Generative Adversarial Networks with Joint Distribution Moment Matching (GAN with JDMM)

Chaomin Shen[1]
Joint work with Yiyin Zhang[1], Hao Feng[1], Yaxin Peng[2], Guixu Zhang[1]

[1]Dept of Computer Science, East China Normal University, Shanghai, China
[2]Dept of Mathematics, Shanghai University, Shanghai, China

Apr. 14, 2018, Nanjing

Generative Adversarial Networks (GAN)

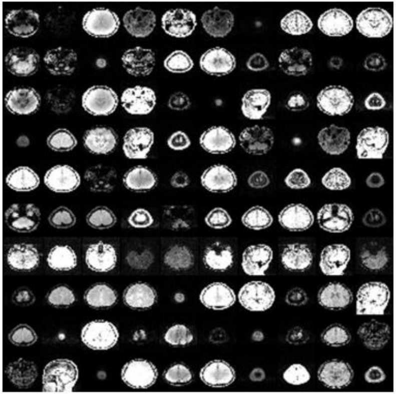
Real MNIST

Generated MNIST

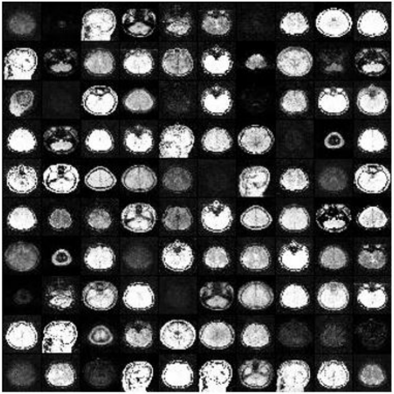
Real CIFAR-10

Generated CIFAR-10

Generative Adversarial Networks (GAN)



Real Brain MRI



Generated Brain MRI

Basic formula of GAN

G: Generator

D: Discriminator

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathbb{P}_g(\boldsymbol{z})} [\log(1 - D(G(\boldsymbol{z})))],$$

Basic formula of GAN

G: Generator

D: Discriminator

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))],$$

- $\boldsymbol{x}$: 真实图像 $\boldsymbol{z}$: 输入网络的噪声

Basic formula of GAN

G: Generator
D: Discriminator

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))],$$

- $\boldsymbol{x}$: 真实图像　　　$\boldsymbol{z}$: 输入网络的噪声
- $D(\boldsymbol{x})$: D判断真实图像$\boldsymbol{x}$是否是真实的概率.

Basic formula of GAN

G: Generator
D: Discriminator

$$\min_{G}\max_{D} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))],$$

- $\boldsymbol{x}$: 真实图像　　　$\boldsymbol{z}$: 输入网络的噪声
- $D(\boldsymbol{x})$: D判断真实图像$\boldsymbol{x}$是否是真实的概率. $D(\boldsymbol{x})$越大越好

Basic formula of GAN

G: Generator
D: Discriminator

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))],$$

- $\boldsymbol{x}$: 真实图像　　　$\boldsymbol{z}$: 输入网络的噪声
- $D(\boldsymbol{x})$: D判断真实图像$\boldsymbol{x}$是否是真实的概率. $D(\boldsymbol{x})$越大越好
- $G(\boldsymbol{z})$: G生成的假图像

Basic formula of GAN

G: Generator
D: Discriminator

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))],$$

- $\boldsymbol{x}$: 真实图像 $\boldsymbol{z}$: 输入网络的噪声
- $D(\boldsymbol{x})$: D判断真实图像$\boldsymbol{x}$是否是真实的概率. $D(\boldsymbol{x})$越大越好
- $G(\boldsymbol{z})$: G生成的假图像 希望能以假乱真, 理想的G应该$D(G(\boldsymbol{z}))$越大越好, i.e., $1-D(G(\boldsymbol{z}))$ 越小越好,

Basic formula of GAN

G: Generator
D: Discriminator

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))],$$

- $\boldsymbol{x}$: 真实图像　　　$\boldsymbol{z}$: 输入网络的噪声
- $D(\boldsymbol{x})$: D判断真实图像$\boldsymbol{x}$是否是真实的概率. $D(\boldsymbol{x})$越大越好
- $G(\boldsymbol{z})$: G生成的假图像　希望能以假乱真, 理想的G应该$D(G(\boldsymbol{z}))$越大越好, i.e., $1-D(G(\boldsymbol{z}))$ 越小越好, 所以$\min G$.

Basic formula of GAN

G: Generator
D: Discriminator

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))],$$

- $\boldsymbol{x}$: 真实图像 $\boldsymbol{z}$: 输入网络的噪声
- $D(\boldsymbol{x})$: D判断真实图像$\boldsymbol{x}$是否是真实的概率. $D(\boldsymbol{x})$越大越好
- $G(\boldsymbol{z})$: G生成的假图像 希望能以假乱真, 理想的G应该$D(G(\boldsymbol{z}))$越大越好, i.e., $1-D(G(\boldsymbol{z}))$ 越小越好, 所以$\min G$.
- $D(\boldsymbol{x})$: 理想的D应该使得 $D(\boldsymbol{x})$大; $D(G(\boldsymbol{z}))$小, 即第2项大,

Basic formula of GAN

G: Generator
D: Discriminator

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))],$$

- $\boldsymbol{x}$: 真实图像　　$\boldsymbol{z}$: 输入网络的噪声
- $D(\boldsymbol{x})$: D判断真实图像$\boldsymbol{x}$是否是真实的概率. $D(\boldsymbol{x})$越大越好
- $G(\boldsymbol{z})$: G生成的假图像　希望能以假乱真, 理想的G应该$D(G(\boldsymbol{z}))$越大越好, i.e., $1-D(G(\boldsymbol{z}))$ 越小越好, 所以$\min G$.
- $D(\boldsymbol{x})$:　理想的D应该使得 $D(\boldsymbol{x})$大; $D(G(\boldsymbol{z}))$小, 即第2项大, 所以$\max D$.

Outline

Generative Adversarial Networks

- A generative network G tries to capture the data distribution.
  Input: random noise $\boldsymbol{z}$ from prior distribution $p(\boldsymbol{z})$ (e.g. normal)
  Output: synthetic data $G(\boldsymbol{z})$.
- An adversarial network D for distinguishing training and generated data.
  Input: $G(\boldsymbol{z})$
  Output: feature indicating whether fake or not
- The objective function of GAN:

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))],$$

  where $\mathbb{P}_r$ is the distribution of the training data
  $\mathbb{P}_g$ is the distribution of the generated data
- Conditional GAN:

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}|y))] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_g(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z}|y)))],$$

  where y is the label.

Variants of GANs

- Traditonal GANs are unsupervised (and also marginal since no label)
- Conditional GANs: generate data with labels in semi-supervised learning
  - **Advantages**: Achieve better performance than traditional GANs

(a)

(b)

  - **Disadvantages**: Generally only minimize the difference between marginal distributions of real data and generated data, neglecting the difference with respect to each class, e.g. [1]

[1] Y. Ren, J. Li, Y. Luo, J. Zhu, "Conditional generative moment-matching networks," NIPS 2016.

Main idea / Novelties

- Minimize the differences of both the marginal and conditional distributions. Therefore, termed as Joint Distribution Moment Matching.
- Use JDMM as a general framework for both unsupervised and semi-supervised generative tasks.

GAN with JDMM

Frequently used notations

Notation	Description
$\boldsymbol{x}, y_r$	training data and its corresponding label
$\tilde{\boldsymbol{x}}, y_g$	generated data and its corresponding label
$\mathbb{P}_r, \mathbb{P}_g$	marginal distribution of $\boldsymbol{x}/\tilde{\boldsymbol{x}}$
$\mathbb{C}_r, \mathbb{C}_g$	conditional distribution of $\boldsymbol{x}/\tilde{\boldsymbol{x}}$
	e.g., $\mathbb{C}(\boldsymbol{x}\vert y)$ or $\mathbb{C}(y\vert\boldsymbol{x})$
$\mathbb{J}_r, \mathbb{J}_g$	joint distribution of $\boldsymbol{x}/\tilde{\boldsymbol{x}}$
N, M	number of real/generated data
C	number of classes

Maximum Mean Discrepancy (MMD)

- Given two sets of samples $X = \{\boldsymbol{x}_i\}_{i=1}^{N}$ and $X' = \{\boldsymbol{x}'_j\}_{j=1}^{M}$ from distributions $\mathbb{P}_X$ and $\mathbb{P}_{X'}$, MMD[1] is an estimator to answer whether $\mathbb{P}_X = \mathbb{P}_{X'}$.

$$\mathsf{MMD}[\mathcal{K}, \mathbb{P}_X, \mathbb{P}_{X'}] := \sup_{f \in \mathcal{K}} (\mathbb{E}_X[f(X)] - \mathbb{E}_{X'}[f(X')]),$$

  where $\mathcal{K}$ is a class of functions in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, with an associated continuous kernel $k(\cdot,\cdot)$.

- In practice, MMD can also be calculated as the squared difference between the empirical kernel mean embeddings:

$$\mathcal{L}^2_{\mathsf{MMD}} = \left\| \frac{1}{N} \sum_{i=1}^{N} \phi(\boldsymbol{x}_i) - \frac{1}{M} \sum_{j=1}^{M} \phi(\tilde{\boldsymbol{x}}_j) \right\|^2,$$

  where ϕ is a feature mapping: $X \to \mathcal{H}$.

[1] A. Gretton, K. Borgwardt, M. Rasch, et al., "A kernel two-sample test," Journal of Machine Learning Research, 13(3):723–773, 2012.

Marginal Distribution Moment Matching

- Computationally expensive to apply MMD directly on the data space.
- MMD can be computed on the features extracted by the discriminator, since the discriminator is a CNN whose output contains features extracted from input images.

Marginal Distribution Moment Matching

$$L_{\mathsf{MMD}}(\mathbb{P}_r, \mathbb{P}_g) = \left\| \frac{1}{N} \sum_{i=1}^{N} \phi(D(\boldsymbol{x}_i)) - \frac{1}{M} \sum_{j=1}^{M} \phi(D(\tilde{\boldsymbol{x}}_j)) \right\|^2$$

where $\phi(\cdot)$ is a feature mapping: $X \to \mathcal{H}$,
$D(\cdot)$ represents the outputs of the discriminator,
$\{\boldsymbol{x}_i\}_{i=1}^{N}$ are samples from the training data distribution,
$\{\tilde{\boldsymbol{x}}_j\}_{j=1}^{M}$ are samples from the generated data distribution.

Conditional Distribution Moment Matching

$$\mathbb{C}_r(y_r = c|\boldsymbol{x}),\ \mathbb{C}_g(y_g = c|\tilde{\boldsymbol{x}}), \qquad \mathbb{C}_r(\boldsymbol{x}|y_r = c),\ \mathbb{C}_g(\tilde{\boldsymbol{x}}|y_g = c)$$

Given labels in the real data, $\mathbb{C}_r(y_r|\boldsymbol{x})$ is known.
We also want to get the labels for the generated data, i.e., $\mathbb{C}_g(y_g|\tilde{\boldsymbol{x}})$

- The conditional distributions $\mathbb{C}_r(y_r|\boldsymbol{x})$ and $\mathbb{C}_g(y_g|\tilde{\boldsymbol{x}})$ should be close as well.
- Matching $\mathbb{C}_r(y_r|\boldsymbol{x})$ and $\mathbb{C}_g(y_g|\tilde{\boldsymbol{x}})$ is nontrivial.
  - The probability density of $\tilde{\boldsymbol{x}}$ is changing through the whole training procedure. Therefore it is impossible to compute $\mathbb{C}_g(y_g|\tilde{\boldsymbol{x}})$.

Conditional Distribution Moment Matching

We employ the nonparametric statistic $\mathbb{C}_r(\boldsymbol{x}|y_r)$ and $\mathbb{C}_g(\tilde{\boldsymbol{x}}|y_g)$ instead

Conditional Distribution Moment Matching *for each class*

$$L_{\text{MMD}}^{(c)}(\mathbb{C}_r^{(c)}, \mathbb{C}_g^{(c)}) = \left\| \frac{1}{N^c} \sum_{i=1}^{N^c} \phi(D(\boldsymbol{x}_i)) - \frac{1}{M^c} \sum_{j=1}^{M^c} \phi(D(\tilde{\boldsymbol{x}}_j)) \right\|^2,$$

where $\mathbb{C}_r^{(c)}$ and $\mathbb{C}_g^{(c)}$: conditional distributions,

$\{\boldsymbol{x}_i : \boldsymbol{x}_i \sim \mathbb{C}_r^{(c)}, y(\boldsymbol{x}_i) = c\}$: set of real samples for class c,

$y(\boldsymbol{x}_i)$: real label,

$\{\tilde{\boldsymbol{x}}_j : \boldsymbol{y}_j \sim \mathbb{C}_g^{(c)}, y(\tilde{\boldsymbol{x}}_j) = c\}$: set of generated samples for class c,

$y(\tilde{\boldsymbol{x}}_j)$: generated label,

N^c and M^c: numbers of real and generated samples for class c.

Joint Distribution Moment Matching

Joint Distribution Moment Matching

$$L(\mathbb{J}_r, \mathbb{J}_g) = L_{\textsf{MMD}}(\mathbb{P}_r, \mathbb{P}_g) + \lambda \sum_{c=1}^{C} L_{\textsf{MMD}}^{(c)}(\mathbb{C}_r^{(c)}, \mathbb{C}_g^{(c)}),$$

where $\mathbb{J}_r$ represents joint distribution of real data

$\mathbb{J}_g$ represents joint distribution of generated data

$\lambda = 0$ indicates the GAN is unsupervised

Framework of GAN with JDMM

A minimax game between the discriminator and generator:

Framework of GAN with JDMM

$$\max_D L(\mathbb{J}_r, \mathbb{J}_g) - \gamma L_C,$$

$$\min_G L(\mathbb{J}_r, \mathbb{J}_g) + \gamma L_C$$

where L_C is the cross entropy $L_C = -\boldsymbol{y}_r \cdot \log(\hat{\boldsymbol{y}}_r) - \boldsymbol{y}_g \cdot \log(\hat{\boldsymbol{y}}_g)$.
$\gamma = 0$ for unsupervised GAN

A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in ICML, pp. 2642-2651, 2017

Framework of GAN with JDMM

We improve the robustness of our model:

Gradient penalty in WGAN[1]

$$GP = \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} [(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2]$$

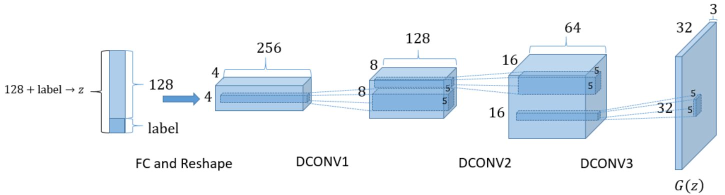
Final framework of GAN with JDMM

$$\max_D L(\mathbb{J}_r, \mathbb{J}_g) - \gamma L_C - \xi GP,$$

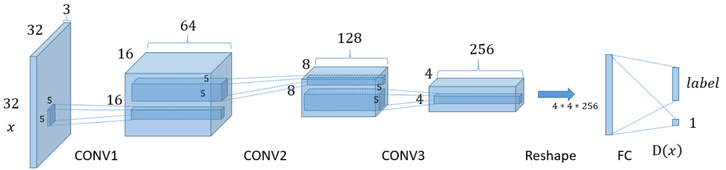
$$\min_G L(\mathbb{J}_r, \mathbb{J}_g) + \gamma L_C$$

[1] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” arXiv preprint arXiv:1701.07875, 2017.

Network structure



Network G



Network D

Implementation

Algorithm 1 GAN with JDMM

Input : n_c: number of iterations for discriminator per generator update
α: learning rate
β_1, β_2: Adam hyperparameters
θ_G and θ_D: Initial generator/discriminator parameters

1. **for** number of training iterations **do**
2.   **for** $t = 1$ to n_c **do**
3.   Sample a minibatch $\boldsymbol{x}$ from $\mathbb{P}_r$ and $\boldsymbol{z}$ from $\mathbb{P}(\boldsymbol{z})$
4.   Generate fake data $G(\boldsymbol{z})$
5.   Compute gradient $g_D = \nabla_D L(\mathbb{J}_r, \mathbb{J}_g) - \gamma \nabla_D L_C - \xi \nabla_D GP$
6. $\theta_D = \text{Adam}(g_D, \alpha, \beta_1, \beta_2)$
7.   **end for**
8.   Compute gradient $g_G = \nabla_G L(\mathbb{J}_r, \mathbb{J}_g) + \gamma \nabla_G L_C$
9. $\theta_G = \text{Adam}(g_G, \alpha, \beta_1, \beta_2)$
10. **end for**

Experimental results (Databases)

MNIST (unsupervised and semi-supervised)

- 50,000 gray-scale images, 10 classes
- Image size: 28×28

CIFAR-10 (semi-supervised)

- 50,000 color images, 10 classes
- Image size: 32×32

Extended Yale Face (semi-supervised)

- 38 classes of gray-scale human faces under different light conditions and poses
- Resize to 32×32

Data provided by 上海市磁共振重点实验室 (unsupervised)

- 3000 images
- Resize to 32×32

手写汉字 (Semi-supervised)

- 300×3750 images
- Image size: 32×32

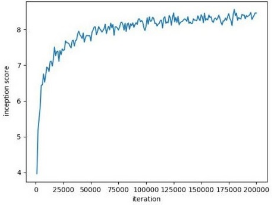
Experimental results (Benchmarks)

WGAN and its variants: One of the best GANs

MMD based GANs: Similar methods

Experimental results (Evaluation rule for CIFAR)

Inception score: a benchmark of measuring sample quality



Method	Score
SteinGAN [Wang and Liu, 2016]	6.35
DCGAN (with labels, in [Radford *et al.*, 2015])	6.58
Improved GAN [Salimans *et al.*, 2016]	8.09±.07
AC-GAN [Odena *et al.*, 2017]	8.25±.07
WGAN-GP ResNet [Gulrajani *et al.*, 2017]	8.42±.10
GAN with JDMM (ours)	**8.59±.10**

Examples of inception scores

Experimental results (MNIST: unsupervised and semi-supervised)

Groundtruth

(a)

(b)

(a) Unsupervised (labels not used); (b) Semi-supervised (labels used)

Experimental results (Comparison with other unsupervised GANs)

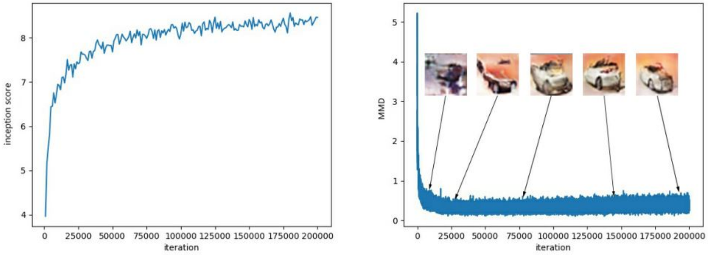
(a) GMMN-C (b) WGAN (c) MMD GAN (d) Ours

Images generated from MNIST in the unsupervised setup

Experimental results (Comparison with other semi-supervised GANs)

Results of WGAN-GP and GAN with JDMM on CIFAR-10 (labels used).
Inception score: (a) 8.42 ± 0.1 (b) **8.59** ± 0.1

Experimental results (Inception score and MMD value)



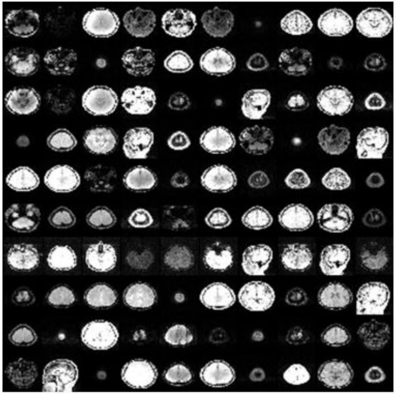
(a) Inception score (b) MMD

Inception score and MMD during the training of GAN with JDMM on CIFAR-10.

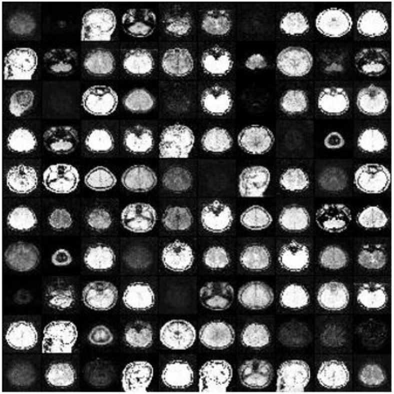
Experimental results (Extended Yale Face: semi-supervised)

Images generated from Extended Yale Face.
Each column represents one particular individual.

Experimental results (MRI: unsupervised)

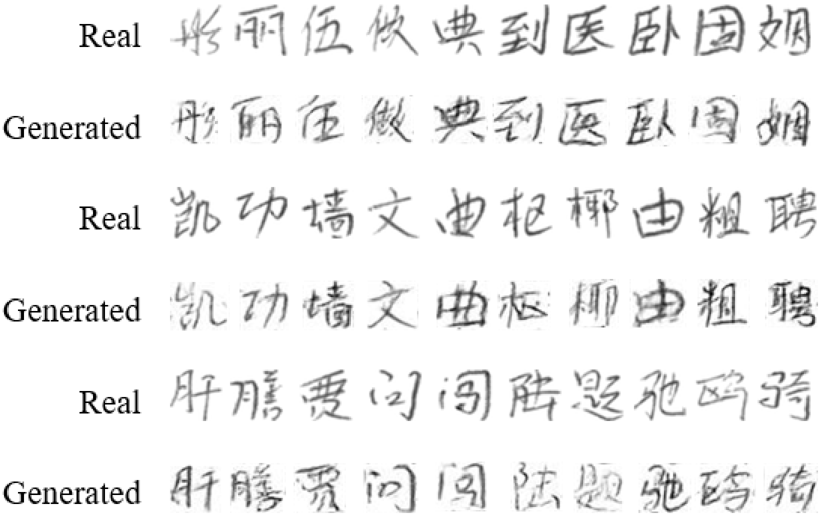


Real Brain MRI



Generated Brain MRI

Experimental results (Chinese characters: supervised)



Conclusion

- Minimize the difference not only of marginal distributions but also of conditional distributions.
- For unsupervised and semi-supervised cases
- Desirable and competitive performance on several databases.

Conclusion

- Minimize the difference not only of marginal distributions but also of conditional distributions.
- For unsupervised and semi-supervised cases
- Desirable and competitive performance on several databases.

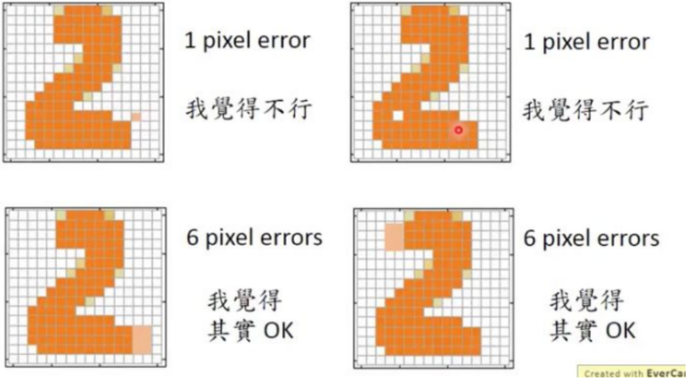
Future Works

- Medical images
  - 512×512
  - Images other than brain images, e.g., CT images of bones

Thanks

cmshen@cs.ecnu.edu.cn

www.umi-china.org.cn


1 pixel error
我覺得不行
1 pixel error
我覺得不行
6 pixel errors
我覺得
其實 OK
6 pixel errors
我覺得
其實 OK
Created with EverCa