

# GAN-EM: GAN Based EM Learning Framework

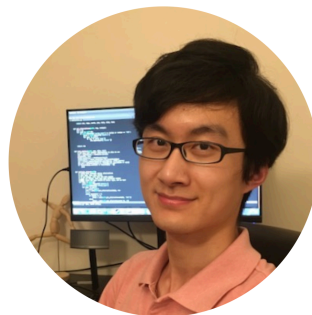
Wentian Zhao<sup>1\*</sup>, Shaojie Wang<sup>1\*</sup>, Zhihuai Xie<sup>2</sup>, Jing Shi<sup>1</sup>, and  
**Chenliang Xu<sup>1</sup>**

<sup>1</sup>University of Rochester

<sup>2</sup>Tsinghua University



Wentian Zhao



Shaojie Wang

\*The two authors contributed equally.

# Introduction

- **Expectation maximization (EM)** algorithm is to find the maximum likelihood solution for models having latent variables.
  - A typical example is the Gaussian mixture model (GMM), which requires a Gaussian assumption. However, natural images are highly non-Gaussian.
- **Generative adversarial network (GAN)** has been proved to be powerful in learning data distribution.
  - Previous work has shown that GAN can be modified to perform maximum likelihood estimation (MLE).
- We propose **GAN-EM**: a GAN based EM learning framework that can maximize the likelihood of images and estimate the latent variables.
  - It can perform **image clustering**, **semi-supervised classification**, and **dimensionality reduction**, etc.

## Related Works – Deep EM

- **Neural-EM (N-EM) [Greff et al. 2017]** trains the parameters of EM using a NN, which derives a differentiable clustering model, and is used for unsupervised segmentation. It can cluster constituent objects.
  - Neural-EM aims to learn the param. of the mixture models using NNs.
  - In GAN-EM, the goal is to learn the weights of neural networks, i.e., the generator, whose output is the generated samples based on the currently learned mixture models.
  - Therefore, the NN plays a different role between these two models.
- **Generative mixture of networks (GMN) [Banijamali et al. 2017]** first uses K-means to obtain prior knowledge of the dataset, and then treats each network as a cluster.
- **Variational deep embedding (VaDE) [Jiang et al. 2017]** combines GMM with variational autoencoder (VAE), which keeps the Gaussian assumption. In M-step, VaDE maximizes the lower bound on the log-likelihood given by Jensen inequality. In E-step, a neural network is used to model the mapping from data to class assignment.

## Related Works – GAN Clustering

- **GANMM [Yu and Zhou 2018]:** Similar to GMM, they fit GANs into the GMM (GANMM). In GANMM, hard label assignment strategy limits the model to K-means, which is an extreme case of EM for mixture model.
  - First, we use soft label assignment, rather than the hard assignment in GANMM. To the best of our knowledge, our work is the first to achieve general EM process using GAN.
  - Second, we use only one GAN, rather than  $K$  GANs where  $K$  is the number of clusters. Our model is scalable with respect of  $K$ .
  - Experimental results show that our GAN-EM outperforms GANMM by a big margin.
- **Other GAN-based methods:** Categorical generative adversarial networks (CatGAN) [Springenberg 2015], InfoGAN [Spurr et al. 2017], adversarial autoencoder [Makhzani et al. 2015], feature matching GAN [Salimans et al. 2016] and pixelGAN autoencoder [Makhzani and Frey 2017]

# Methodology

- M-step:
  - Update parameters by training GAN
  - Loss: label assignment from E-step

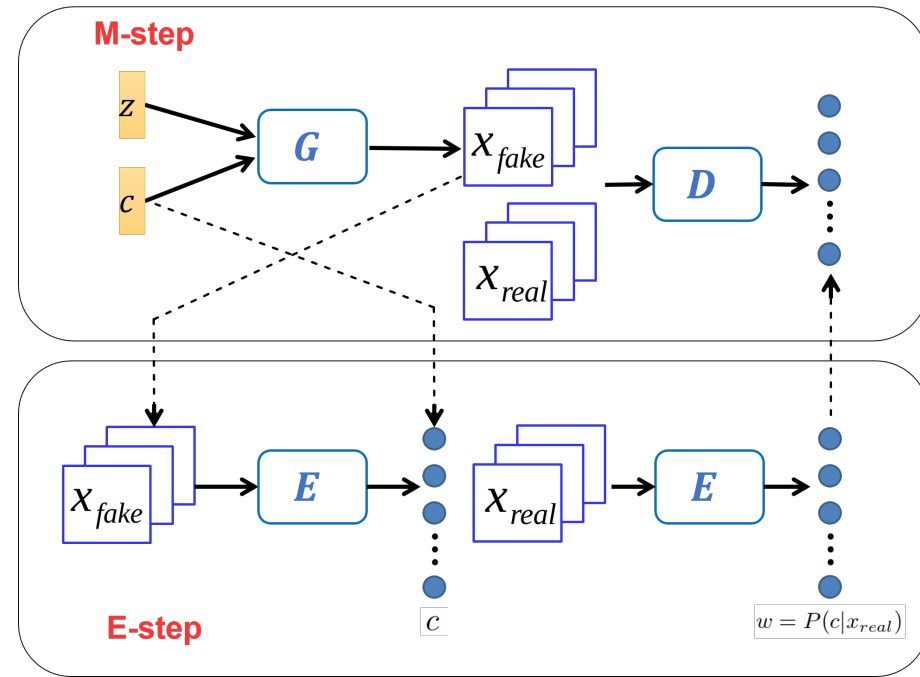
$$L_G = -\frac{1}{2} \mathbb{E}_{c \sim U} \mathbb{E}_{z \sim \mathcal{N}} \exp\{\sigma^{-1}[D_c(G(z, c))]\}$$

$$L_D = \mathbb{E}_{c \sim U} \mathbb{E}_{z \sim \mathcal{N}} \sum_{i=1}^K \log(1 - D_i(G(z, c)))$$

$$+ \mathbb{E}_{x \sim P_r} \sum_{i=1}^K w_i \log(D_i(x))$$

- E-step:
  - Sample from G, train E-net
  - Update label assignment (soft)

$$L_E = \mathbb{E}_{z \sim \mathcal{N}} \mathbb{E}_{c \sim \phi} \text{CE}\{E(G(z, c)), u\}$$



- Finally, the E-net is used for prediction.

# Methodology

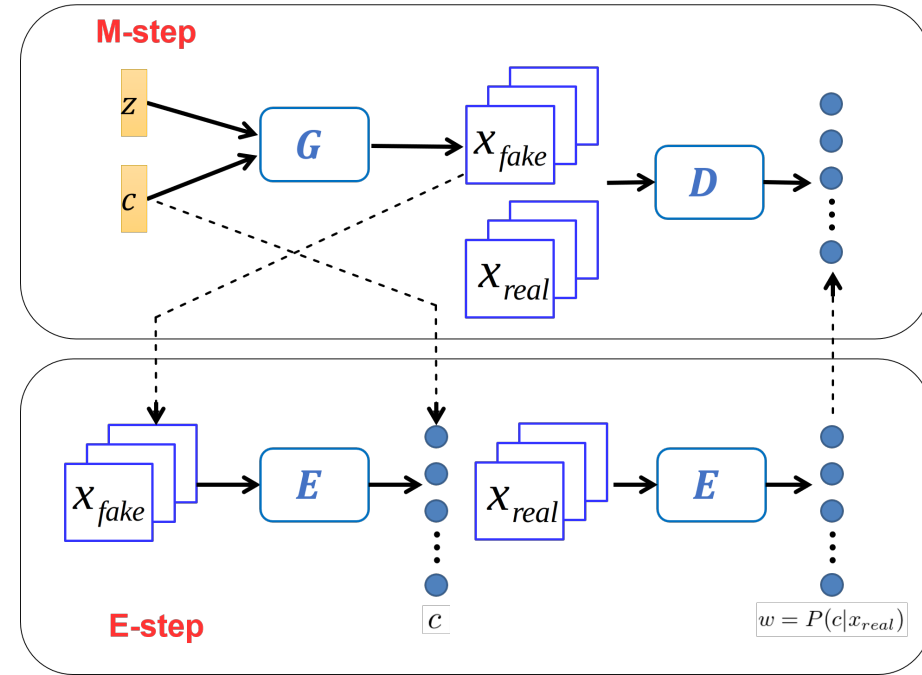
---

## Algorithm 1 GAN-EM

---

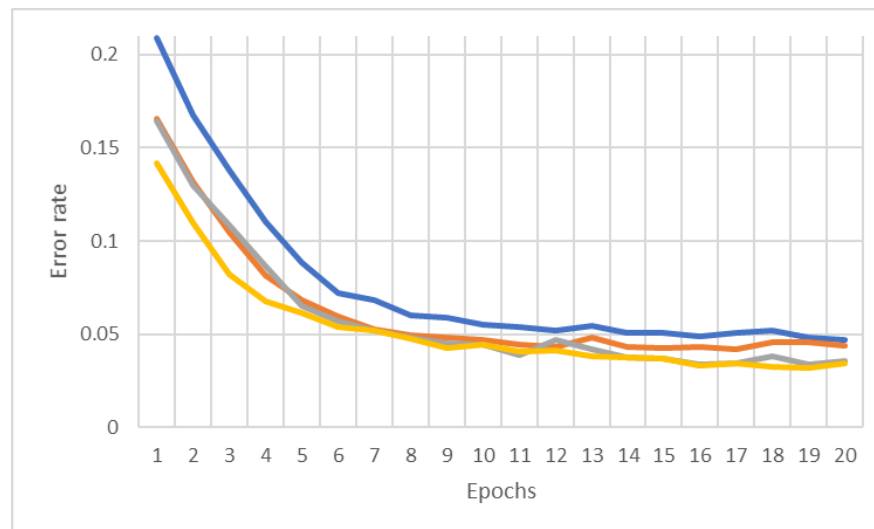
1: **Initialization:**  $w_i = 1/K$  for  $i = 1 \dots K$   
 2: **for**  $iteration = 1 \dots n$  **do**  
 3:   **update**  $\phi$ : ▷ M-step  
 4:      $\phi_i = N_i/N$  for  $i = 1 \dots K$   
 5:   **update**  $\psi$ :  
 6:     **for**  $epoch = 1 \dots p$  **do**  
 7:       train GAN:  $\min_{\psi_G} L_G, \max_{\psi_D} L_D$   
 8:     **end for**  
 9:     **for**  $epoch = 1 \dots q$  **do** ▷ E-step  
 10:       Sample a batch of  $c \sim \phi$ , and obtain  $G(z, c)$   
 11:       train E-net:  $\min_{\eta} L_E$  ( $\eta$ : weights of E-net<sup>2</sup>)  
 12:     **end for**  
 13:     **update label assignment:**  $w = E(x_{real})$   
 14: **end for**  
 15: **Predict:**  $w = E(x_{real})$

---



# Methodology

- In the ideal case, where NN researches global minima, the convergence can be guaranteed.
- The solution of NN optimization is local minima instead of global minima, which cannot guarantee the likelihood to increase monotonically.
- However, the experiment results show that the error rate for clustering decreases smoothly and converges within 20 epochs in most cases .



# Experiments

- **Unsupervised Clustering:** we perform unsupervised clustering on **MNIST** and **CelebA** datasets.
- Evaluate the error rate:
  - Used in most other clustering methods

$$Err = 1 - \max_{m \in \mathcal{M}} \frac{\sum_{i=1}^N \mathbb{1}\{l_i = m(c_i)\}}{N}$$

Ground truth label      Predicted label for the  $i^{\text{th}}$  sample

One-to-one mapping from GT to predicted labels of all samples in the cluster

Number of all samples



# Experiments

- **Unsupervised Clustering:** GAN-EM achieves SOTA results on **MNIST** with 10, 20 and 30 clusters and on **CelebA**.

	MNIST (Unsupervised)	MNIST (100 labels)	MNIST (1000 labels)	SVHN (1000 labels)	CelebA (Unsupervised)	CelebA (100 labels)
K-means →	46.51 <sup>1</sup>	-	-	-	-	-
GMM →	32.61(±0.06) <sup>1</sup>	-	-	-	-	-
DEC →	15.7 <sup>1</sup>	-	-	-	-	-
VAE	-	3.33(±0.14)	2.40(±0.02)	36.02(±0.10)	-	45.38
AAE →	4.10(± 1.13) <sup>3</sup>	1.90(±0.10)	1.60(±0.08)	17.70(±0.30)	42.88	31.03
CatGAN →	4.27 <sup>2</sup>	1.91(±0.10)	1.73(±0.28)	-	44.57	34.78
InfoGAN	5.00 <sup>4</sup>	-	-	-	-	-
Improved GAN	-	<b>0.93(±0.06)</b>	-	8.11(±1.30)	-	-
VaDE →	5.54 <sup>1</sup>	-	-	-	43.64	-
PixelGAN	5.27(±1.81) <sup>3</sup>	1.08(±0.15)	-	6.96(±0.55)	44.27	32.54
GANMM →	35.70(±0.45) <sup>1</sup>	-	-	-	49.32	-
<b>GAN-EM</b> →	4.20(± 0.51) <sup>1</sup>	1.09(± 0.18)	<b>1.03(± 0.15)</b>	<b>6.05(±0.26)</b>	<b>42.09</b>	<b>28.82</b>
	<b>4.04(± 0.42)<sup>2</sup></b>	-	-	-	-	-
	<b>3.97(± 0.37)<sup>3</sup></b>	-	-	-	-	-

Table 1: Experiment results of different models. <sup>1</sup>Clustering with  $K = 10$ . <sup>2</sup> $K = 20$ . <sup>3</sup> $K = 30$ . <sup>4</sup>Cluster number not specified.

# Experiments

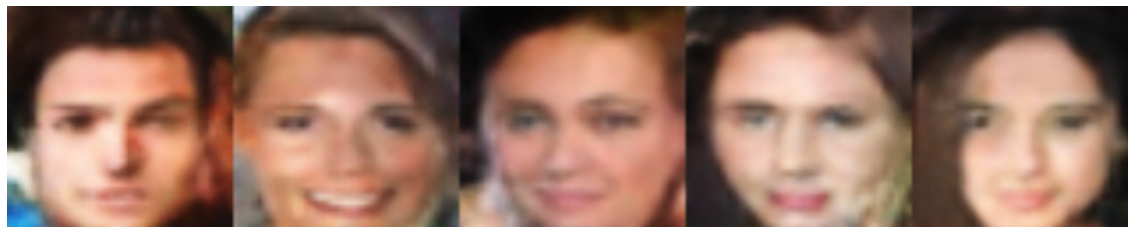
- **Unsupervised Clustering:** we perform unsupervised clustering on **MNIST** and CelebA datasets. **Results from G.**



# Experiments

- **Unsupervised Clustering:** we perform unsupervised clustering on MNIST and **CelebA** datasets. **Results from G.**

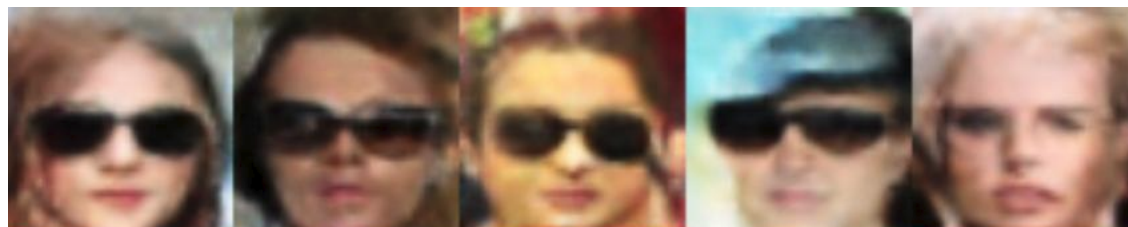
“no hat”



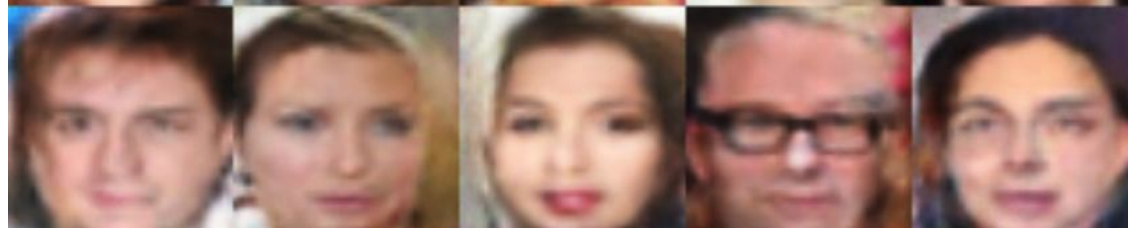
“hat”



“glass”



“no glass”



# Experiments

- **Semi-Supervised Classification:** add supervision to the training of the E-net

	MNIST (Unsupervised)	MNIST (100 labels)	MNIST (1000 labels)	SVHN (1000 labels)	CelebA (Unsupervised)	CelebA (100 labels)
K-means	46.51 <sup>1</sup>	-	-	-	-	-
GMM	32.61(±0.06) <sup>1</sup>	-	-	-	-	-
DEC	15.7 <sup>1</sup>	-	-	-	-	-
VAE	-	3.33(±0.14)	2.40(±0.02)	36.02(±0.10)	-	45.38
AAE	4.10(± 1.13) <sup>3</sup>	1.90(±0.10)	1.60(±0.08)	17.70(±0.30)	42.88	31.03
CatGAN	4.27 <sup>2</sup>	1.91(±0.10)	1.73(±0.28)	-	44.57	34.78
InfoGAN	5.00 <sup>4</sup>	-	-	-	-	-
Improved GAN	-	<b>0.93(±0.06)</b>	-	8.11(±1.30)	-	-
VaDE	5.54 <sup>1</sup>	-	-	-	43.64	-
PixelGAN	5.27(±1.81) <sup>3</sup>	1.08(±0.15)	-	6.96(±0.55)	44.27	32.54
GANMM	35.70(±0.45) <sup>1</sup>	-	-	-	49.32	-
<b>GAN-EM</b>	4.20(± 0.51) <sup>1</sup>	1.09(± 0.18)	<b>1.03(± 0.15)</b>	<b>6.05(±0.26)</b>	<b>42.09</b>	<b>28.82</b>
	<b>4.04(± 0.42)<sup>2</sup></b>	-	-	-	-	-
	<b>3.97(± 0.37)<sup>3</sup></b>	-	-	-	-	-

Table 1: Experiment results of different models. <sup>1</sup>Clustering with  $K = 10$ . <sup>2</sup> $K = 20$ . <sup>3</sup> $K = 30$ . <sup>4</sup>Cluster number not specified.

# Experiments

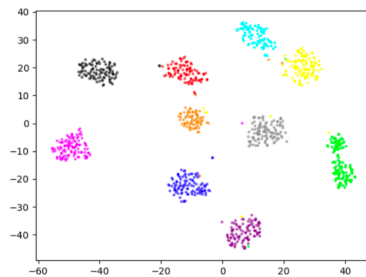
- **Semi-Supervised Classification:** adding supervision to the training of the E-net. **Results from G.**



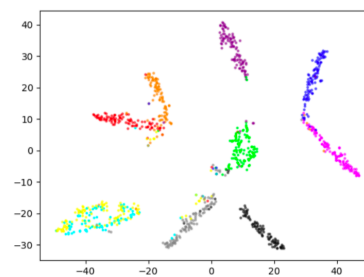
SVHN  
(1000 labels)

# Experiments

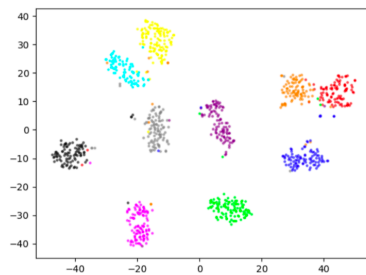
- **Dimensionality Reduction:** add a new layer to the E-net before the output layer
  - After convergence of training, we take the output of this newly added layer as extracted features.



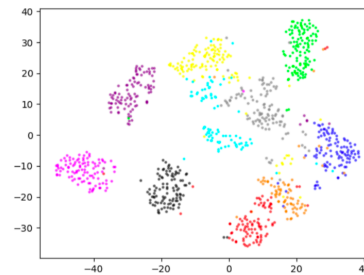
(a) Supervised 1000D



(b) Unsupervised 2D



(c) Unsupervised 100D



(d) Unsupervised 1000D

Figure 3: Representation of unsupervised dimensionality reduction on MNIST. Each color denotes one class of digit.

# Conclusion

- Achieve general EM process using GAN.
- The new GAN-EM can perform clustering, semi-supervised classification, dimensionality reduction, etc.
- It can also be embedded by many other GAN variants and better results are expected.



- Try it yourself! (code)

## Thank you! Questions?

**Acknowledgements:** This work was supported in part by NSF IIS 1741472, IIS 1813709, and CHE 1764415. This article solely reflects the opinions and conclusions of its authors and not the funding agents.