Triple Generative Adversarial Nets

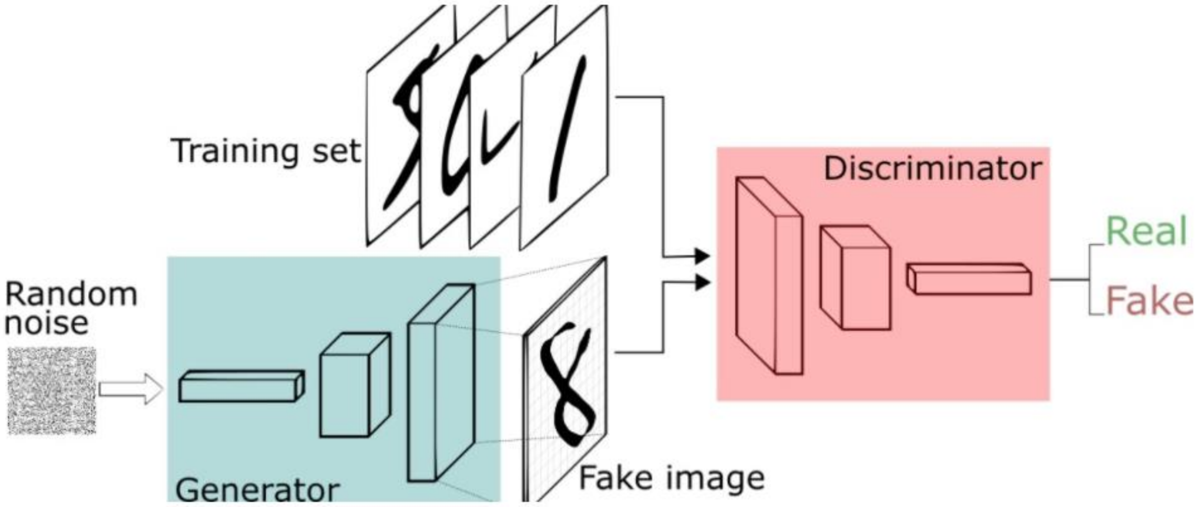
Chongxuan Li, Kun Xu, Jun Zhu,[*] Bo Zhang
Dept. of Comp. Sci. & Tech., TNList Lab, State Key Lab of Intell. Tech. & Sys.,
Center for Bio-Inspired Computing Research, Tsinghua University, Beijing, 100084, China
{licx14, xu-k16}@mails.tsinghua.edu.cn, {dcszj, dcszb}@mail.tsinghua.edu.cn

NIPS-2017

Contents

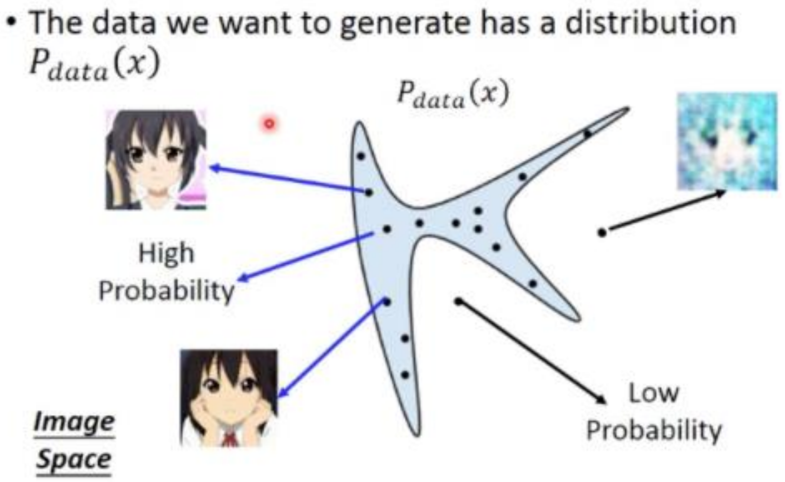
- Introduction
- Motivation
- Method
- Experiment

Introduction – GAN

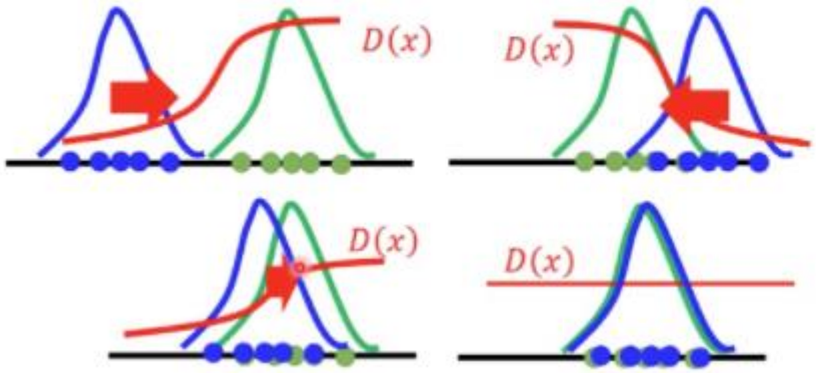


GAN

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

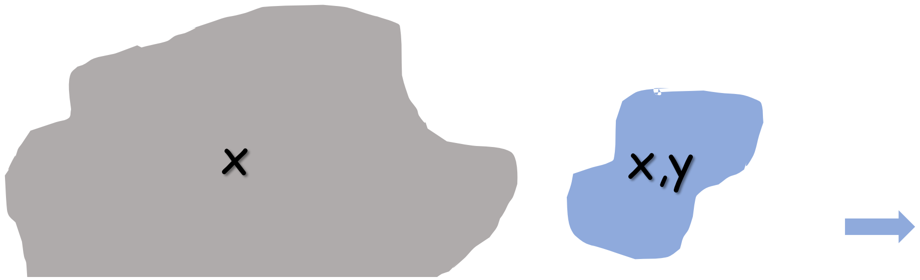


$P_{data}(x)$



$P_g(x)$ $P_{data}(x)$

Introduction — SSL-GAN



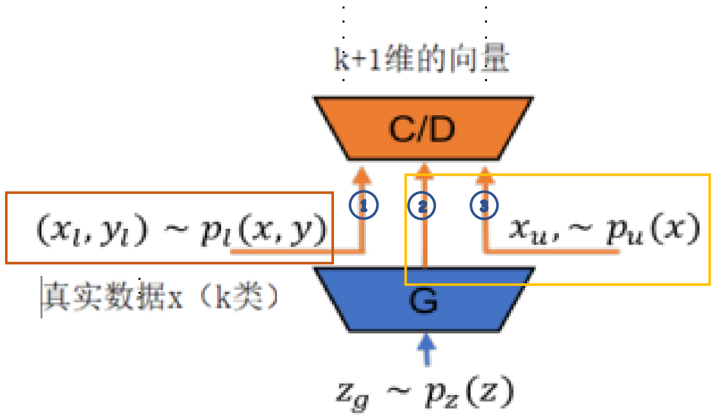
$$L_{C/D} = L_{\text{supervised}} + L_{\text{unsupervised}}$$

$$L_{\text{supervised}} = \mathop{\mathbb{E}}_{\boldsymbol{x}, y \sim p_l(\boldsymbol{x}, y)} [-\log(p_{C/D}(y|\boldsymbol{x}, y < K+1))]$$

$$L_{\text{unsupervised}} = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_u(\boldsymbol{x})} [-\log(1 - p_{C/D}(y = K+1|\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_g(\boldsymbol{x})} [-\log(p_{C/D}(y = K+1|\boldsymbol{x}))]$$

$$L_G = \left\| \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_u} (\boldsymbol{f}(\boldsymbol{x})) - \mathop{\mathbb{E}}_{\boldsymbol{z}_g \sim p_z(z)} (\boldsymbol{f}(G(\boldsymbol{z}_g))) \right\|_2^2$$

- Generative model



semi-supervised GAN[1]

The unsupervised loss requires C/D to put the synthetic data from generator $x \sim P_g(x)$ into the (K + 1)th class, while putting the unlabeled data $x \sim P_u(x)$ into the real K classes.

[1]Improved techniques for training gans. In Advances in neural information processing systems, 2016.

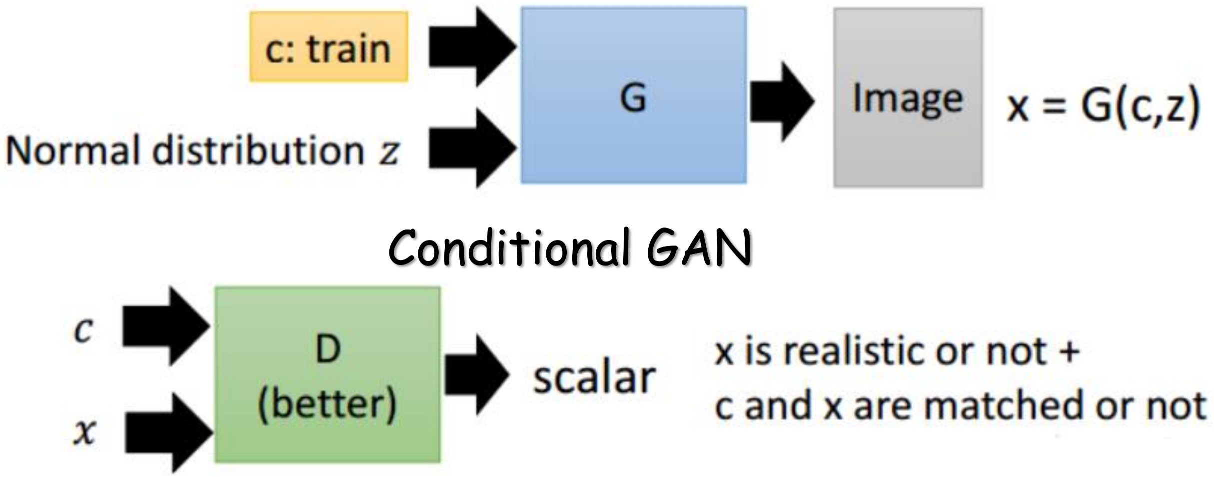
Motivation

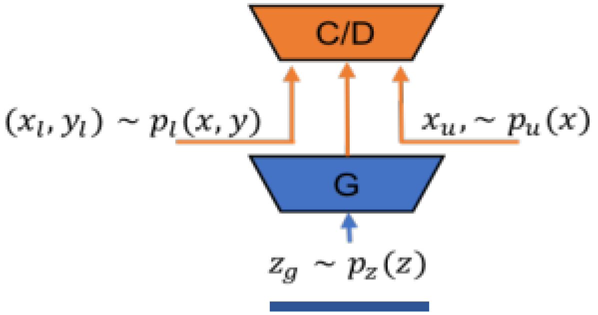
- The generator and the discriminator (i.e. the classifier) may not be optimal at the

G is

D:

D h





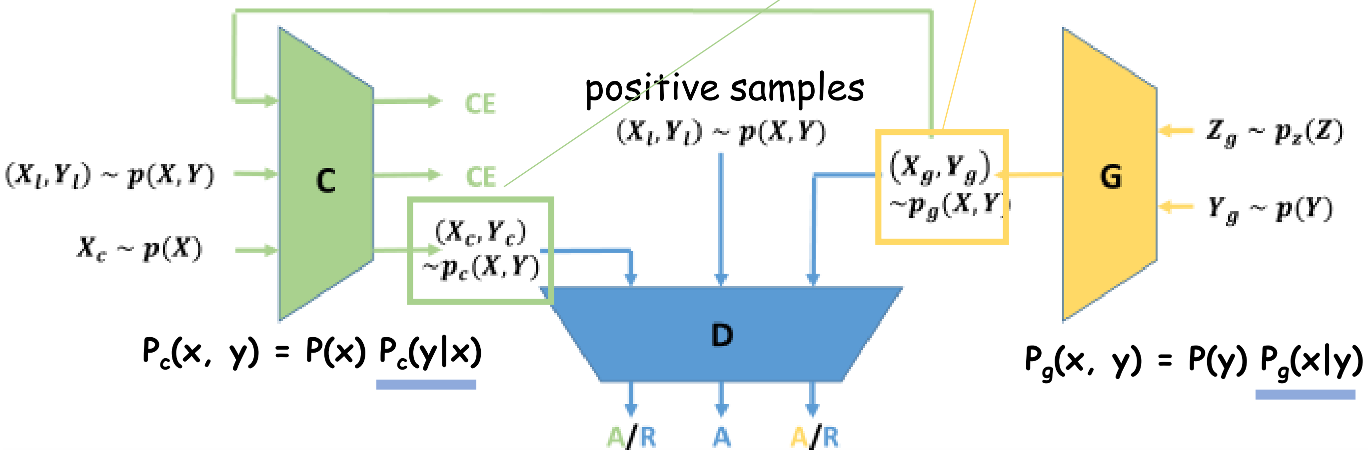
- The generator cannot control the semantics of the generated samples.

Specifically, the discriminators in take a single data instead of a data-label pair as input and the label information is totally ignored when justifying whether a sample is real or fake.

Method

Triple-GAN

$P_\alpha(x, y) := (1 - \alpha)P_g(x, y) + \alpha P_c(x, y) \quad \alpha = 1/2$



- $p(x)$ is the empirical distribution of inputs
- $p(y)$ is assumed same to the distribution of labels on labeled data
- $p(x,y) = p_g(x,y) = p_c(x,y)$

Object function: $\min_{C,G} \max_{D} U(C, G, D) = E_{(x,y)\sim p(x,y)}[\log D(x, y)] + \alpha E_{(x,y)\sim p_c(x,y)}[\log(1 - D(x, y))] + (1-\alpha)E_{(x,y)\sim p_g(x,y)}[\log(1 - D(G(y, z), y))],$ $\alpha = 1/2$

- It achieves its equilibrium if and only if $P(x, y) = (1 - \alpha)Pg(x, y) + \alpha Pc(x, y)$
- The equilibrium indicates that if one of C and G tends to the data distribution, the other will also go towards the data distribution, which addresses the competing problem.

Method

Standard supervised (cross-entropy) loss R_L:

$$R_L = D_{KL}(p(x,y)||p_c(x,y))$$
$$= \iint_{(x,y)} p(x,y) \log \frac{p((x,y)}{p_c(x,y)} dxdy$$
$$= \iint_{(x,y)} p(x,y) \log \frac{p(x)p(y|x)}{p_c(x)p_c(y|x)} dxdy$$

$$\mathcal{R}_{\mathcal{L}} = E_{(x,y)\sim p(x,y)}[-\log p_c(y|x)]$$

Pseudo discriminative loss R_P:

$$D_{KL}(p_g(x,y)||p_c(x,y)) + H_{p_g}(y|x) - D_{KL}(p_g(x)||p(x))$$
$$= \iint p_g(x,y) \log \frac{p_g(x,y)}{p_c(x,y)} + p_g(x,y) \log \frac{1}{p_g(y|x)} dxdy - \int p_g(x) \log \frac{p_g(x)}{p(x)} dx$$
$$= \iint p_g(x,y) \log \frac{p_g(x,y)}{p_c(x,y)p_g(y|x)} dxdy - \iint p_g(x,y) \log \frac{p_g(x)}{p(x)} dxdy$$
$$= \iint p_g(x,y) \log \frac{p_g(x,y)p(x)}{p_c(x,y)p_g(y|x)p_g(x)} dxdy$$
$$= E_{p_g}[-\log p_c(y|x)].$$

Method

Algorithm 1 Minibatch stochastic gradient descent training of Triple-GAN in SSL.

for number of training iterations **do**

- Sample a batch of pairs $(x_g, y_g) \sim p_g(x, y)$ of size m_g, a batch of pairs $(x_c, y_c) \sim p_c(x, y)$ of size m_c and a batch of labeled data $(x_d, y_d) \sim p(x, y)$ of size m_d.
- Update D by ascending along its stochastic gradient:

$$\nabla_{\theta_d}\left[\frac{1}{m_d}\Big(\sum_{(x_d,y_d)}\log D(x_d,y_d)\Big)+\frac{\alpha}{m_c}\sum_{(x_c,y_c)}\log(1-D(x_c,y_c))+\frac{1-\alpha}{m_g}\sum_{(x_g,y_g)}\log(1-D(x_g,y_g))\right].$$

- Compute the unbiased estimators $\tilde{\mathcal{R}}_{\mathcal{L}}$ and $\tilde{\mathcal{R}}_{\mathcal{P}}$ of $\mathcal{R}_{\mathcal{L}}$ and $\mathcal{R}_{\mathcal{P}}$ respectively.
- Update C by descending along its stochastic gradient:

$$\nabla_{\theta_c}\left[\frac{\alpha}{m_c}\sum_{(x_c,y_c)} p_c(y_c|x_c)\log(1-D(x_c,y_c))+\tilde{\mathcal{R}}_{\mathcal{L}}+\alpha_{\mathcal{P}}\tilde{\mathcal{R}}_{\mathcal{P}}\right].$$

- Update G by descending along its stochastic gradient:

$$\nabla_{\theta_g}\left[\frac{1-\alpha}{m_g}\sum_{(x_g,y_g)}\log(1-D(x_g,y_g))\right].$$

end for

Experiment-Classification

Table 1: Error rates (%) on partially labeled MNIST, SHVN and CIFAR10 datasets, averaged by 10 runs. The results with † are trained with more than 500,000 extra unlabeled data on SVHN.

Algorithm	MNIST $n = 100$	SVHN $n = 1000$	CIFAR10 $n = 4000$
M1+M2 [11]	3.33 (±0.14)	36.02 (±0.10)	
VAT [18]	2.33		24.63
Ladder [23]	1.06 (±0.37)		20.40 (±0.47)
Conv-Ladder [23]	**0.89** (±0.50)		
ADGM [17]	0.96 (±0.02)	22.86 †	
SDGM [17]	1.32 (±0.07)	16.61(±0.24)†	
MMCVA [15]	1.24 (±0.54)	**4.95** (±0.18) †	
CatGAN [26]	1.39 (±0.28)		19.58 (±0.58)
Improved-GAN [25]	0.93 (±0.07)	8.11 (±1.3)	18.63 (±2.32)
ALI [5]		7.3	18.3
Triple-GAN (**ours**)	**0.91** (±0.58)	**5.77**(±0.17)	**16.99** (±0.36)

Table 2: Error rates (%) on MNIST with different number of labels, averaged by 10 runs.

Algorithm	$n = 20$	$n = 50$	$n = 200$
Improved-GAN [25]	16.77 (±4.52)	2.21 (±1.36)	0.90 (±0.04)
Triple-GAN (**ours**)	**4.81** (±4.95)	**1.56** (±0.72)	**0.67** (±0.16)

Experiment-Generation

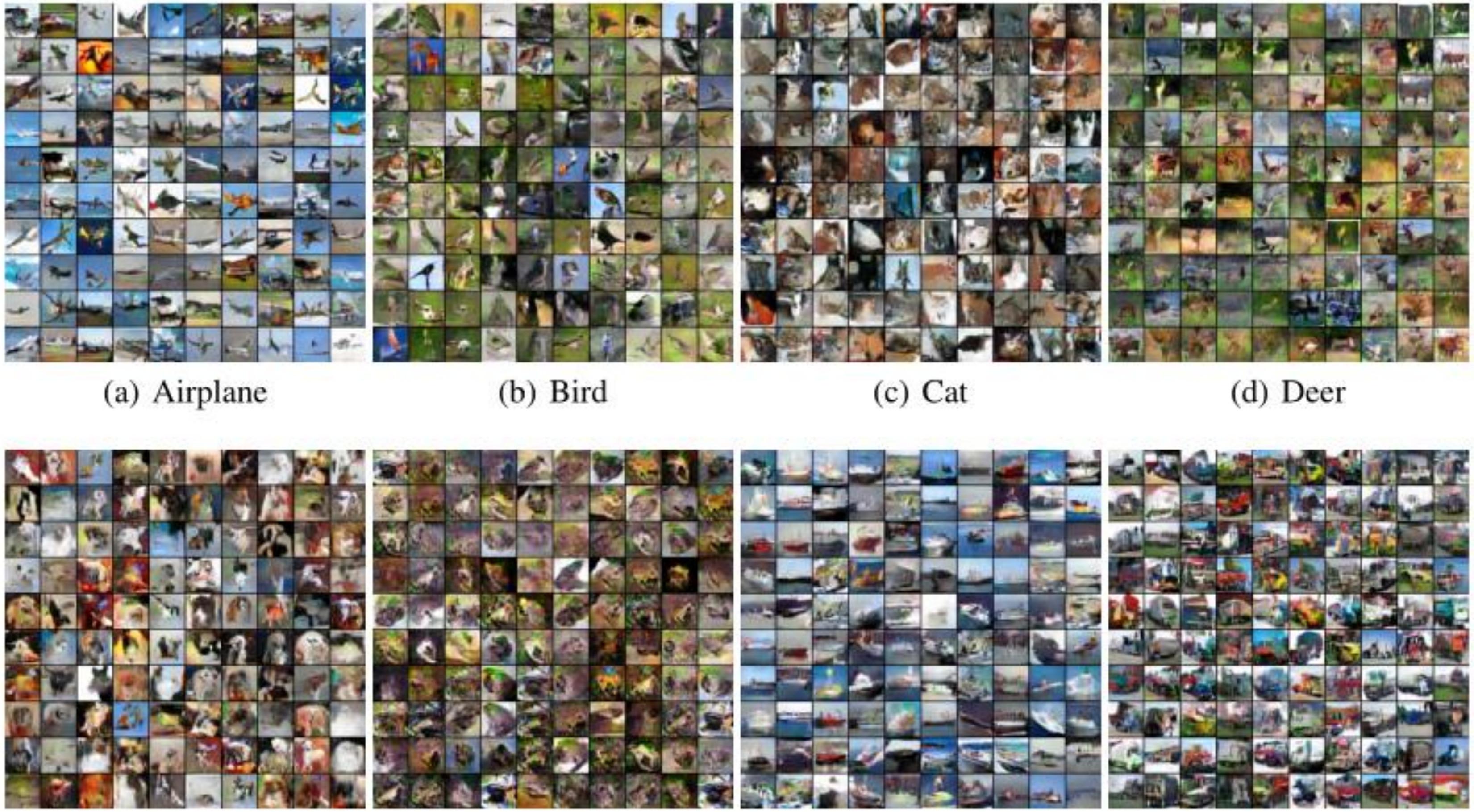
(a) Feature Matching (b) Triple-GAN (c) Automobile (d) Horse

Figure 2: (a-b) Comparison between samples from Improved-GAN trained with feature matching and Triple-GAN on SVHN. (c-d) Samples of Triple-GAN in specific classes on CIFAR10.

(a) Feature Matching (b) Triple-GAN (c) Feature Matching (d) Triple-GAN

Figure 1: (a) and (c): Samples generated from Improved-GAN trained with feature matching on MNIST and CIFAR10 datasets. Strange patterns repeat on CIFAR10. (b) and (d): Samples generated from Triple-GAN.

Experiment-Generation



(a) Airplane (b) Bird (c) Cat (d) Deer

(e) Dog (f) Frog (g) Ship (h) Truck

Figure 2: Samples from Triple-GAN given certain class on CIFAR10.

Experiment-Generation

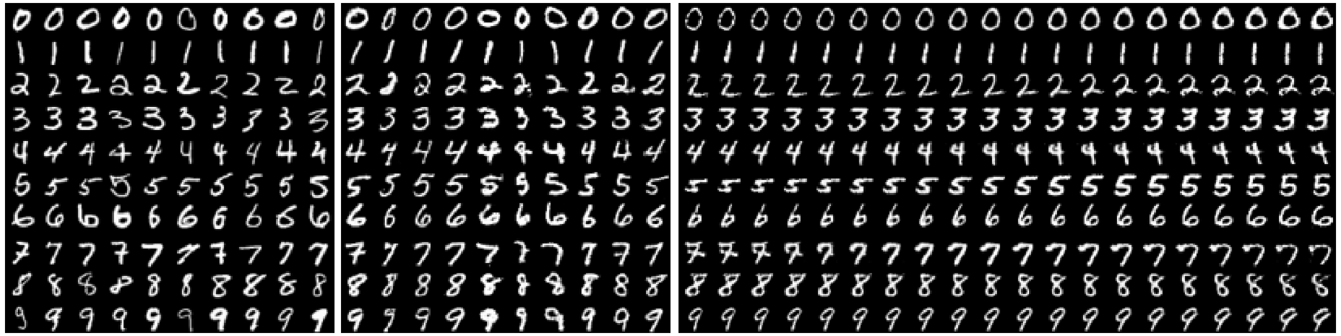
(a) SVHN data (b) SVHN samples (c) CIFAR10 data (d) CIFAR10 samples

Figure 3: (a) and (c) are randomly selected labeled data. (b) and (d) are samples from Triple-GAN, where each row shares the same label and each column shares the same latent variables.

Experiment-Generation

(a) SVHN (b) CIFAR10

Figure 4: Class-conditional latent space interpolation. We first sample two random vectors in the latent space and interpolate linearly from one to another. Then, we map these vectors to the data level given a fixed label for each class. Totally, 20 images are shown for each class. We select two endpoints with clear semantics on CIFAR10 for better illustration.



(a) Data (b) Samples (c) Linear Interpolation

Figure 3: (a): randomly sampled MNIST data; (b) disentanglement of class and style; (c) class-conditional interpolation for Triple-GAN on MNIST.

End

Thanks