Generative Deep-Neural-Network Mixture Modeling with Semi-Supervised MinMax+EM Learning

Nilay Pande

Suyash P. Awate

Computer Science & Engineering (CSE) Department,
Indian Institute of Technology (IIT) Bombay

Outline

- Introduction
- Our Approach
- Experiments & Results
- Conclusion
- References

Introduction

Non-linear Generative Mixture Modeling :

- Useful way to model high dimensional data distributions.
- Has various applications in the field of image analysis such as clustering, interpolation or data generation.
- We propose a novel statistical framework for a DNN-based mixture model (DNN-MM) using a generator, an encoder and a discriminator.

Introduction

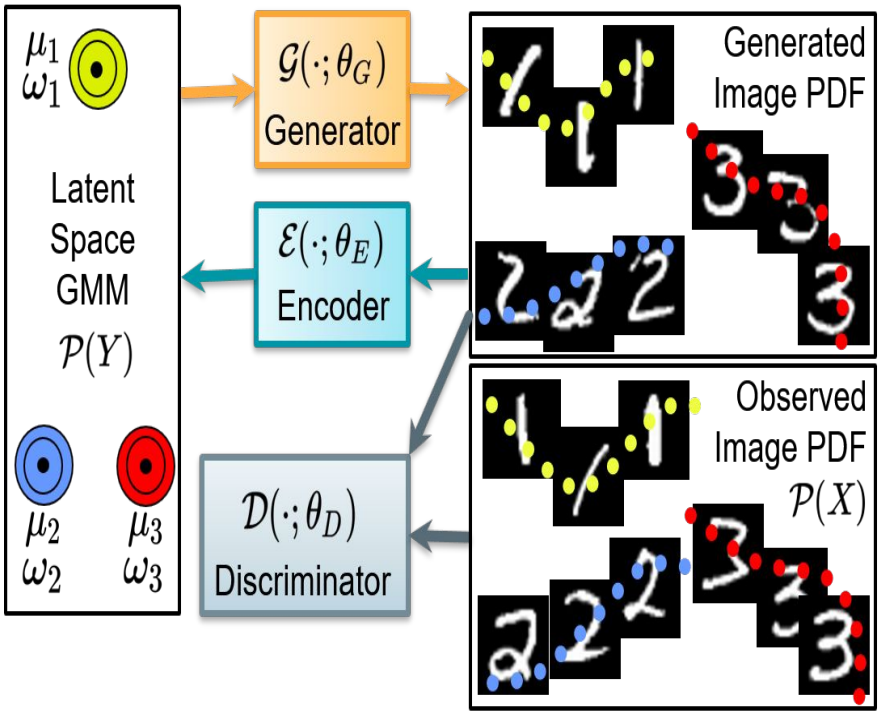
MinMax learning + EM

- Propose a novel data-likelihood term relying on a well regularized/constrained GMM in the latent space along with a prior term on the DNN weights.

- Propose a novel learning formulation by combining minmax learning with EM-based learning, termed MinMax+EM, leveraging a variational lower bound that analytically guarantees tightness to the log-likelihood of the data

- Finally, we extend our model to the semi-supervised setting, where the labels are available for a fraction of the dataset

Our Approach

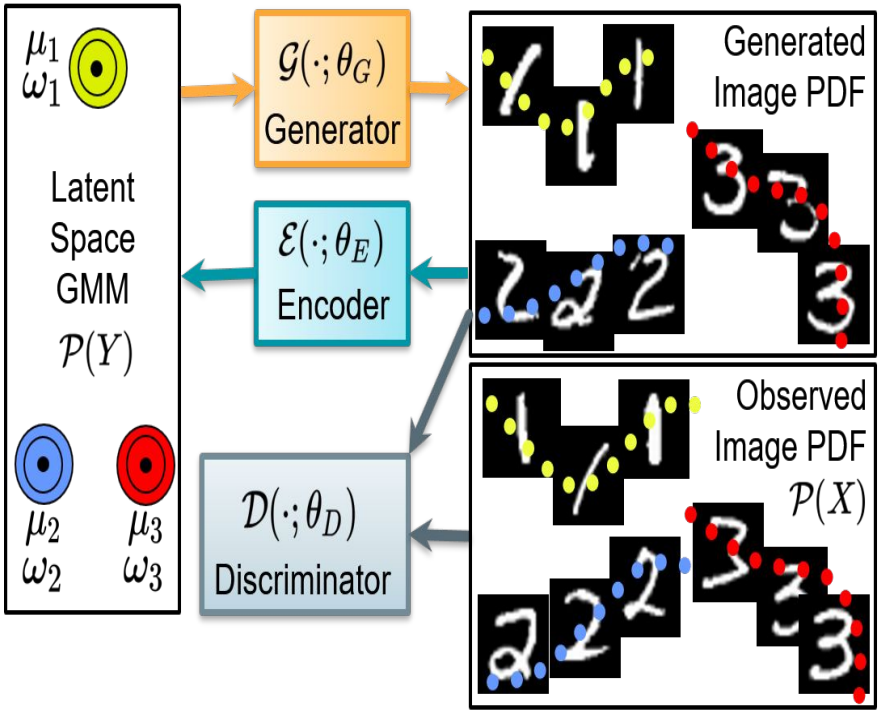


Generator Modelling: DNN-based generator, $\mathcal{G}(\cdot;\theta_G)$ parameterized by DNN weights θ_G to generate images belonging to the real data distribution P(X) through the nonlinear transformation on a random vector Y having a known PDF P(Y)

Encoder Modelling: Encoder mapping $\mathcal{E}(\cdot;\theta_E)$, parameterized by DNN weights θ_E, that maps images X to the latent space Y

Discriminator modelling: Model a mapping, $\mathcal{D}(\cdot;\theta_D)$ parameterized by DNN weights θ_D, such that $\mathcal{D}(X';\theta_D)$ gives the probability of image X' being drawn from the PDF P(X) of real-world images.

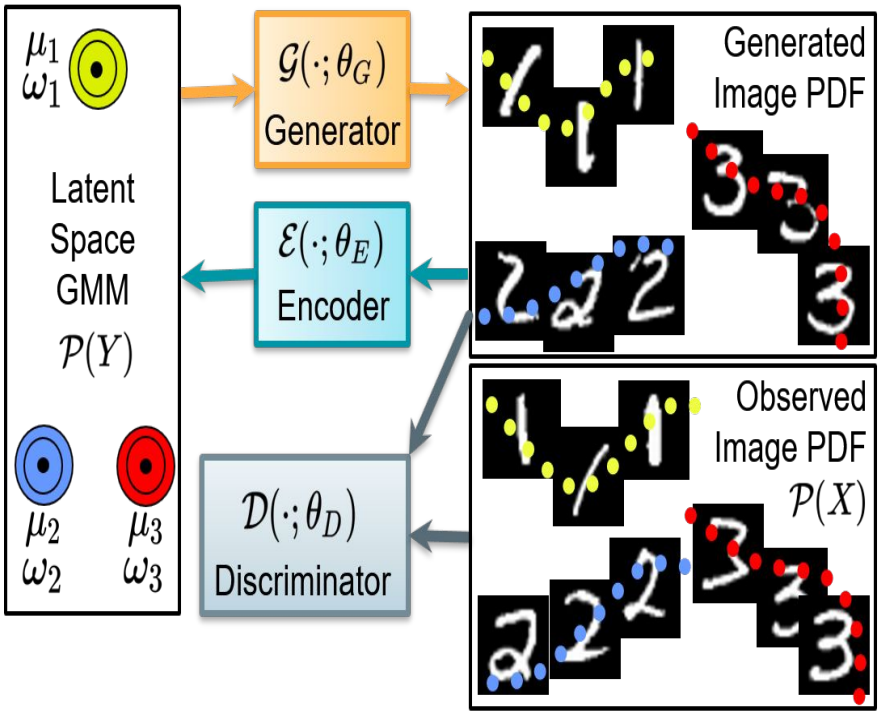
Our Approach



Latent space modelling:

- Latent-space PDF P(Y) as a mixture of K (fixed) Gaussians in latent space
- Covariance being the identity matrix I
- Mixture weights ω_k (learnable)
- Let Z be a hidden categorical random variable indicating the mixture component to which image X belongs and let Z take integer values within [1, K]. Thus, the prior becomes, $P(Z = k) = \omega_k$, where $\sum_{k=1}^{K} \omega_k = 1$

Our Approach - Probability Modelling



Likelihood for an image X given the prior weights ω_k :

$$P(X|\theta_E,\omega) := \sum_{k=1}^{K} \omega_k P(X|Z=k,\theta_E). \quad (1)$$

Probability density for image X drawn from cluster k :

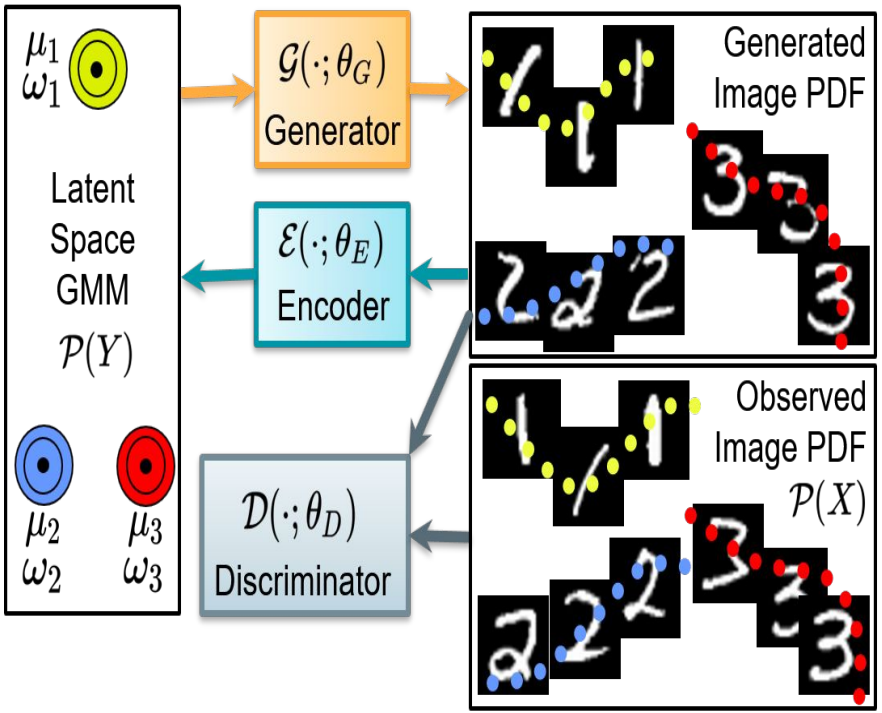
$$P(X|Z=k,\theta_E) := \mathcal{N}(\mathcal{E}(X;\theta_E);\mu_k,\mathbf{I}). \quad (2)$$

Calculating membership of image X to cluster k and thus, log-likelihood:

$$P(Z=k|X,\theta_E,\omega) = \frac{\omega_k \mathcal{N}(\mathcal{E}(X;\theta_E);\mu_k,\mathbf{I})}{\sum_{k'=1}^{K} \omega_{k'} \mathcal{N}(\mathcal{E}(X;\theta_E);\mu_{k'},\mathbf{I})}. \quad (3)$$

$$E_{P(X)} \log \left[\sum_{k=1}^{K} \omega_k \mathcal{N}(\mathcal{E}(X;\theta_E);\mu_k,\mathbf{I}) \right]. \quad (4)$$

Our Approach - Probability Modelling



Consistency Prior on Generator + Encoder:

To ensure that Encoder mappings and Generator mappings are inverses of each other, we propose a log-prior $\log P(\theta_G, \theta_E)$ -

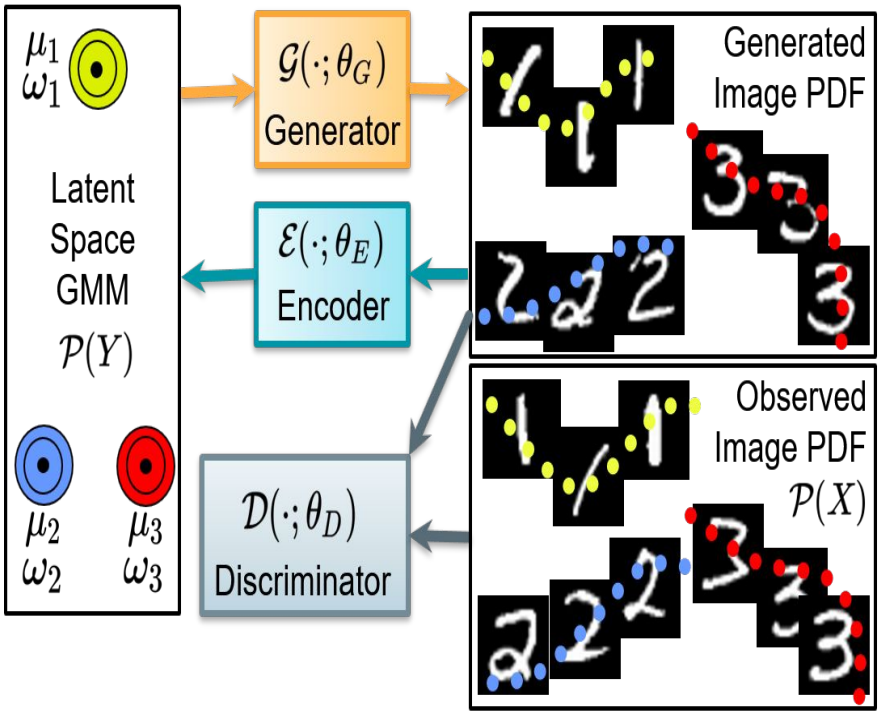
$$E_{P(Y)}[-\|Y - \mathcal{E}(\mathcal{G}(Y;\theta_G);\theta_E)\|_2^2] \tag{5}$$

$$= \sum_{k=1}^{K} \omega_k E_{Y_k \sim \mathcal{N}(\mu_k, \mathbf{I})}[-\|Y_k - \mathcal{E}(\mathcal{G}(Y_k;\theta_G);\theta_E)\|_2^2]. \tag{6}$$

GAN loss terms:

$$E_{P(X)}[-\log \mathcal{D}(X;\theta_D)] + E_{P(Y)}[\log \mathcal{D}(\mathcal{G}(Y;\theta_G);\theta_D)] \tag{7}$$
$$= E_{P(X)}[-\log \mathcal{D}(X;\theta_D)]$$
$$+ \sum_{k=1}^{K} \omega_k E_{Y_k \sim \mathcal{N}(\mu_k, \mathbf{I})}[\log \mathcal{D}(\mathcal{G}(Y_k;\theta_G);\theta_D)], \tag{8}$$

Our Approach - EM lower bound



Optimal lower bound on the log-likelihood:

In the E step, we simplify the log-likelihood through its optimal lower bound as follows. Consider iteration t, with current parameter estimates $\{\theta_G^t, \theta_E^t, \theta_D^t, \omega^t\}$. The E step then designs the function

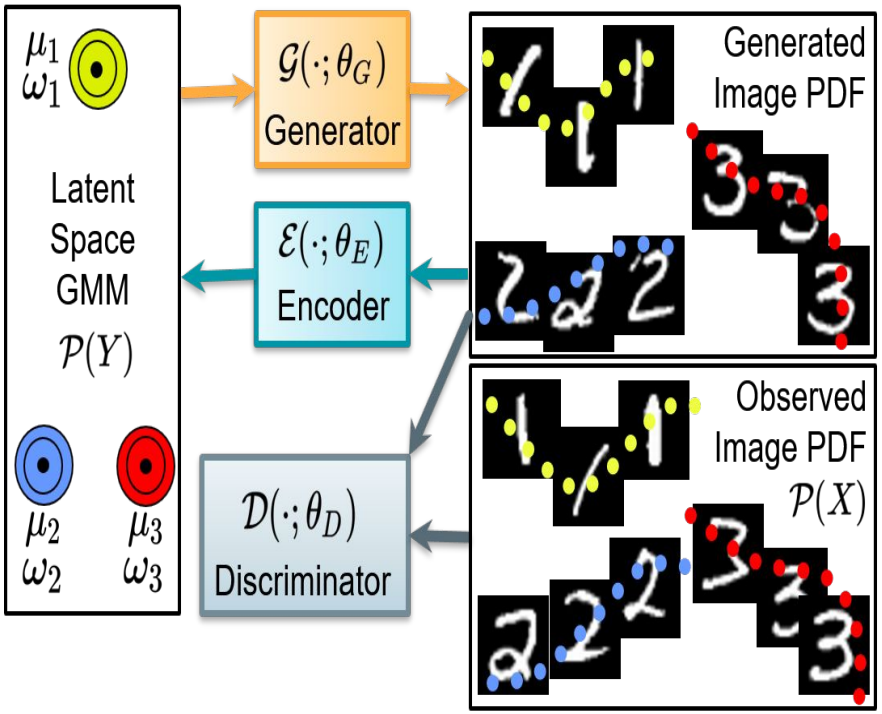
$$Q(\theta_E, \omega; \theta_E^t, \omega^t)$$

$$:= E_{P(X)} E_{P(Z|X,\theta_E^t,\omega^t)}[\log P(X, Z|\theta_E, \omega)] \tag{10}$$

$$= E_{P(X)}\left[\sum_{k=1}^{K} P(Z=k|X,\theta_E^t,\omega^t)\log P(X|Z=k,\theta_E,\omega)\right]$$

$$+ E_{P(X)}\left[\sum_{k=1}^{K} P(Z=k|X,\theta_E^t,\omega^t)\log \omega_k\right], \tag{11}$$

Our Approach - Unsupervised Learning

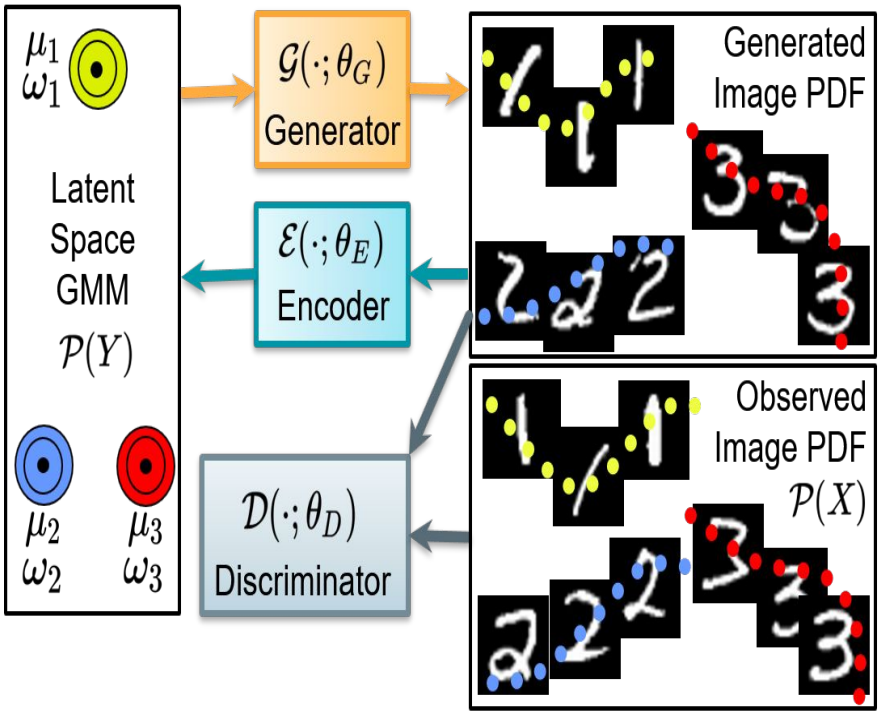


Final objective function at time step t:

$$\min_{\theta_D} \max_{\omega,\theta_G,\theta_E} \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}^{t} \left(\log \omega_k + \log \mathcal{N}(\mathcal{E}(x_n;\theta_E);\mu_k,\mathbf{I})\right)$$
$$-\lambda_1 \sum_{k=1}^{K} \omega_k \sum_{s=1}^{S} \|y_k^s - \mathcal{E}(\mathcal{G}(y_k^s;\theta_G);\theta_E)\|_2^2$$
$$-\lambda_2 \sum_{n=1}^{N} \log \mathcal{D}(x_n;\theta_D)$$
$$+\lambda_2 \sum_{k=1}^{K} \omega_k \sum_{s=1}^{S} \log \mathcal{D}(\mathcal{G}(y_k^s;\theta_G);\theta_D). \qquad (12)$$

Where γ_{nk}^{t} represents posterior membership of data point x_n to the kth cluster based on parameter estimates at time step t

Our Approach - Semi-Supervised Learning



Have a small set of images $\{\widetilde{X}_m\}_{m=1}^{M}$, with cluster labels $\{\widetilde{Z}_m \in [1, K]\}_{m=1}^{M}$ and consider the membership function for these images to be crisp

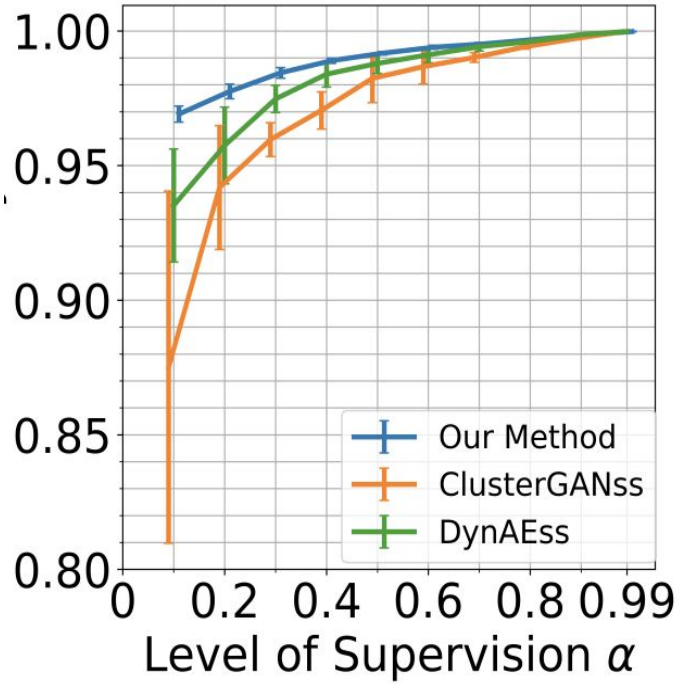
$$\min_{\theta_D} \max_{\omega,\theta_G,\theta_E} \sum_{m=1}^{M}\sum_{k=1}^{K} \mathcal{I}(\widetilde{Z}_m, k)\,(\log\omega_k + \log\mathcal{N}(\mathcal{E}(\widetilde{x}_m;\theta_E);\mu_k,\mathbf{I})) + \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}^{t}\,(\log\omega_k + \log\mathcal{N}(\mathcal{E}(x_n;\theta_E);\mu_k,\mathbf{I})) - \lambda_1 \sum_{k=1}^{K}\omega_k \sum_{s=1}^{S} \|y_k^s - \mathcal{E}(\mathcal{G}(y_k^s;\theta_G);\theta_E)\|_2^2 - \lambda_2 \sum_{n=1}^{N} \log\mathcal{D}(x_n;\theta_D) - \lambda_2 \sum_{m=1}^{M} \log\mathcal{D}(\widetilde{x}_m;\theta_D) + \lambda_2 \sum_{k=1}^{K}\omega_k \sum_{s=1}^{S} \log\mathcal{D}(\mathcal{G}(y_k^s;\theta_G);\theta_D), \quad (13)$$

Here, $\mathcal{I}(\widetilde{Z}_m, k)$ is the indicator function that takes a value of 1 when $\widetilde{Z}_m = k$ and takes a value of 0 otherwise

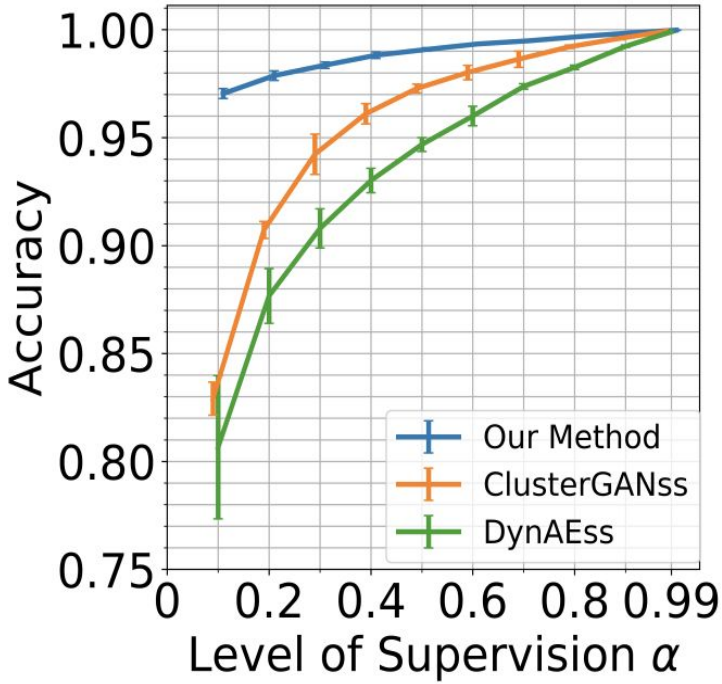
Experiments & Results

- **Datasets & Metrics:**
  - MNIST, CIFAR10 (Noisy), CelebA (Noisy)
  - Accuracy, ARI, NMI
- **Baselines:**
  - ClusterGANss:
    - Semi-supervised version of ClusterGAN with an additional loss term penalizing the cross entropy between its encoder-estimated encodings and the true one-hot encodings for the labelled subset of training set
  - DynAEss:
    - Semi-supervised version of DynAE with a similar loss term

MNIST Dataset: 5 & 7 classes, 1000 images of each digit

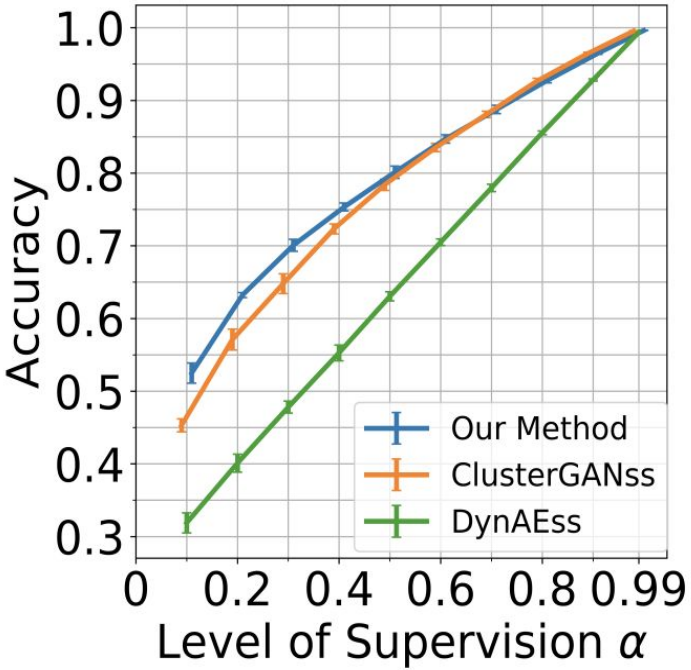


(a1) Accuracy: 5 clusters

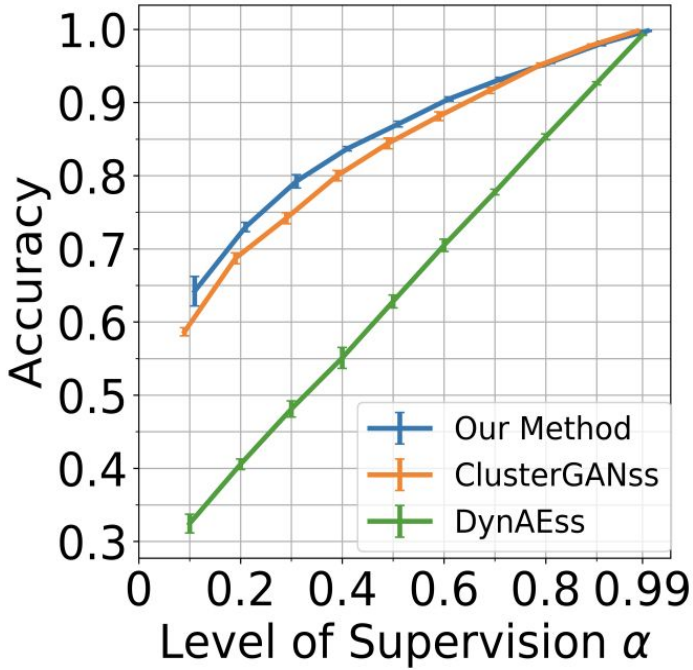


(a2) Accuracy: 7 clusters

CIFAR10 & CelebA Dataset: 5 classes, 1000 images of each class



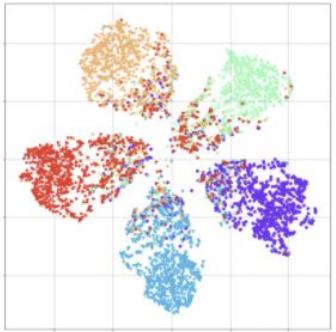
(a1) Accuracy: CIFAR10



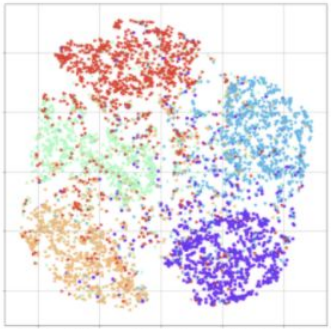
(a2) Accuracy: CelebA

t-SNE visualizations for latent space PDFs at 0.5 supervision

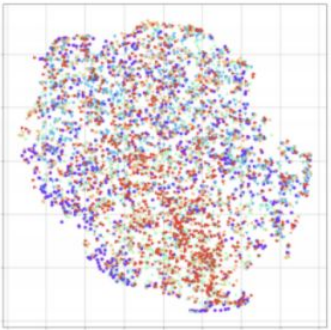
CIFAR10 (5 classes):



(a1) Ours: CIFAR10

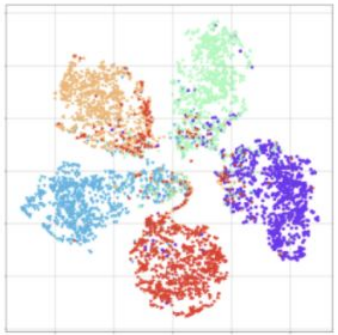


(b1) ClusterGANss: CIFAR10

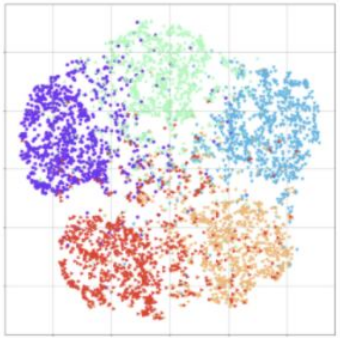


(c1) DynAEss: CIFAR10

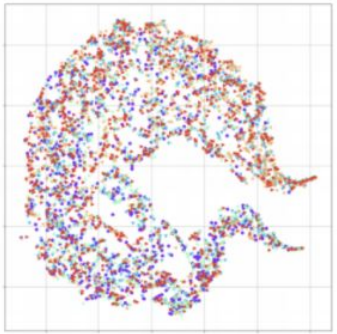
CelebA (5 classes):



(a2) Ours: CelebA



(b2) ClusterGANss: CelebA



(c2) DynAEss: CelebA

Conclusion

- Our GMM-based data-likelihood maximizing formulation leads to statistically significantly better performance than ClusterGANss and DynAEss, especially at smaller levels of supervision α, indicating improved robustness to noise, for all the datasets.
- Unlike VAE-based methods, our min-max learning increases the data likelihood using a tight variational lower bound using EM
- Results on three real-world image datasets demonstrate the benefits of our compact modeling and learning formulation over the state of the art for nonlinear generative image (mixture) modeling and image clustering

References

- [1] - S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, “ClusterGAN: Latent space clustering in generative adversarial networks,” in AAAI Conf. Artific. Intell., 2019, pp. 4610–7.
- [2] - K. Ghasedi, X. Wang, C. Deng, and H. Huang, “Balanced self-paced learning for generative adversarial clustering network,” in IEEE Comp. Vis. Patter. Recog., 2019, pp. 4386–4395.
- [3] - N. Mrabah, N. Khan, and R. Ksantini, “Deep clustering with a dynamic autoencoder,” arxiv.org/abs/1901.07752, 2019.
- [4] - X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,” in Adv. Neur. Info. Proc. Sys., 2016, p. 21808.
- [5] Y. Yu and W.-J. Zhou, “Mixture of GANs for clustering,” in Int. J. Conf. Artific. Intell., 2018, p. 304753

Thank you