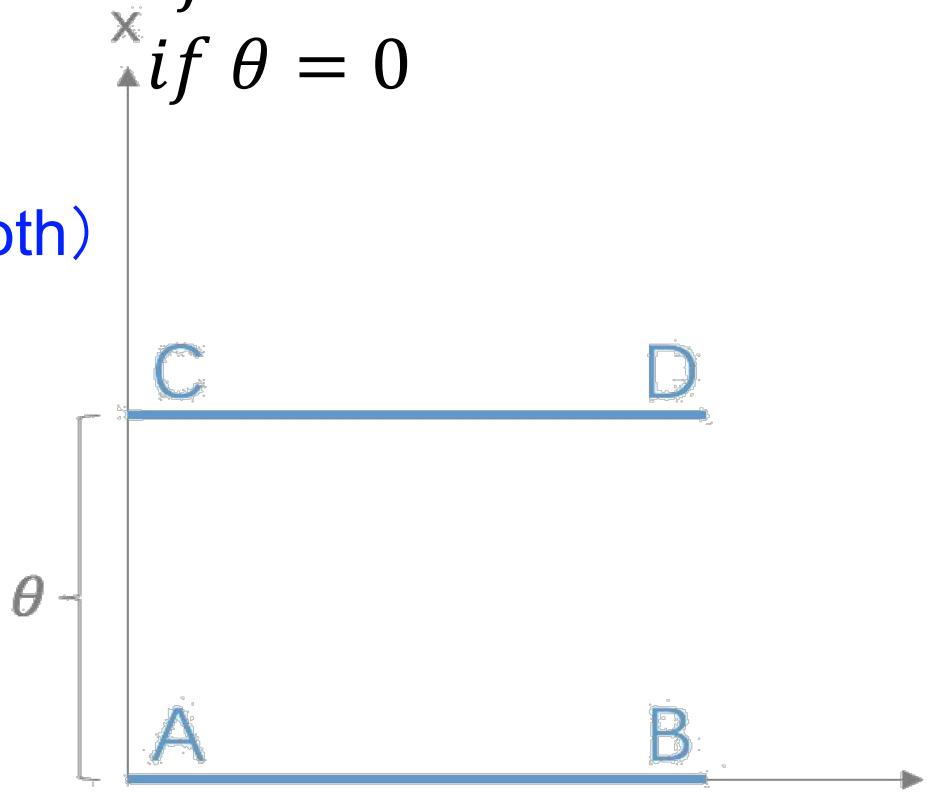
Image credit to: sbl.inria.fr

Examples

$$KL(P_1||P_2) = \begin{cases} +\infty & if\ \theta \neq 0 \\ 0 & if\ \theta = 0 \end{cases}$$

$$JS(P_1||P_2) = \begin{cases} \log 2 & if\ \theta \neq 0 \\ 0 & if\ \theta = 0 \end{cases}$$

$$\mathrm{W}(P_1, P_2) = |\theta| \text{ (Smooth)}$$



Wasserstein GANs

- The Earth-Mover (EM) distance or Wasserstein-1:

$$W(P||Q) = \inf_{\gamma \in \Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$$

have nicer properties when optimized than JS

- However, the infimum is highly intractable.

- Wasserstein distance has a duality form

$$W(P_r, P_g) = \sup_{||f||_L \leq 1} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_g}[f(x)]$$

$$= \frac{1}{K} \sup_{||f||_L \leq K} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_g}[f(x)]$$

where supremum is over all the K-Lipschitz functions

$$\|f(x) - f(y)\| \leqslant L\|x - y\|$$

Wasserstein GANs

- Wasserstein distance has a duality form

$$W(P_r, P_g) = \sup_{||f||_L \leq 1} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_g}[f(x)]$$

$$= \frac{1}{K} \sup_{||f||_L \leq K} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_g}[f(x)]$$

where supremum is over all the K-Lipschitz functions

- Consider a w-parameterized family of functions $\{f_w\}_{w \in W}$ that are all K -Lipschitz

$$W(P_r, P_g) = \max_{w \in W} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{x \sim P_g}[f_w(x)]$$

For example, $W = [-c, c]^l$. To satisfy this requirement, WGAN enforces the weights of D lie within a compact space [-c, c] by applying weight clipping

Wasserstein GANs

- The loss for **discriminator/critic** f_w

$$\mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{x\sim P_g}[f_w(x)]$$

- f_w's goal: maximize Wasserstein distance between real data distribution and generative distribution

- The loss for **generator**

$$-\mathbb{E}_{x\sim P_g}[f_w(x)] = -\mathbb{E}_{z\sim p(z)}[f_w(g_\theta(z))]$$

- g_θ's goal: minimize Wasserstein distance between real data distribution and generative distribution

Algorithm

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**

  for $t = 0, ..., n_{\text{critic}}$ **do**

    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.

    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.

    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$ → compute gradient of f_w

    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$ → gradient ascent on f_w

    $w \leftarrow \text{clip}(w, -c, c)$ → weight clipping to satisfy that functions $\{f_w\}_{w \in W}$ are all K - Lipschitz

  end for

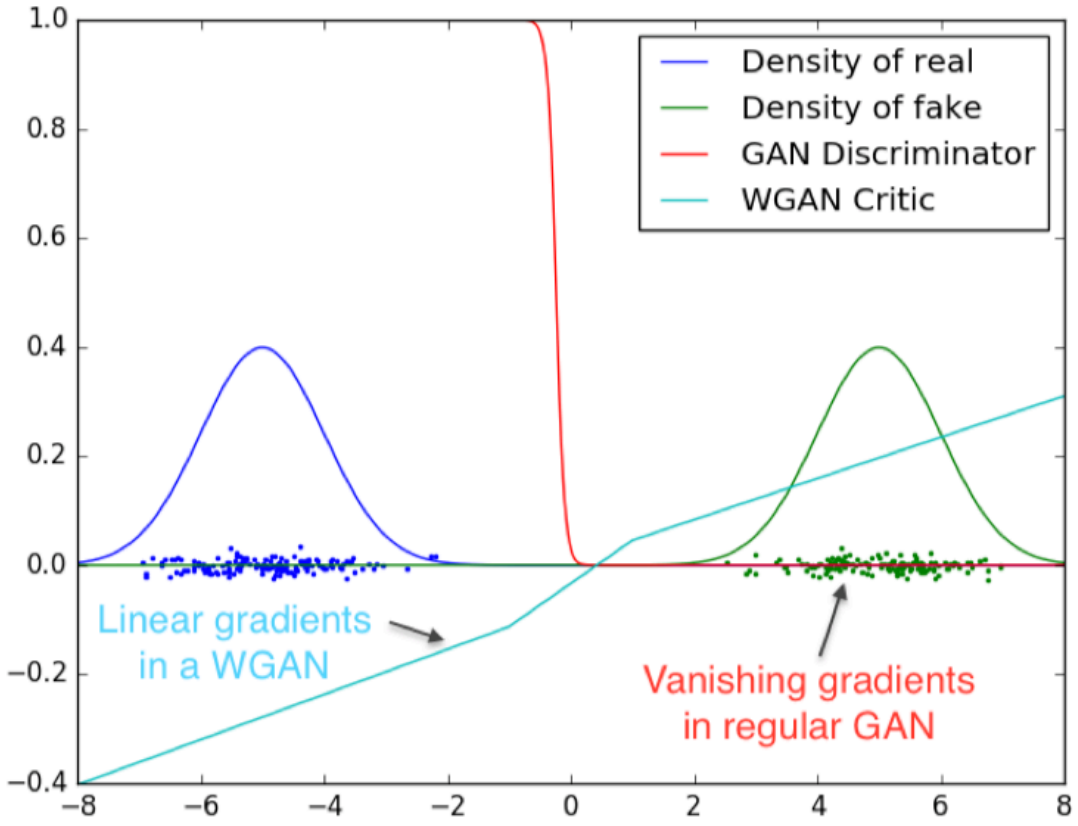
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.

  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$ → compute gradient of g_θ

  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$ → gradient decent on g_θ

end while

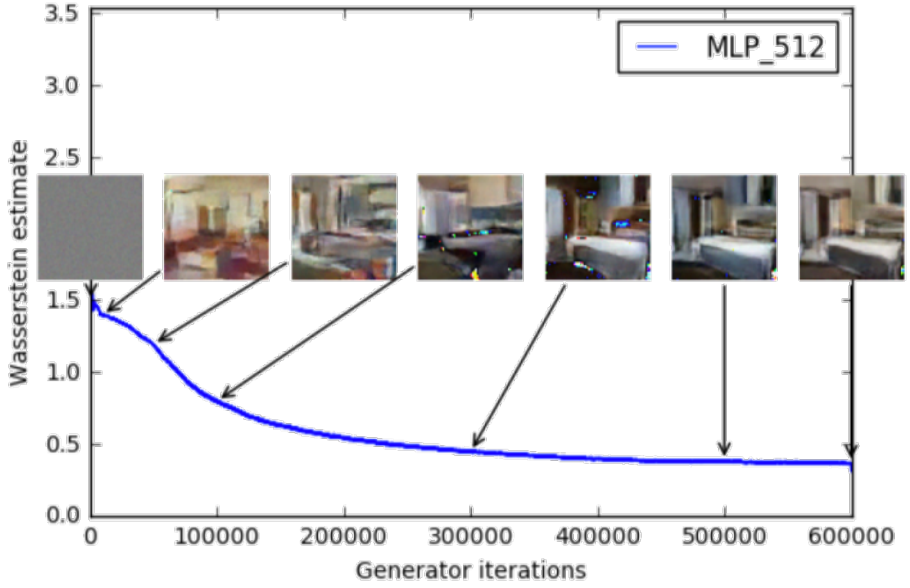
Algorithm



Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the discriminator of a minimax GAN saturates and results in vanishing gradients. WGAN critic provides very clean gradients on all parts of the space.

Image credit to [Martin Arjovsky et al., Wasserstein GAN, 2017]

Meaningful loss metric



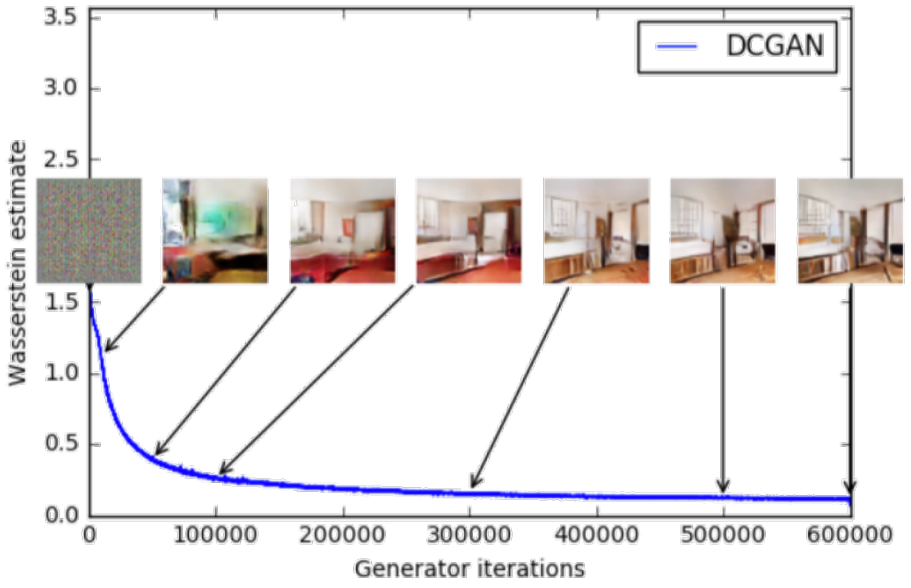
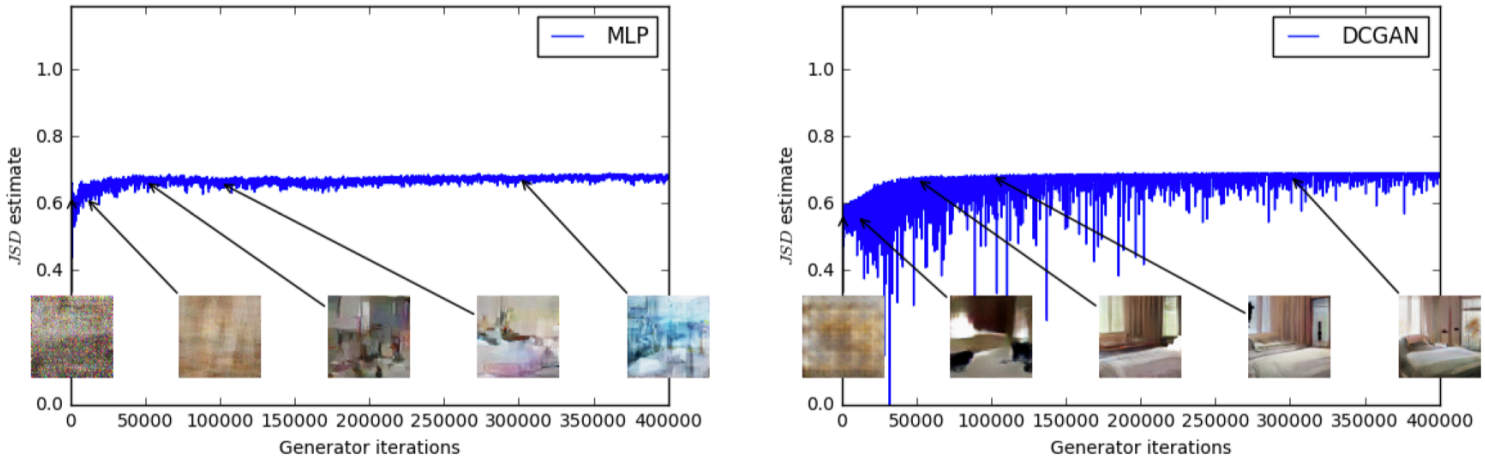


Image credit to [Martin Arjovsky et al., Wasserstein GAN, 2017]

Meaningful loss metric

- Vanilla GANs



JS estimates for an MLP generator (upper left) and a DCGAN generator (upper right) trained with the standard GAN procedure. Both had a DCGAN discriminator. Both curves have increasing error. Samples get better for the DCGAN but the JS estimate increases or stays constant, pointing towards no significant correlation between sample quality and loss.

Image credit to [Martin Arjovsky et al., Wasserstein GAN, 2017]

Image-to-Image Translation

Image de-raining

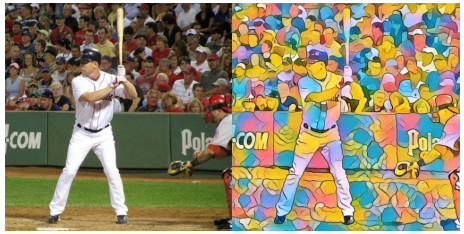
Image impainting

Sketch -> Image

Image Super-resolution

Image coloration

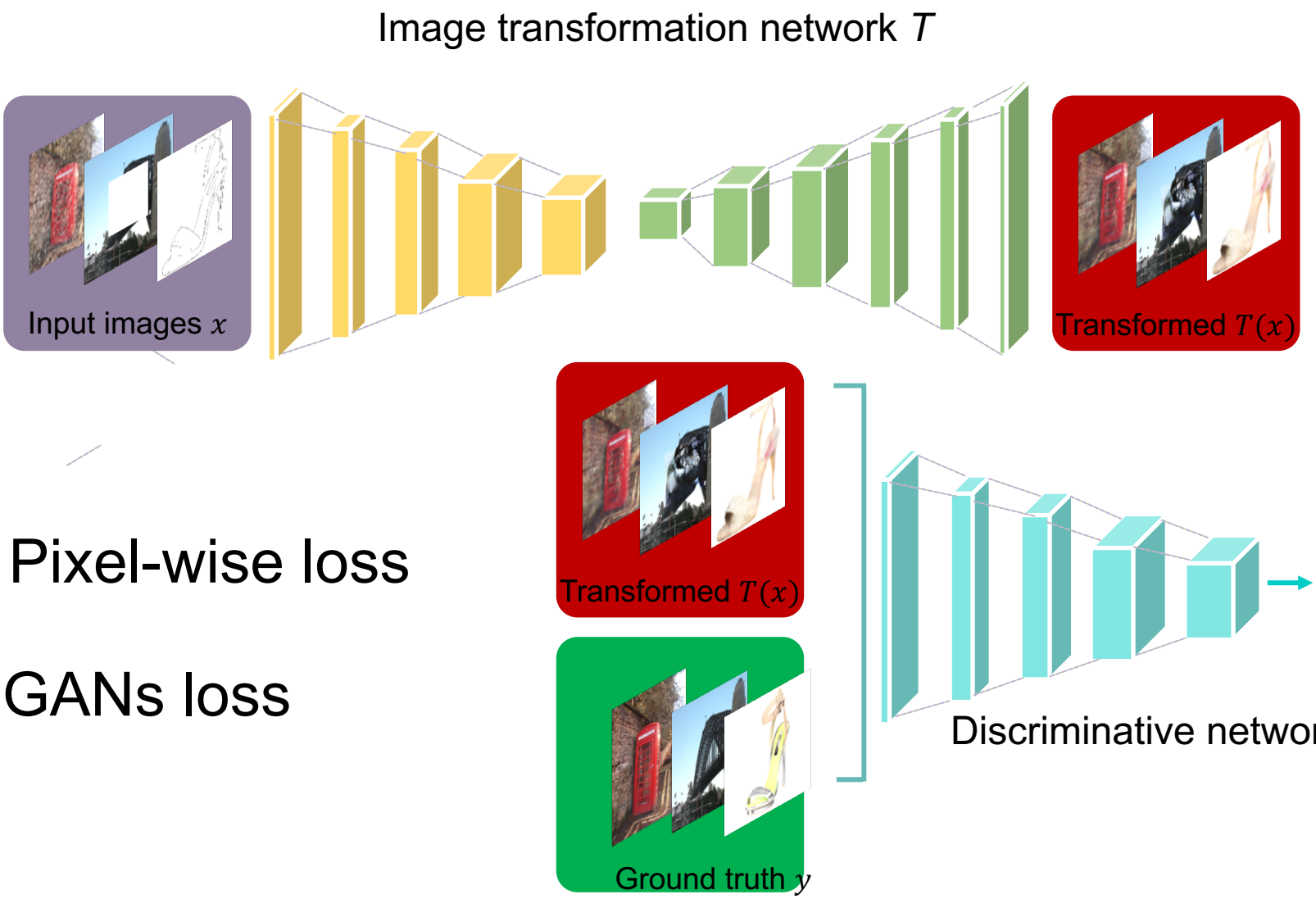
Rotated images



Style transfer

Day -> Night

Image-to-Image Translation

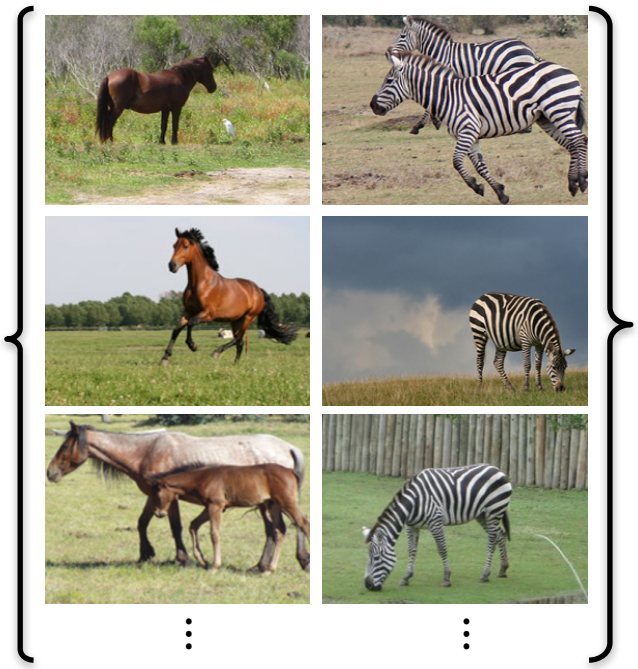


Pixel-wise loss

GANs loss

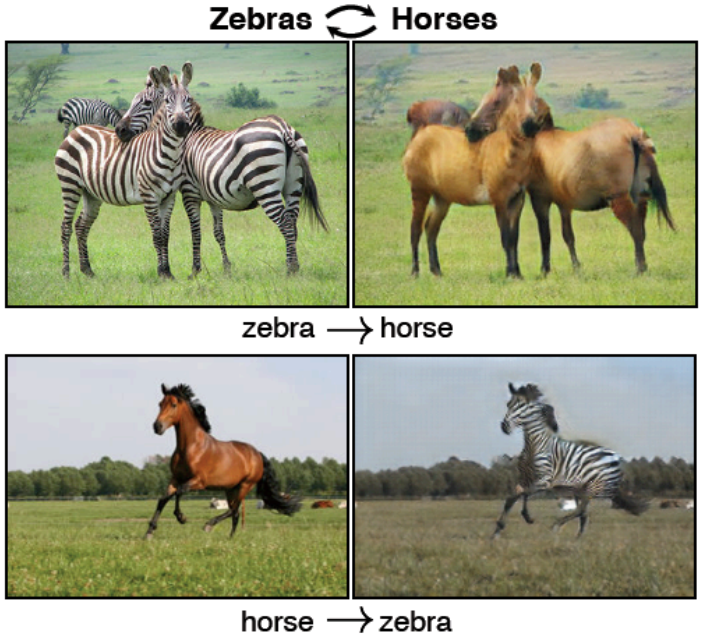
Image-to-Image Translation

- Unpaired Image-to-image Translation (CycleGAN)



horse zebra

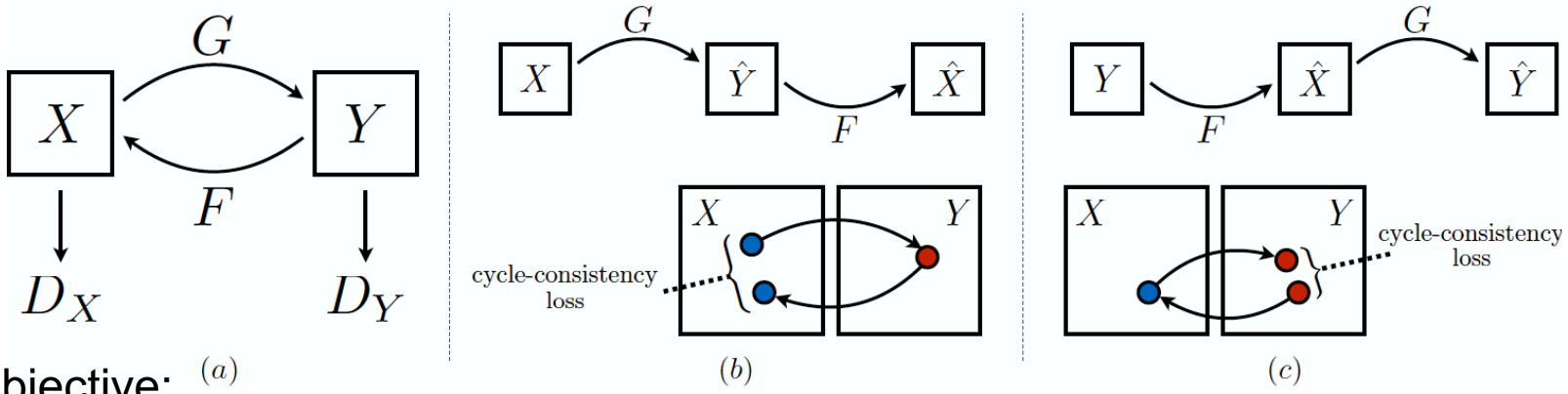
Unpaired Datasets



Object Transfiguration

CycleGAN

- Unpaired Image-to-image Translation (CycleGAN)



Full objective:

$$\mathcal{L}(\mathcal{G}, \mathcal{F}, D_X, D_Y) = \mathcal{L}_{GAN}(\mathcal{G}, D_Y, X, Y) + \mathcal{L}_{GAN}(\mathcal{F}, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(\mathcal{G}, \mathcal{F}),$$

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \in X}\left[\|\mathcal{F}(\mathcal{G}(x)) - x\|_1\right] + \mathbb{E}_{y \in Y}\left[\|\mathcal{G}(\mathcal{F}(y)) - y\|_1\right].$$

$$\mathcal{L}_{GAN}(\mathcal{G}, D_Y, X, Y) = \mathbb{E}_{y \in Y}\left[D_Y^2(y)\right] + \mathbb{E}_{x \in X}\left[(D_Y(\mathcal{G}(x)) - 1)^2\right],$$

Attention-GAN for object transfiguration in wild images [ECCV 2018]

- Two responsibilities for generator G:
  1. detect the objects of interests
  2. convert the object from source domain to another domain.

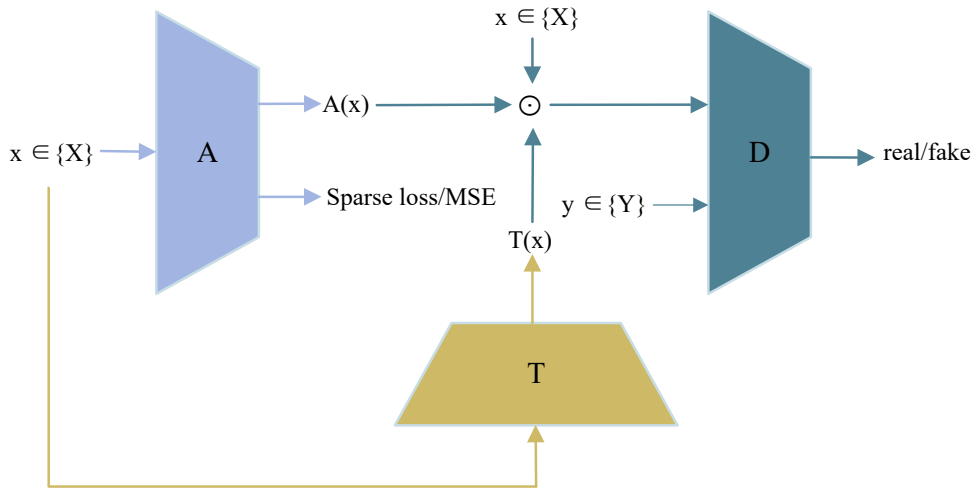


Figure 3 An illustration of Attention-GAN. *A, T, D* respectively represent the attention network, the transformation network and the discriminative network. A(x) denotes the attention map predicted by the attention network. T(x) denotes the transformed images. denotes the layered operation.

Attention-GAN

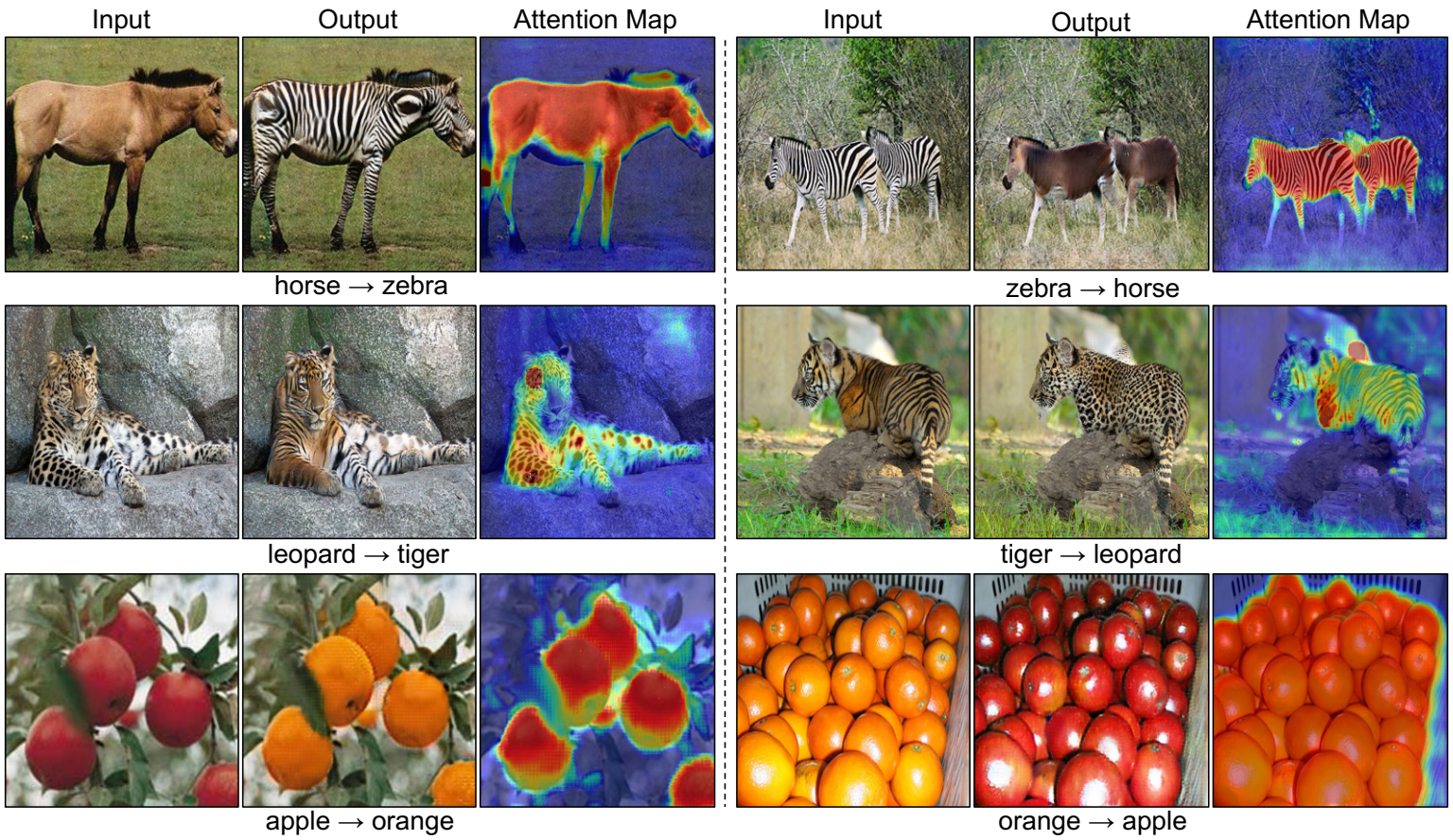


Figure 1 Results of object transfiguration on tasks: horse ↔ zebra, leopard ↔ tiger and apple↔orange. In each case, the first image is the original images, the second image is the synthesized image, and the third image is the predicted attention map. Our proposed model only manipulates the attention parts of image and preserves the background.

Architecture

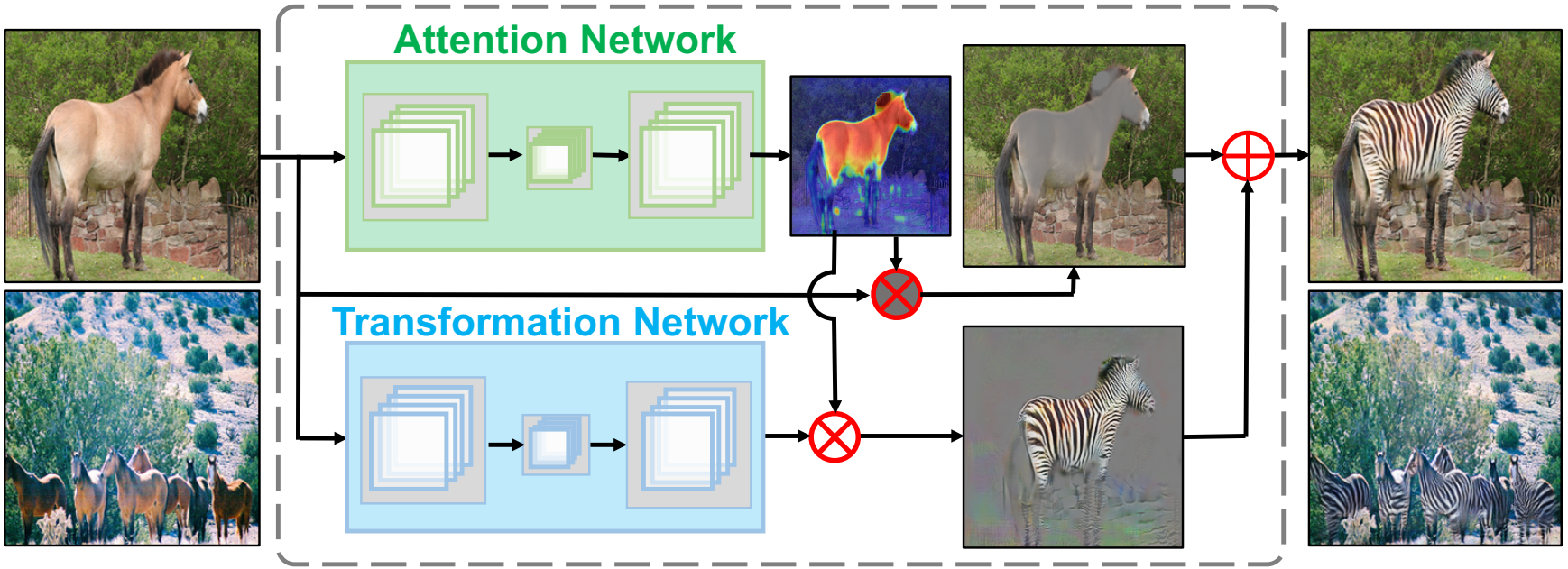


Figure 4 The proposed Attention-GAN for object transfiguration from one class to another. The attention network predicts the attention maps. The transformation network synthesizes the target object. A layered operation is applied on the background and transformed images to output the resulting image.

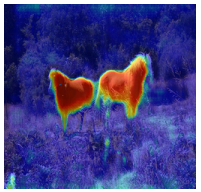
· Formula: $\mathcal{G}(x) \equiv A_X(x) \odot T_X(x) + \big(1 - A_X(x)\big) \odot x$

$$\mathcal{F}(y) \equiv A_Y(y) \odot T_Y(y) + \big(1 - A_Y(y)\big) \odot y.$$

Attention Loss

x y


$A_x(x)$



$A_y(y)$

The regions of interest in the original image and the transformed image should be consistent,

$$\mathcal{L}_{A_{cyc}}(A_X, A_Y)$$
$$= \mathbb{E}_{x \in X}\left[\left\|A_X(x) - \mathrm{A_Y}\big(\mathcal{G}(x)\big)\right\|_1\right] + \mathbb{E}_{y \in Y}\left[\left\|A_Y(y) - \mathrm{A_X}\big(\mathcal{F}(y)\big)\right\|_1\right]$$

Sparse loss encourages the attention network to pay attention to a small region related to the object:

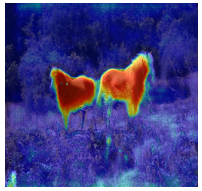
$$\mathcal{L}_{sparse}(A_X, A_Y) = \mathbb{E}_{x \in X}[\|A_X(x)\|_1] + \mathbb{E}_{y \in Y}[\|A_Y(y)\|_1]$$

Attention GAN

x

y

$A_x(x)$



$A_y(y)$

- **Full objective Loss**

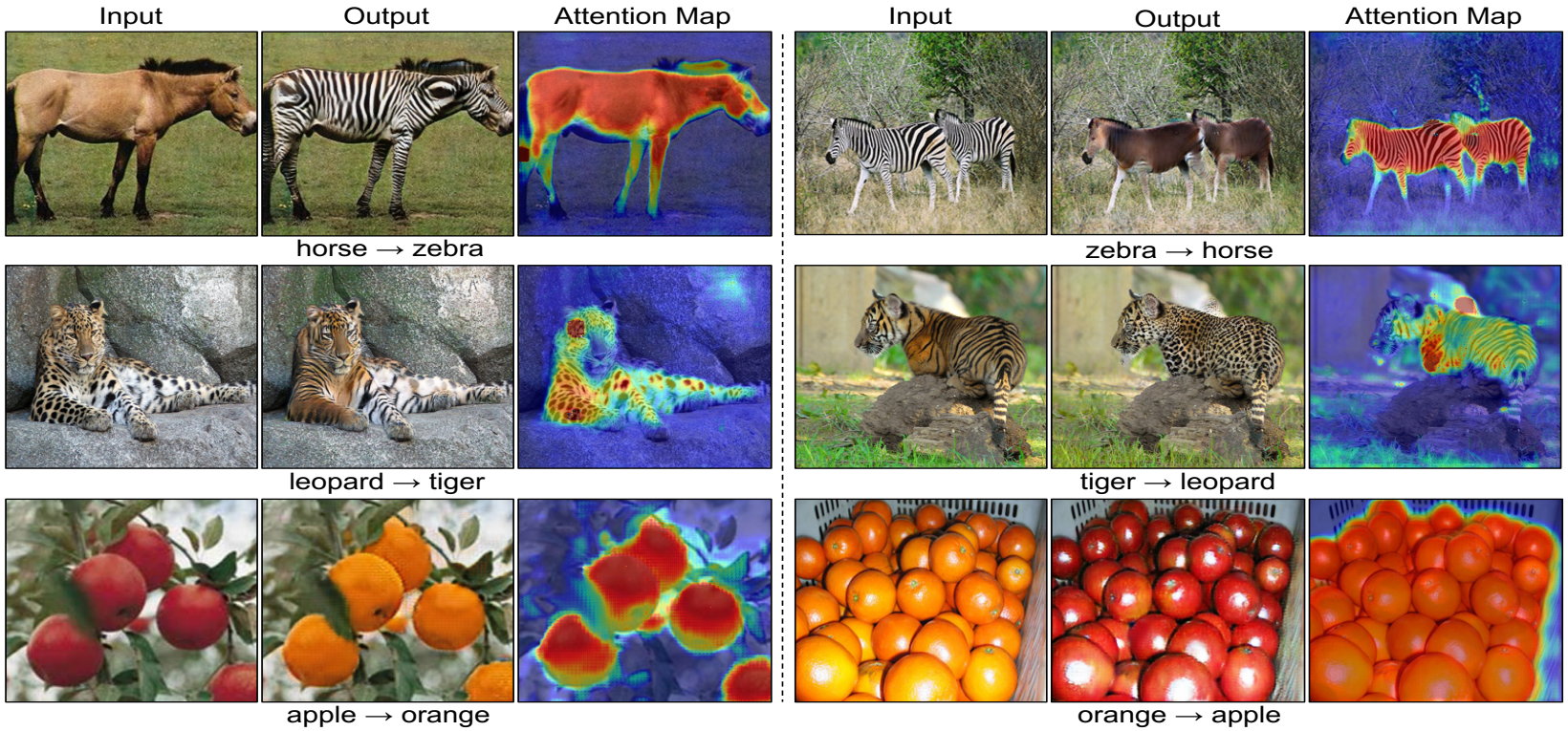
$$\mathcal{L}(T_X, T_Y, D_X, D_Y, A_X, A_Y)$$
$$= \mathcal{L}_{GAN}(\mathcal{G}, D_Y, X, Y) + \mathcal{L}_{GAN}(\mathcal{F}, D_X, X, Y)$$
$$+ \lambda_{cyc}\mathcal{L}_{cyc}(T_X, T_Y, D_X, D_Y, A_X, A_Y) + \lambda_{A_{cyc}}\mathcal{L}_{A_{cyc}}(A_X, A_Y)$$
$$+ \lambda_{sparse}\mathcal{L}_{\text{sparse}}(A_X, A_Y)$$

The networks can can be solved from the following min-max game:

$$\arg \min_{T_X, T_Y, A_X, A_Y} \max_{D_X, D_Y} \mathcal{L}(T_X, T_Y, D_X, D_Y, A_X, A_Y).$$

Experiment

➢ Qualitative results



Human Study

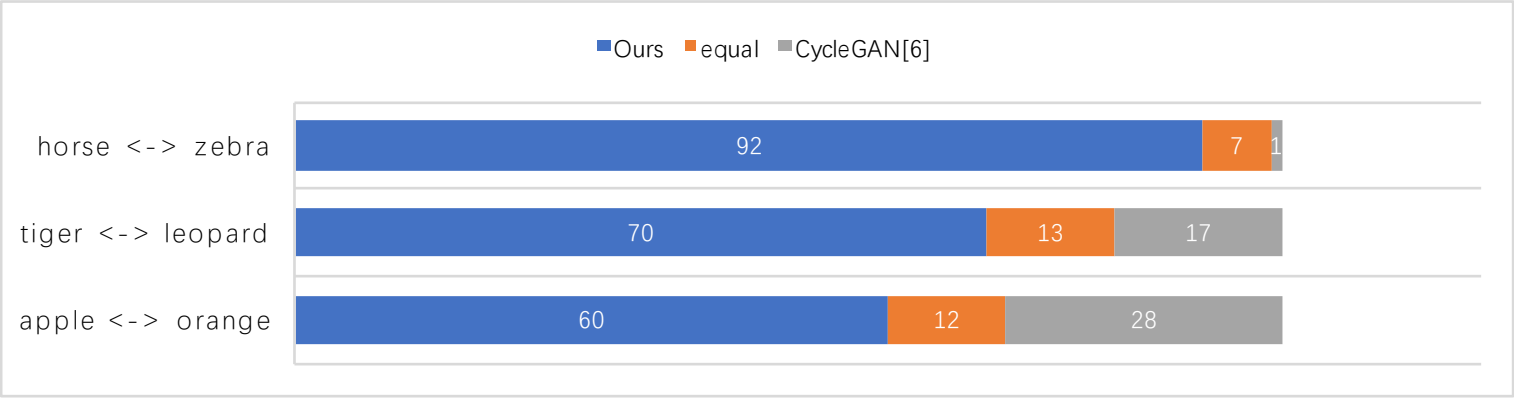


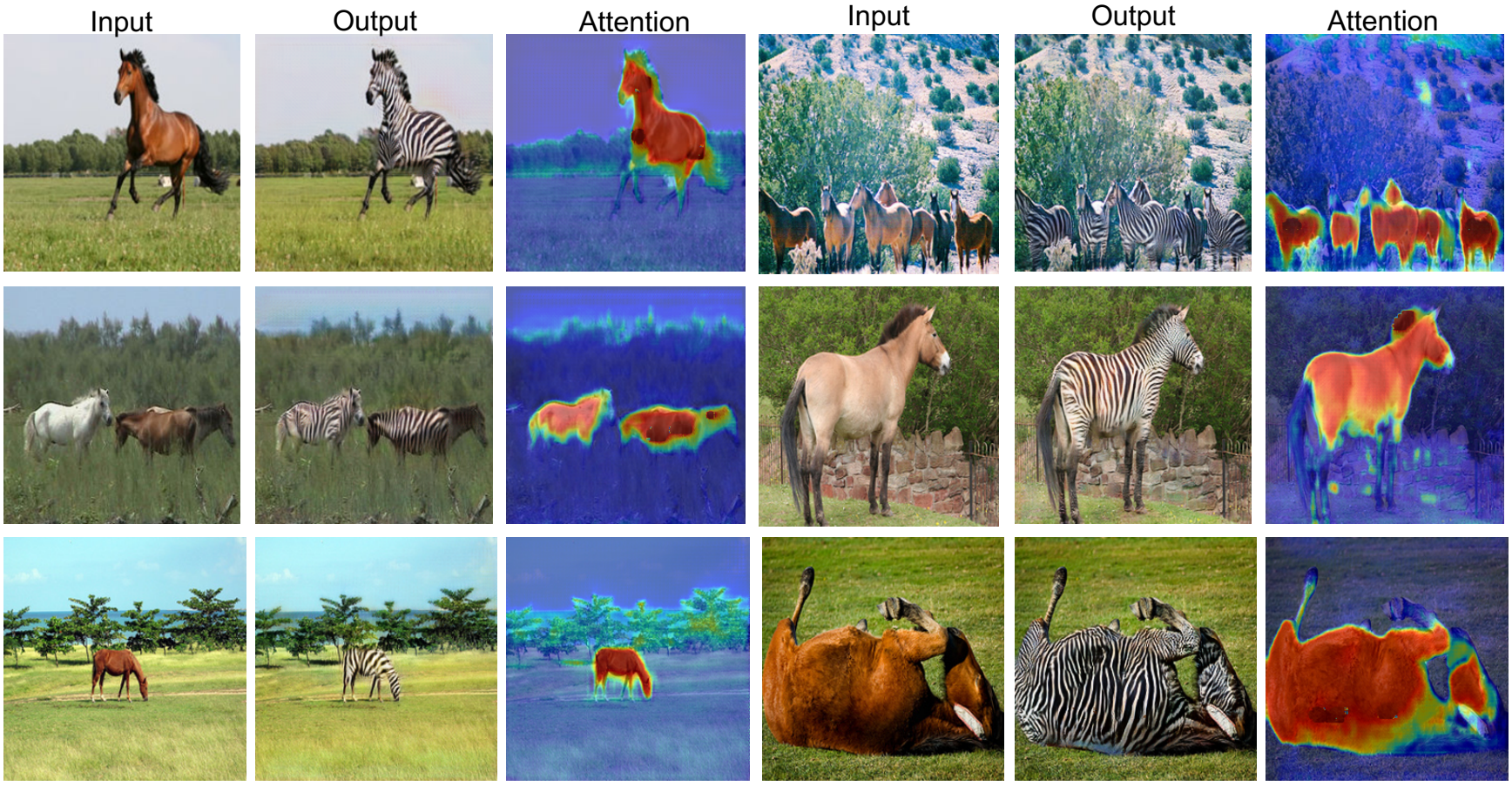
Figure. The stacked bar chart of participants preferences for our methods compared to CycleGAN [21]. The blue bar indicates the number of images that more participants prefer our results. The gray bar indicates the number of images that more participants prefer CycleGAN‘s results. The orange bar indicates the number of images where two methods get a equal number of votes from 10 participants.

Experiments

- **Quantitative comparison**:

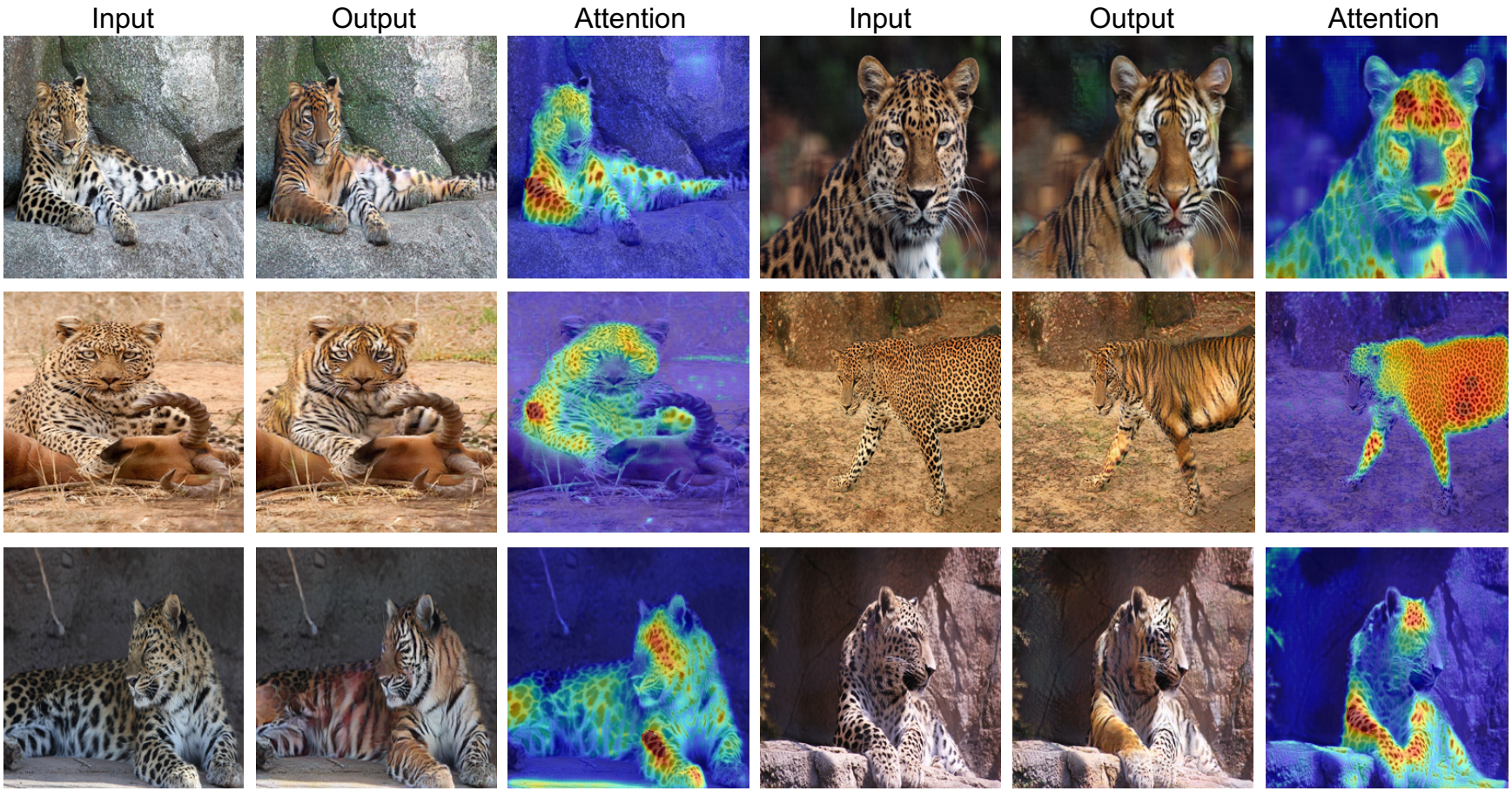
	Task	CycleGAN	DistanceGAN	Ours (Unsupervised)	Ours (Supervised)
PSNR	horse → zebra	18.1875	11.1896	22.2629	24.589
	zebra → horse	18.1021	10.1153	21.5360	23.9330
SSIM	horse → zebra	0.6725	0.2630	0.9003	0.9482
	zebra → horse	0.7155	0.3627	0.8988	0.9534

More Results



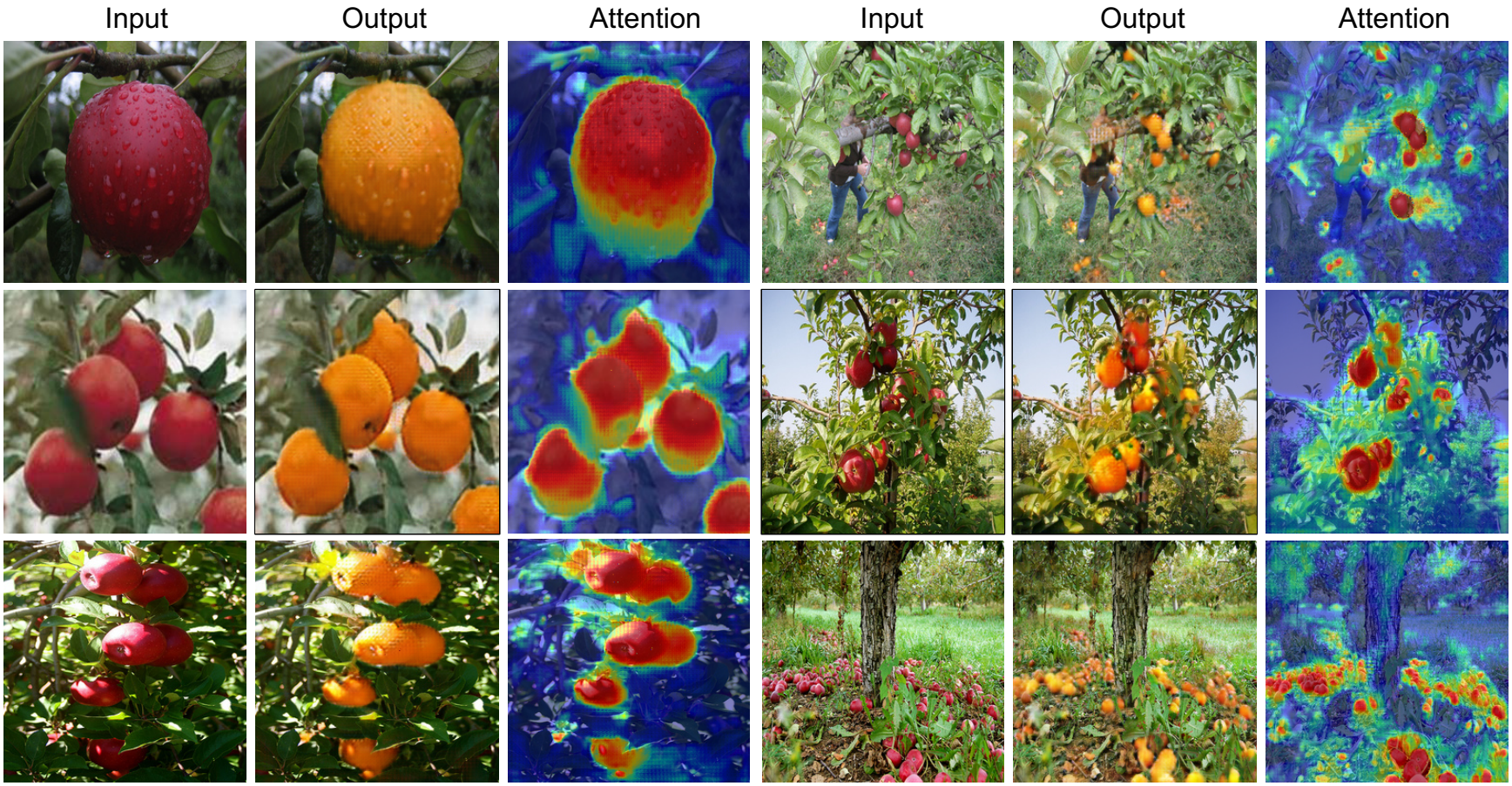
horse → zebra

More Results



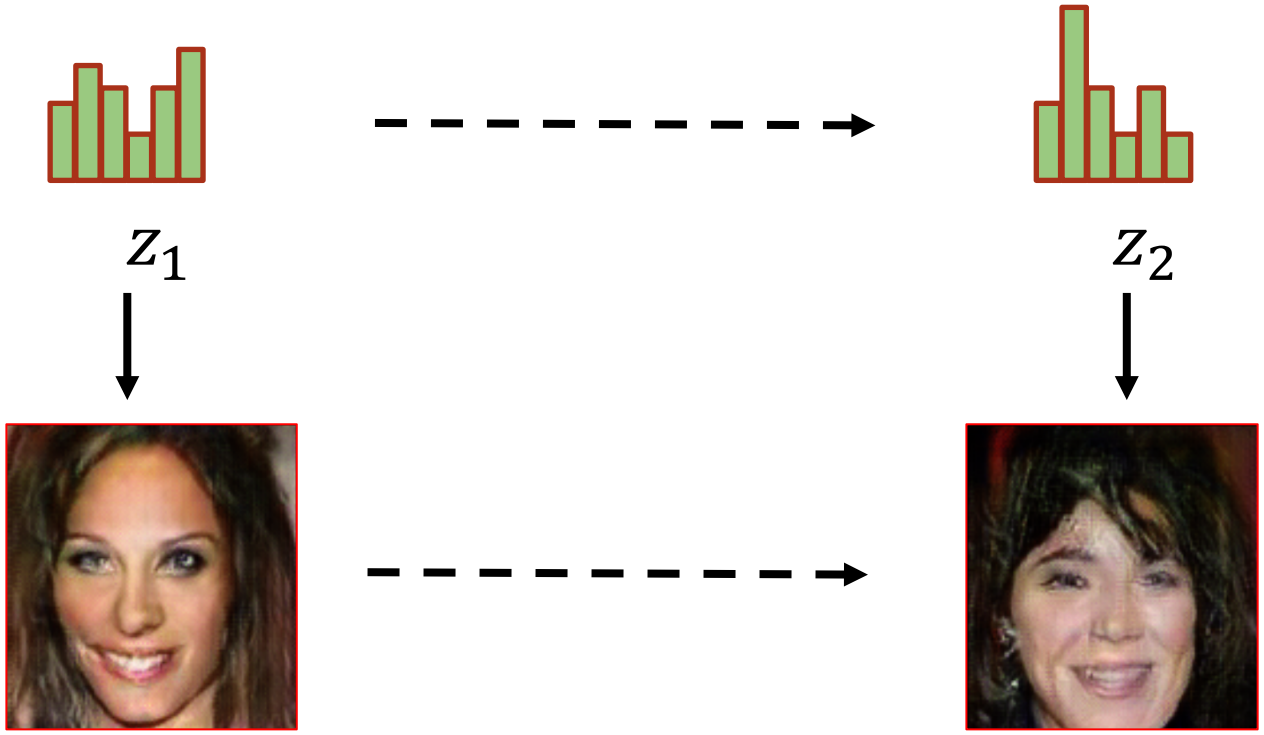
leopard→ tiger

More Results



Apple → Orange

Image Interpolation



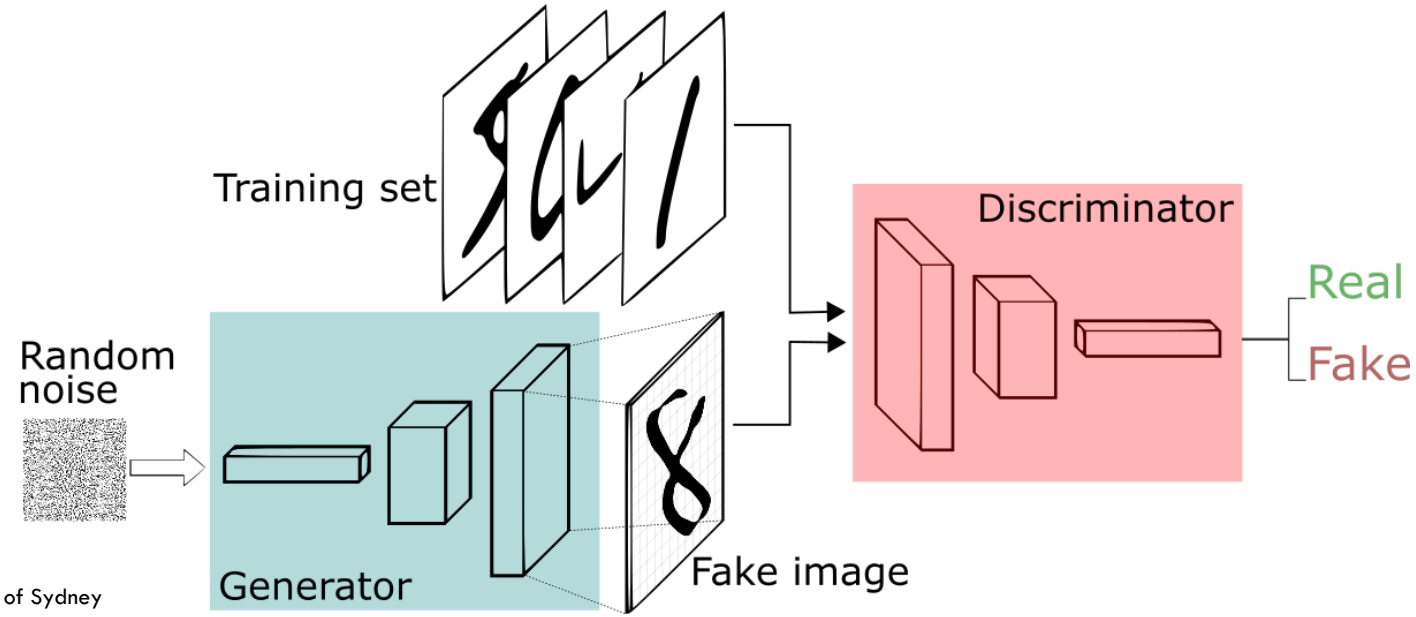
Noise vector inputs for GANs can be taken as low-dimensional representations of images. As widely accepted in representation learning, the closeness of two data points is supposed to be preserved before and after transformation.

Smooth Deep Image Generator from Noises [AAAI 2019]

A **smooth generator** is then expected to have the following pixel-wise property:

$$\left|G_i(\boldsymbol{z}+\boldsymbol{\delta}) - G_i(\boldsymbol{z})\right| < \epsilon$$

where G is the generator, δ stands for a small perturbation over input noise z, and $\epsilon > 0$ is a small constant number.



Smooth Generator

$$\left|G_i(\boldsymbol{z}+\boldsymbol{\delta})-G_i(\boldsymbol{z})\right|<\epsilon$$

How to specify the small perturbation δ?

Theorem 1. *Fix $\epsilon > 0$. Given $\boldsymbol{z} \in \mathbb{R}^d$ as the noise input of generator G_i, if the perturbation $\boldsymbol{\delta} \in \mathbb{R}^d$ satisfies*

$$\|\boldsymbol{\delta}\|_q < \frac{\epsilon}{\max_{\hat{\boldsymbol{z}}\in B_p(\boldsymbol{z},R)} \|\nabla_{\hat{\boldsymbol{z}}} G_i(\hat{\boldsymbol{z}})\|_p}, \tag{3}$$

with $\frac{1}{p}+\frac{1}{q}=1$, we have $\left|G_i(\boldsymbol{z}+\boldsymbol{\delta})-G_i(\boldsymbol{z})\right|<\epsilon$.

By minimizing the denominator in the above equation, the model is expected to tolerate larger perturbation δ under fixed difference ϵ on the i-th pixel.

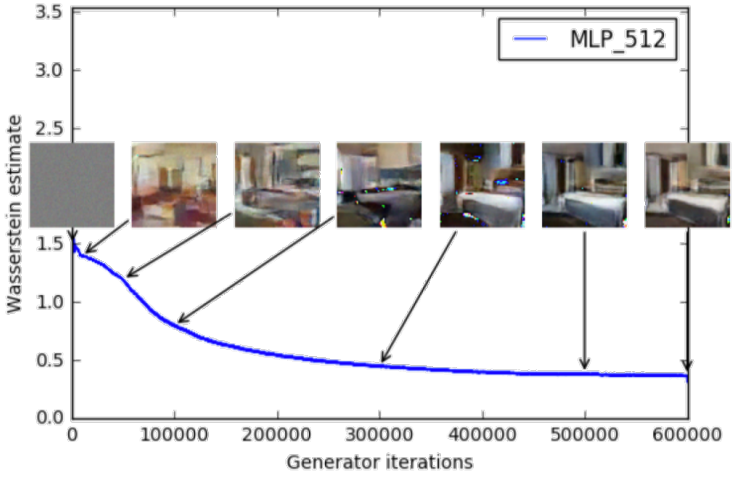
$$\mathcal{L} = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_z} \max(0, \|\nabla_{\boldsymbol{z}} G(\boldsymbol{z})\|_p^2 - k).$$

Smooth GANs

Besides the smoothness of generated images, how about their quality?

In W-GAN, loss function could reflect the image quality:

$$\min_{D} \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_z}[D(G(\boldsymbol{z}))] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_d}[D(\boldsymbol{x})].$$



To decrease the quality gap between images generated from closed noise inputs, we need to ensure that $D(G(z))$ would not drop significantly when the input variates

$$\left|D[G(\boldsymbol{z}+\boldsymbol{\delta})] - D[G(\boldsymbol{z})]\right| < \epsilon.$$

Smooth GANs

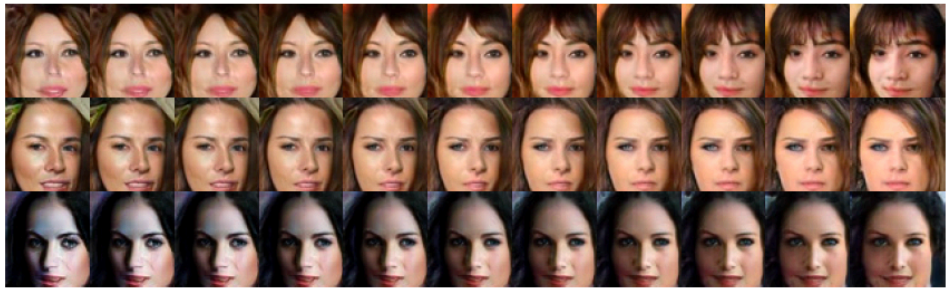
Theorem 2. *Fix $\epsilon > 0$. Consider generator G and discriminator D in GANs. Given a noise input $\boldsymbol{z} \in \mathbb{R}^d$, the generated image is $\hat{\boldsymbol{x}} = G(\hat{\boldsymbol{z}})$. If the perturbation $\boldsymbol{\delta} \in \mathbb{R}^d$ satisfies*

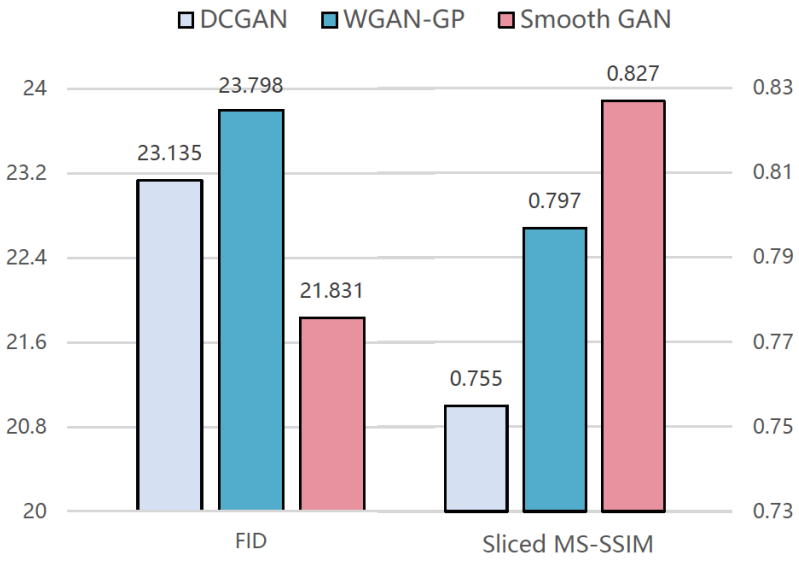
$$\|\boldsymbol{\delta}\|_q < \frac{\epsilon}{\max_{\hat{\boldsymbol{z}} \in B_p(\boldsymbol{z}, R)} \|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_p \|\nabla_{\hat{\boldsymbol{z}}} G(\hat{\boldsymbol{z}})\|_p}, \quad (17)$$

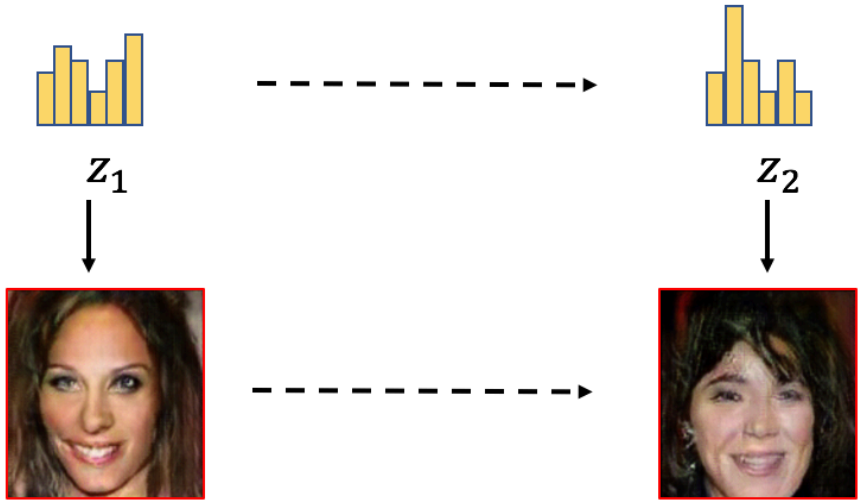
with $\frac{1}{p} + \frac{1}{q} = 1$, we have $\left| D[G(\boldsymbol{z} + \boldsymbol{\delta})] - D[G(\boldsymbol{z})] \right| < \epsilon$.

- Connection with WGAN-GP;
- Encourage the norm of the gradient of generator to stay at a lower level.

Experiments





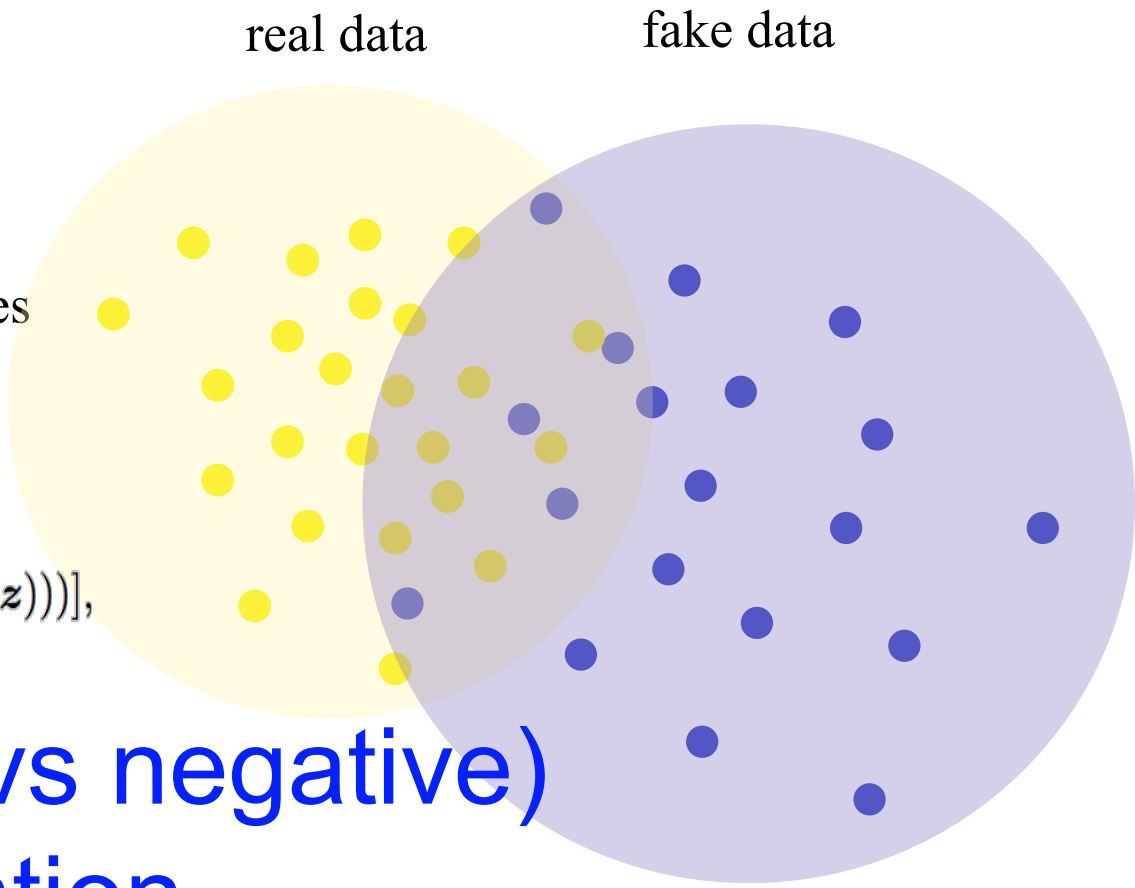


FID and sliced MS-SSIM of interpolated images on the CIFAR-10 dataset.

On Positive-Unlabeled Classification in GAN [CVPR 2020]

Traditional GANs

- Separating real and fake data
- Ignoring good generated samples



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

Binary (positive vs negative) classification

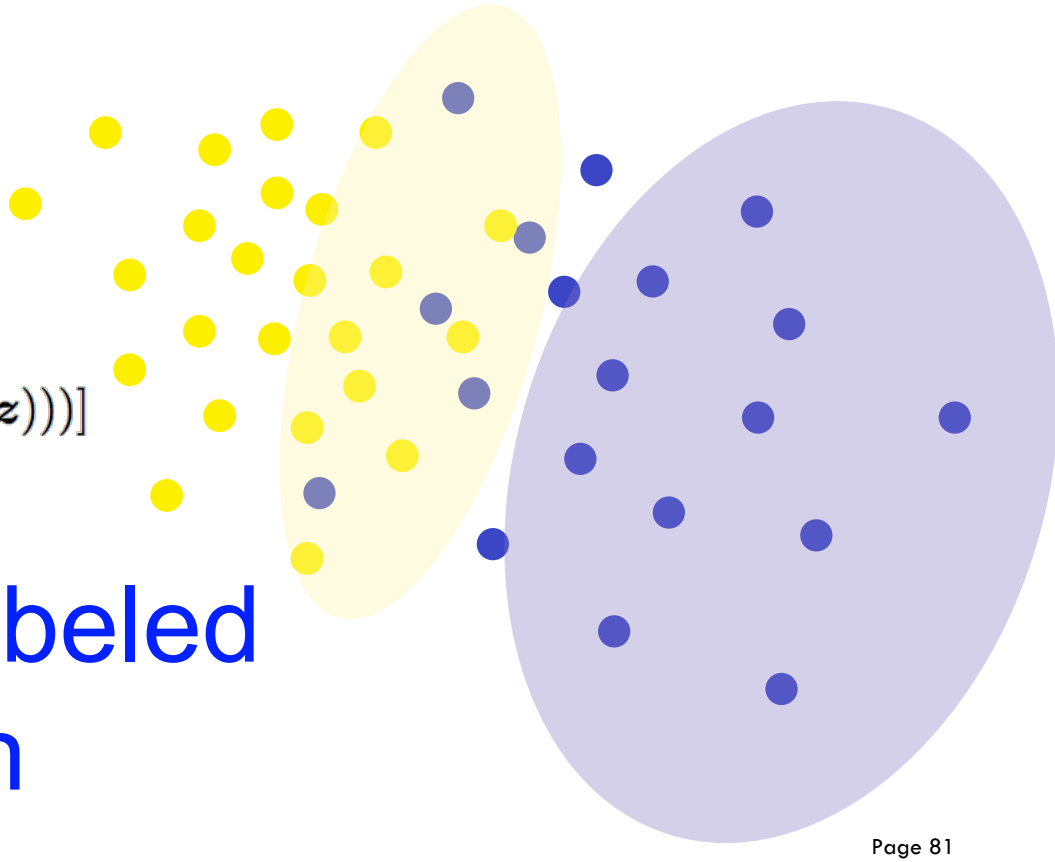
Positive-unlabeled GAN (PUGAN)

- Separating good samples from generated samples referring to real data

real data fake data

- Focusing on low-quality samples

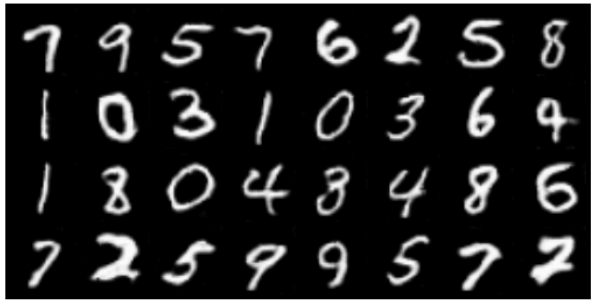
$$\max_G \min_D V(D,G) = \pi \mathbb{E}_{p_{data}}[f_1(D(\boldsymbol{x}))] + \max\left\{0, \mathbb{E}_{p_z}[f_2(D(G(\boldsymbol{z})))] - \pi \mathbb{E}_{p_{data}}[f_2(D(\boldsymbol{x})]\right\}.$$



Positive and unlabeled classification

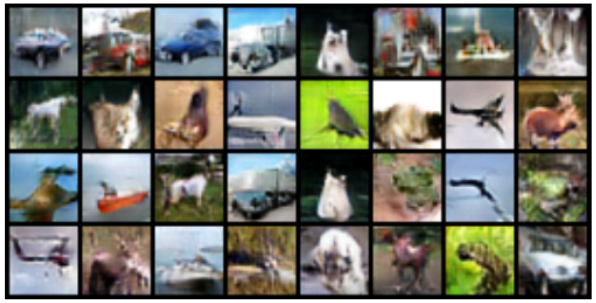
Positive-unlabeled GAN (PUGAN)

Loss	SGAN	PUSGAN	HingeGAN	PUHingeGAN	LSGAN	PULSGAN	WGAN-GP	PUWGAN-GP
MNIST	18.65	16.53	21.48	16.91	**13.47**	13.96	17.42	14.77
Fashion	25.72	24.33	28.39	25.31	32.05	26.72	26.50	**23.12**
CIFAR-10	43.39	31.02	43.85	36.37	27.64	**22.32**	36.86	31.85
CelebA-64	48.44	43.89	46.13	43.94	51.67	47.80	36.09	**35.72**
CAT-64	46.13	**16.90**	29.52	25.72	57.22	26.79	21.86	17.35

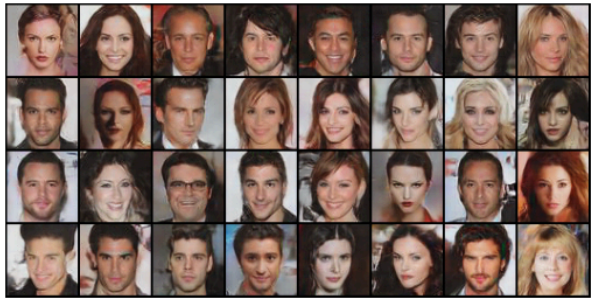


(a) MNIST

(b) Fashion

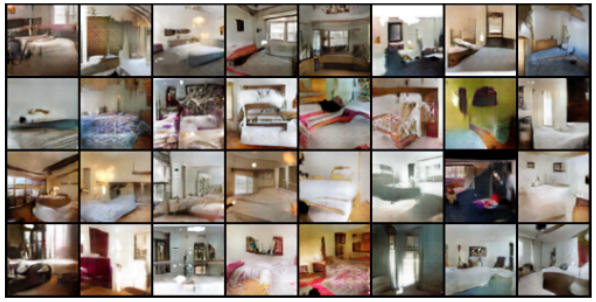


(c) CIFAR-10



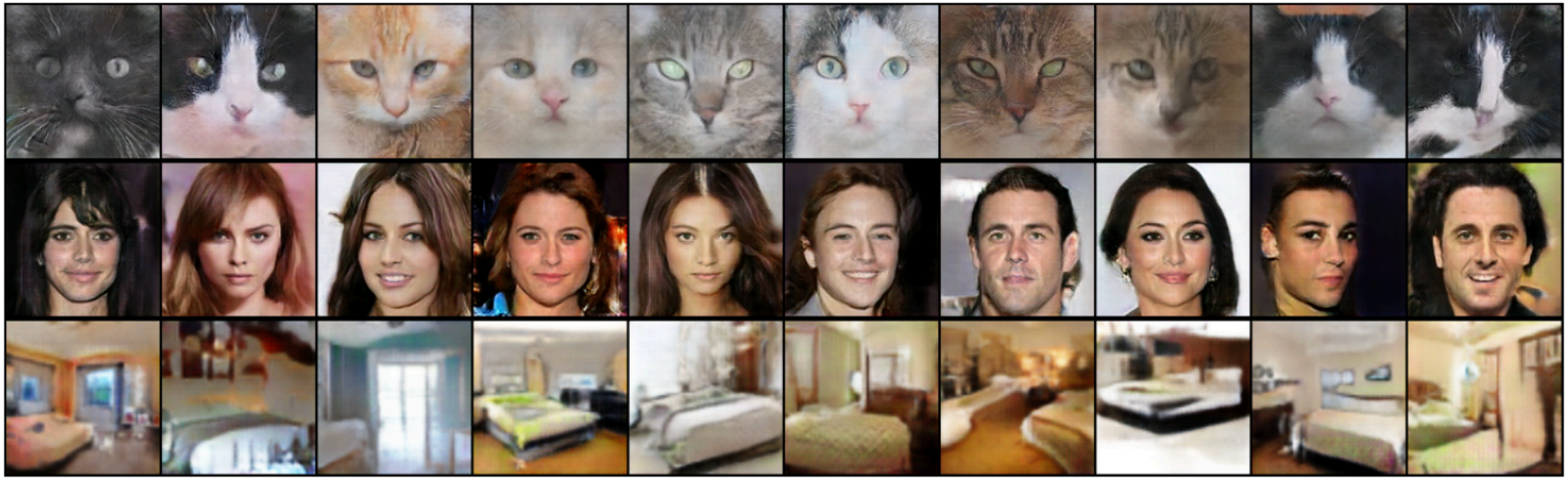
(d) CelebA-64

(e) CAT-64



(a) LSUN-64

Figure 1: Generated samples obtained by the proposed method on image datasets.



(a) Generation samples on 128 resolution datasets.

(b) Generation samples on the CAT-256 dataset.

Figure 2: High-resolution generated samples.

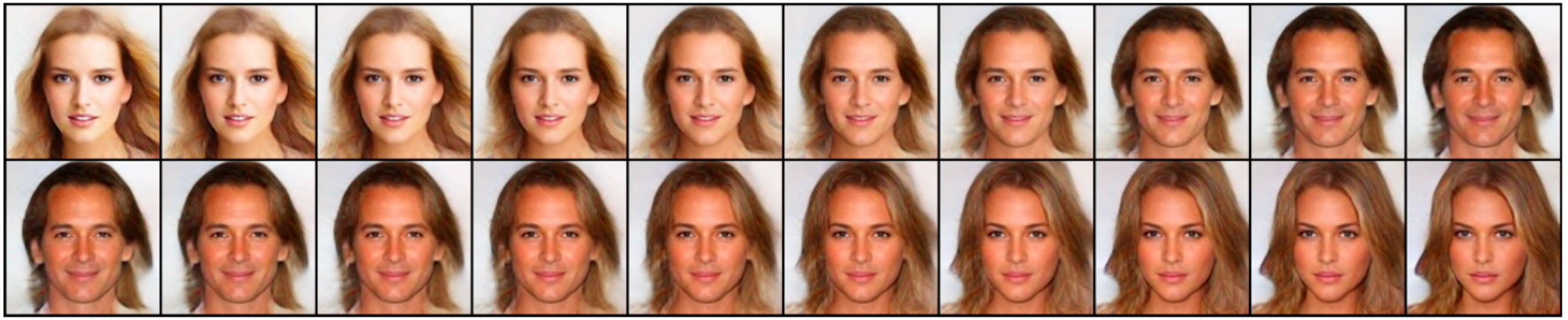


Figure 3: High-resolution generated samples.