2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

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

Nilay Pande and Suyash P. Awate
Computer Science and Engineering Department,
Indian Institute of Technology (IIT) Bombay, Mumbai 400076. India.

***Abstract*—Deep neural networks (DNNs) for nonlinear generative mixture modeling typically rely on unsupervised learning that employs hard clustering schemes, or variational learning with loose / approximate bounds, or under-regularized modeling. We propose a novel statistical framework for a *DNN mixture model* using a single *generative adversarial network*. Our learning formulation proposes a novel data-likelihood term relying on a well-regularized / constrained Gaussian mixture model in the latent space along with a prior term on the DNN weights. Our *min-max learning* increases the data likelihood using a tight variational lower bound using *expectation maximization* (EM). We leverage our min-max EM learning scheme for *semi-supervised learning*. 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.**

***Index Terms*—Deep neural network, nonlinear generative mixture model, adversarial learning, expectation maximization, semi-supervision, generative image modeling, image clustering.**

I. Introduction

Generative mixture modeling has applications across many fields including image analysis, e.g., clustering, interpolation, and generation. Some of the earliest such methods model real-world data distributions using variants of a multivariate-Gaussian mixture model (GMM), where efficient model fitting relies on expectation maximization (EM) [1]. To better model the nonlinear manifolds that real-world data typically lie around, some extended approaches rely on kernel methods [2] and spectral clustering methods [3] which implicitly nonlinearly map the data to a higher-dimensional space, e.g., a reproducing kernel Hilbert space (RKHS), and fit a parametric mixture model (e.g., a GMM) in the mapped space. Analogous to the kernel-GMM is the kernelized dictionary modeling approach that relies to sparsity-based regularization [4]. However, all such methods rely on modeling data using hand-crafted features / kernels, and the methods' performance can be sensitive to these manual designs. Moreover, it can become difficult / infeasible to visualize key RKHS statistics, e.g., the mean or modes of variation, as pre-images in the absence of explicit mappings between the input space and the RKHS.

More recent methods for mixture modeling or clustering rely on deep neural networks (DNNs) that enable data-driven feature learning optimized to the task at hand. Unlike many earlier methods, DNNs enable the explicit modeling of nonlinear mappings to model the manifolds that real-world data reside around. Some such DNN methods [5], [6] use auto-encoder (AE) based formulations for mixture modeling, or clustering, by employing k-means based clustering in the associated latent space. Such approaches enforce *hard* / crisp cluster *memberships* leading to a strong prior that assumes the inter-cluster probability density functions (PDFs) to be well separated. Some DNN methods [7], [8] employ variational AE (VAE) formulations that model a distribution of possible latent-space encodings for each datum, and marginalize out their effects during learning. To do so, they typically rely on variational bounds on the log-likelihood function, where the *bounds lack tightness*, both analytically and empirically. Most of the recent DNN methods [9], [10], [11] leverage generative adversarial networks (GANs) [12], [13] that employ an adversarial discriminator DNN that aids the generator in updating the generated-sample PDF towards the observed-data PDF. However, some such methods use approximate lower bounds that can make learning unreliable, and some employ multiple-GAN frameworks that seriously risk over-fitting. In contrast, we propose a novel *DNN-based mixture model* that (i) uses a single-GAN framework with a well-regularized learning formulation, (ii) models the latent-space PDF as a constrained GMM, and (iii) formulates min-max learning to learn the adversary and jointly increase the data likelihood using a tight variational lower bound using EM.

Most DNN methods for generative mixture modeling or clustering rely on unsupervised learning. We propose to extend our learning framework to *semi-supervised learning* where a small amount of expert supervision, through cluster labels for a subset of the data, has the potential to improve the learning. Thus, we extend our unsupervised adversarial-learning EM-based framework to semi-supervised learning.

This paper makes several contributions. First, it proposes a novel statistical framework for a *DNN-based mixture model* (DNN-MM) using a single GAN, i.e., one generator, one encoder, and one discriminator. Our learning formulation proposes a novel data-likelihood term relying on a well-regularized / constrained Gaussian mixture model in the latent space along with a prior term on the DNN weights. Second, we propose a novel learning formulation by combining *min-max learning* with *EM-based learning*, termed MinMax+EM, leveraging a variational lower bound that analytically guar-

The authors are grateful for support from the Infrastructure Facility for Advanced Research and Education in Diagnostics grant funded by Department of Biotechnology (DBT), Government of India (BT/INF/22/SP23026/2017).

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412739

antees tightness to the log-likelihood of the data. Third, we propose to extend our MinMax+EM learning to *semi-supervised learning*. Fourth, 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.

II. Related Work

We describe related methods for mixture PDF estimation or clustering that rely on the number of mixture components or clusters being known. One class of methods rely on clustering the data using *hand-crafted features*. Some of the early and popular methods of clustering includes k-means and k-means++ [14], both of which rely on hard clustering leading to discrete optimization problems that are NP hard. Later methods that became popular include Gaussian mixture models (GMMs) that enabled modeling the cluster PDFs as multivariate Gaussian, and lead to efficient fits using EM optimization [1] that also gave fractional memberships (posterior probability) for each datum belonging to each cluster. Modeling fractional memberships to clusters typically improves clustering performance when the cluster PDFs have some amount of overlap, by leading to better model fits that account for the possibility / uncertainty of datum to belonging to multiple clusters. To enable modeling a mixture-component PDF as complex non-Gaussian, later methods extended GMMs to kernel GMMs [15] that modeled each mixture-component PDF as a Gaussian in the implicitly-mapped RKHS [2]. Other methods propose dictionary modeling with sparsity based prior on the coefficients [16], principal geodesic analysis in RKHS [17], dictionary modeling in RKHS [18], generalized Gaussian modeling in RKHS [19], and spectral clustering [3]. The performance of these aforementioned methods often depends heavily of the quality of the hand-crafted features.

DNNs enable the joint learning of the features along with the clustering for a given class of observed data, thereby relieving the designer from hand-crafting features. In this way, for clustering a large amount of data, DNN based methods typically outperform methods using hand-crafted features. A survey of DNN-based clustering methods appears in [20]. This paper focuses on methods that rely on generative mixture modeling of the data. Such methods can effectively be partitioned into three categories: (i) AE based methods, (ii) VAE based methods, and (iii) GAN based methods. In general, these methods use a learning framework that combines a network loss (for regularization or consistency) and a clustering loss (to promote grouping of the data). The network loss can include the reconstruction loss in AEs, the variational loss in VAEs, or the adversarial loss in GANs.

AE-based Methods. DCN [5] trains an AE and jointly optimizes a k-means-based hard-clustering loss in latent-space along with the reconstruction loss of the AE. DynAE [6] is a recent improved version of DCN that uses a heuristic to dynamically update the subset of the training set for each kind of loss (clustering and reconstruction). Both DCN and DynAE perform hard clustering (instead of soft clustering) and explicitly estimate the cluster means in latent space. Because the nonlinearity in the DNN mappings that can easily adapt to linear / nonlinear transformations of the means and covariances, explicit mean estimation can make the model over-flexible / under-regularized and make it prone to, say, mode collapse. DSC-Nets [21] uses a self-expressive layer that enforces a sparse representation of each latent-space encoding using other encodings. Consequently, it leads to quadratic complexity in time and space both, limiting its use for large datasets [20]. This also limits the applicability of batch-based backpropagation schemes because the gradient of the self-expressive loss depends on the entire dataset. DEPICT [22] augments an AE with a classifier on latent space and, to avoid trivial solutions, enforces a strong prior on the classifier output to produce a close-to-uniform distribution on empirical label distribution. In contrast, our method relies on soft-clustering models and estimates the mixture-component scaling factors from the data using EM optimization, while making it easy to incorporate priors on the scaling factors.

VAE-based Methods. VaDE [7] extends the VAE formulation by replacing the Gaussian prior in the latent space to a GMM prior. Akin to [5], [6], they explicitly estimate the component means and covariances, making the model under-regularized. Furthermore, they inherit from the VAE formulation the limitation of the evidence lower bound (ELBO) scheme being an approximation to optimizing the true objective function because the bound is *not* tight. [8] extends the GMM prior in the VAE formulation to include another prior involving a graph-embedding model that enforces similar data to have similar embeddings. They aim to improve VaDE's optimization scheme to get a tight variational bound by minimizing a Kullback-Leibler (KL) divergence, but their scheme lacks the guarantee to make KL divergence zero, both analytically and empirically. Moreover, designing an effective graph-embedding affinity matrix relies on learning a Siamese DNN that needs rich prior information unavailable with unsupervised clustering methods. In contrast, our method relies on an EM-based variational lower bound that analytically guarantees tightness to the log-likelihood of the data.

GAN-based Methods. InfoGAN [9] has a generator that takes as input a noise vector and a latent code (which indicates the cluster assignment when InfoGAN is adapted for the clustering problem). InfoGAN maximizes mutual information between the latent space and the generated images. Like VAE-based methods, InfoGAN also suffers from the true posterior associated with the variational lower bound being unknown in practice, with only the hope of approximating it by the DNN during learning.

Mixture of GANs [10] proposes a separate GAN to model each cluster within an EM framework, unlike our method that utilizes the same generator and discriminator across all clusters. Thus, [10] acknowledges that their method is prone to early convergence in EM, risking trivial solutions, and, thereby, purposely introduces an error in the variational bound in the E step, calling it ϵ-EM. Thus, [10] loses convergence guarantees underlying EM. [10] uses a classifier that gives

cluster probabilities directly, unlike our model that explicitly introduces a GMM in latent space. To deal with class imbalance in the training set, [10] uses a heuristic for data augmentation, because they have a different GAN for each cluster. In contrast, our model is well regularized because of sharing one GAN to model all clusters, together with an auto-encoding based regularizing prior.

ClusterGAN [11] extends InfoGAN to replace mutual information by and auto-encoding loss and a cross-entropy term, and, thereby, avoids ELBO related approximations. Unlike VAE-based methods that estimate all GMM parameters, which risks under-regularization, ClusterGAN fixes the parameterization of the encoded PDFs in latent space; this is similar in spirit to our method. ClusterGAN models the latent space as discrete-continuous, and its architecture counter-intuitively uses the same encoder to map the input image (i) to a continuous-valued (noise) vector and (ii) to a one-hot encoding. In contrast, our method models the latent space more conveniently as a Euclidean space comprising an explicit multivariate GMM that, under the nonlinear generator mapping, models the nonlinear mixture model of the observed images. ClusterGAN's formulation avoids modeling different proportions of probability masses / data corresponding to different clusters, while our formulation explicitly models the proportions through the scaling parameters. Moreover, we formulate the learning problem as an extension of a maximum-likelihood / maximum-posterior estimation problem for which we propose an efficient EM optimization coupled with the min-max optimization problem underlying the adversarial learning. Finally, our unsupervised learning formulation coupled with EM based optimization lends itself naturally to a semi-supervised learning formulation, while none of the aforementioned methods (including ClusterGAN) extend their work to the semi-supervised learning scenario.

III. Methods

We describe a novel DNN-based learning framework for semi-supervised nonlinear generative mixture modeling using MinMax+EM, with applications to clustering.

A. Nonlinear Generative DNN Mixture Model (DNN-MM)

Let a set of N random fields $\{X_n\}_{n=1}^N$ model the observed set of images, all of which have an identical PDF $P(X)$. We assume that $P(X)$ comprises K mixture components, where each component represents a nonlinear distribution. This paper tunes free parameter K using cross validation.

1) **Generator Modeling:** We propose to learn a DNN-based generative model for the PDF $P(X)$ of a given class / kind of images. We propose to learn a DNN-based generator $\mathcal{G}(\cdot;\theta_G)$, parameterized by DNN weights θ_G, that can generate images belonging to $P(X)$ through the action of the *nonlinear transformation function* $\mathcal{G}(\cdot;\theta_G)$ on a random vector Y having a known PDF $P(Y)$ in some *latent space*. We model the latent space as the L-dimensional Euclidean space $\mathbb{R}^L$ (this paper sets $L := 30$; typically $L \gg K$). Specifically, we model $\mathcal{G}(\cdot;\theta_G)$ as a nonlinear transformation function that transforms a *multivariate Gaussian mixture* PDF $P(Y)$ in the latent space to the PDF $P(X)$ of the class of images.

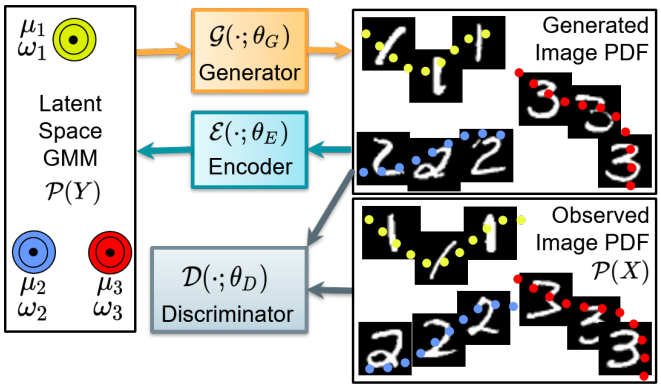


Fig. 1. **Our Architecture for DNN Mixture Modeling** using semi-supervised MinMax+EM learning.

Knowing that the $P(X)$ is a mixture of K components, and that the generator mapping is typically highly nonlinear stemming from the DNN architectural design, we model the latent-space PDF $P(Y)$, without loss of generality, as a mixture of K (fixed) Gaussians in latent space, with each mean lying at a corner of a K-simplex in the positive orthant of the latent-space (at a distance of 4 units from the origin) and each covariance being the identity matrix $\mathbf{I}$. Let the Gaussian means be $\{\mu_k \in \mathbb{R}^L\}_{k=1}^K$. Let the set of scaling factors associated with each component in the mixture be $\omega := \{\omega_k\}_{k=1}^K$, under with constraints that $\sum_{k=1}^K \omega_k = 1$ and $\omega_k > 0, \forall k$. The scaling factors are unknown and we propose to estimate them from the training data. Our learning strategy is to have each Gaussian component in $P(Y)$, under transformation $\mathcal{G}(\cdot;\theta_G)$, to lead to the corresponding mixture component in $P(X)$.

2) **Encoder Modeling:** During learning, if we knew which observed image X came from which component / cluster k, then it would be easy to learn, say, a separate DNN to model each mixture component in $P(X)$. However, obtaining such cluster labels may (in the worst case) be intractable in practice or (in the best case) entail a laborious process of sifting through the data and labeling each image. Thus, in many real-world applications involving large datasets, such labeled information is unavailable for the observed data X.

We propose to reciprocate the generator mapping $\mathcal{G}(\cdot;\theta_G)$ by modeling an *encoder* mapping $\mathcal{E}(\cdot;\theta_E)$, parameterized by DNN weights θ_E, that maps images X to the latent space. The learning strategy for the encoder is that, if X was drawn from the k-th mixture component in $P(X)$, then $\mathcal{E}(X;\theta_E)$ maps to an encoding closer to the k-th mean μ_k in the latent space. In this way, the encoder mapping enables us to infer (i) the probability density of image X being drawn from cluster k and (ii) the probability / membership of an image X belonging to the k-th cluster, by evaluating the Gaussian mixture components in the latent space at the encoding $\mathcal{E}(X;\theta_E)$. Let Z be a *hidden categorical* random variable indicating the mixture component to which image X belongs. Let Z take

integer values within $[1, K]$. Let $P(Z = k) = \omega_k$ be the prior probability occurrence of the k-th component of the mixture model. Then, the likelihood for an image X is

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

The probability density for image X drawn from cluster k is

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

Moreover, the encoder enables us to estimate the probability that image X was drawn from cluster k, i.e., the membership of image X to cluster k, by using Bayes rule, as

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

Thus, the log-likelihood function for the entire training set is

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

3) **Consistency Prior on Generator + Encoder:** To learn an encoder mapping that reciprocates the generator mapping, we want to ensure that the mapping from a $y \sim P(Y)$ in latent space to $\mathcal{G}(y; \theta_G)$ in the image space, followed by the the encoder mapping back to the latent space, i.e., $\mathcal{E}(\mathcal{G}(y; \theta_G); \theta_E)$, remains close to the initial y. This promotes the encoder learning to avoid a "collapse" of its mapping where all generated images map to a subset of the K mixture components in $P(Y)$ in the latent space. So, we propose a log-prior $\log P(\theta_G, \theta_E)$, upto the normalizing constant, as

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

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

4) **Discriminator Modeling:** We want the PDF of the generated samples $\mathcal{G}(Y; \theta_G)$ to match the PDF $P(X)$ underlying the observed data. So, we introduce a DNN-based discriminator as an adversarial learning component that plays two roles iteratively: (i) it learns a decision boundary between the distribution of generated images (output by the generator) and the distribution $P(X)$ of observed images, and (ii) it leverages the decision boundary to help the generator learn to produce images to which the discriminator assigns a larger probability of being drawn from the PDF $P(X)$ underlying the observed images. Let the DNN-based *discriminator / classifier* 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. Then, the discriminator-based terms in the objective function are

$$E_{P(X)}[-\log \mathcal{D}(X; \theta_D)] + E_{P(Y)}[\log \mathcal{D}(\mathcal{G}(Y; \theta_G); \theta_D)] \quad (7)$$

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

where the learning seeks to update the generator weights θ_G to increase the objective function, and update the discriminator weights θ_D to decrease the objective function.

B. DNN Mixture Model (DNN-MM) Learning

We combine the objective-function terms from (i) the encoder-based likelihood, (ii) the prior on the generator-encoder combination, and (iii) the discriminator-based adversarial learning component to propose a novel statistical learning formulation for unsupervised DNN-based mixture modeling. Given the training set $\{x_n\}_{n=1}^N$, the unsupervised learning problem is the min-max optimization as

$$\begin{aligned} \min_{\theta_D} \max_{\omega, \theta_G, \theta_E} & \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \omega_k \mathcal{N}(\mathcal{E}(x_n; \theta_E); \mu_k, \mathbf{I}) \right) \\ & - \lambda_1 \sum_{k=1}^{K} \omega_k \sum_{s=1}^{S} \|y_k^s - \mathcal{E}(\mathcal{G}(y_k^s; \theta_G); \theta_E)\|_2^2 \\ & - \lambda_2 \sum_{n=1}^{N} \log \mathcal{D}(x_n; \theta_D) \\ & + \lambda_2 \sum_{k=1}^{K} \omega_k \sum_{s=1}^{S} \log \mathcal{D}(\mathcal{G}(y_k^s; \theta_G); \theta_D), \end{aligned} \quad (9)$$

where $\{y_k^s\}_{s=1}^S$ is an independent sample drawn from the PDF $\mathcal{N}(\mu_k, \mathbf{I})$ in the latent space, and where $\lambda_1, \lambda_2 \in \mathbb{R}^+$ are (free) weighting parameters tuned by cross validation.

C. Unsupervised MinMax+EM Learning for DNN-MM

In the learning formulation in Section III-B, the log-likelihood term $\log \left(\sum_{k=1}^{K} \omega_k \mathcal{N}(\mathcal{E}(x_n; \theta_E); \mu_k, \mathbf{I}) \right)$ comprises a logarithm of a summation that cannot be simplified further analytically. Nevertheless, variational learning using the iterative expectation-maximization (EM) algorithm provides a way around it by (i) the introduction of the hidden random variable Z indicating the cluster for each input X, (ii) designing a minorization of the data log-likelihood that touches the log-likelihood function at the current parameter estimate, using the $Q(\theta_E, \omega)$ function, in the E step, and (iii) updating the parameters to improve the value of the minorized function in the M step. Thus, we propose to 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

$$\begin{aligned} & Q(\theta_E, \omega; \theta_E^t, \omega^t) \\ & := E_{P(X)} E_{P(Z|X, \theta_E^t, \omega^t)}[\log P(X, Z|\theta_E, \omega)] \quad (10) \\ & = E_{P(X)} \left[\sum_{k=1}^{K} P(Z = k|X, \theta_E^t, \omega^t) \log P(X|Z = k, \theta_E, \omega) \right] \\ & + E_{P(X)} \left[\sum_{k=1}^{K} P(Z = k|X, \theta_E^t, \omega^t) \log \omega_k \right], \quad (11) \end{aligned}$$

which is an optimal lower bound to the log-likelihood function $\log P(X|\theta_E, \omega)$, upto an additive constant that is independent of the parameters θ_E and ω. Here, $P(Z = k|X, \theta_E^t, \omega^t)$ is

the membership of image X to the k-th cluster based on the current parameter estimates. Let the membership of the observed training-set image x_n to the k-th cluster, based on parameter estimates θ_E^t, ω^t, be γ_{nk}^t. Then, at iteration t within the EM algorithm, the objective function becomes

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

D. Semi-Supervised MinMax+EM Learning for DNN-MM

The performance of unsupervised mixture-model learning on tasks like clustering or classification can improve greatly by improving the efficacy of the learning using a small amount of labeled data, i.e., with a small set of images $\{\widetilde{X}_m\}_{m=1}^M$, for which the cluster labels $\{\widetilde{Z}_m\in[1,K]\}_{m=1}^M$ are provided. We propose an extension of the EM-based variation learning scheme that leverage this small amount of labeled training data. For those input images for which the label is provided by experts, we do *not* need to introduce a hidden variable and we can consider the membership function to be crisp, i.e., the membership belongs completely to one and only one cluster. This introduces two additional terms in the objective function involving the variables $\widetilde{X}_m$, analogous to those involving the images X_n. Thus, the semi-supervised learning formulation is

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

where $\mathcal{I}(\widetilde{Z}_m,k)$ is the indicator function that takes a value of 1 when $\widetilde{Z}_m=k$, and takes a value of 0 otherwise. We define the *level of supervision* as $\alpha:=M/(M+N)$ that takes real values within $[0,1]$. Here, $M=0\implies\alpha=0$ leading to the unsupervised learning case, and $N=0\implies\alpha=1$ leading to the supervised learning case. In general, $M>0$ and $N>0$ leads to the semi-supervised learning case where $\alpha\in(0,1)$.

E. MinMax+EM Optimization

Within the MinMax+EM formulation in Section III-D, within each iteration t, we propose an alternate minimization scheme for the parameters $\theta_G,\theta_E,\theta_D,\omega$. We use Adam [23] for updating the DNN weights $\theta_G,\theta_E,\theta_D$ and we use projected gradient descent to update the scaling factors ω, where the projection is on the convex set comprising positive ω_k values that sum to one. After updating $\theta_G,\theta_E,\theta_D,\omega$ at iteration, we move to iteration $t+1$ and repeat the iterations. We tune the free parameters, i.e., K, λ_1, and λ_2, using the validation set to maximize the clustering accuracy.

1) **Pretraining Strategy:** For any clustering method, especially relying on unsupervised learning or relying on learning with a small amount of supervision, the initialization strategy for the variables being optimized can be important for improved performance. In the pretraining stage, we propose to find a good initializations for the DNN parameters $(\theta_G,\theta_E,\theta_D)$ using a sequential pretraining scheme as follows.

Semi-supervised K-means. We first run semi-supervised kmeans on the images in the training set data to get an initial clustering. During this process, we fix the cluster label for those images X for which it has been provided as part of the expert supervision. We also update ω_k to be the fraction of the images assigned to cluster to cluster k, leading to a pretrained estimate of the latent-space PDF $P'(Y)$.

Encoder Pretraining. We use the kmeans clustering to pretrain the encoder such that, if an image X mapped to cluster $z_n\in[1,K]$, then the encoder learns to map that X close to the mean μ_{z_n} in the latent space. Thus, we pretrain θ_E to

$$\theta_E':=\arg\max_{\theta_E}\prod_{n=1}^{N}\mathcal{N}(\mathcal{E}(x_n;\theta_E);\mu_{z_n},\mathbf{I}).\tag{14}$$

Generator Pretraining. After fixing the encoder parameters θ_E, we pretrain the generator weights θ_G to

$$\begin{aligned}\theta_G':=\arg\min_{\theta_G}&E_{P'(Y)}[\|Y-\mathcal{E}(\mathcal{G}(Y;\theta_G);\theta_E')\|_2^2]\\ &+E_{P(X)}[\|X-\mathcal{G}(\mathcal{E}(X;\theta_E');\theta_G)\|].\end{aligned}\tag{15}$$

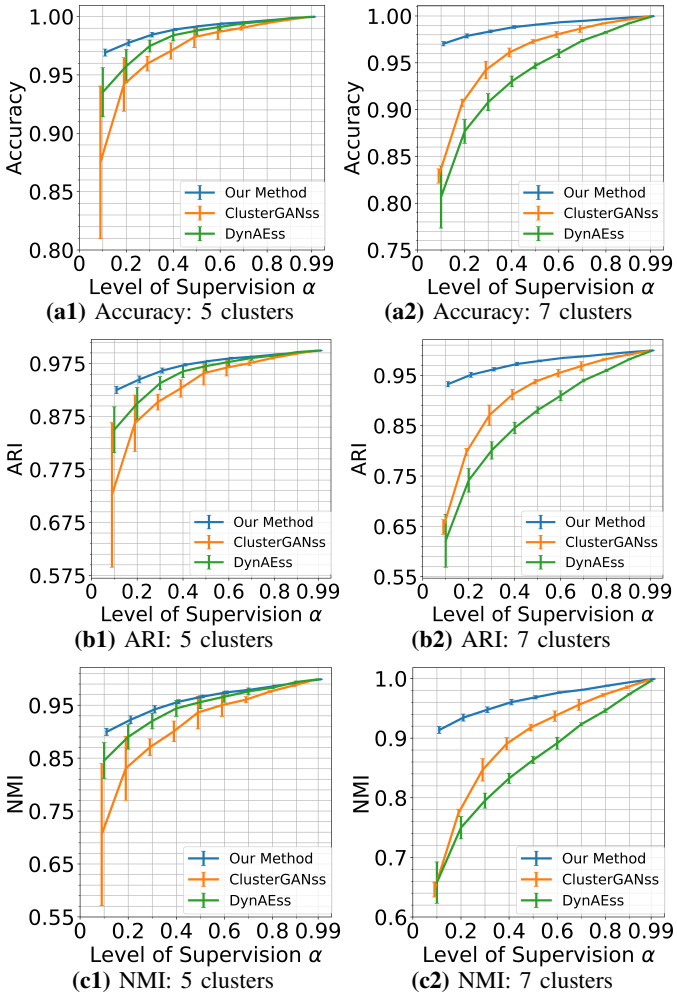
Discriminator Pretraining. After pretraining the generator, we pretrain the discriminator to separate the PDFs of the generated data and the observed data. Thus, we pretrain θ_D to

$$\begin{aligned}\theta_D':=\arg\min_{\theta_D}&E_{P(X)}[-\log\mathcal{D}(X;\theta_D)]\\ &+E_{P'(Y)}[\log\mathcal{D}(\mathcal{G}(Y;\theta_G');\theta_D)].\end{aligned}\tag{16}$$

IV. Results and Discussion

For evaluation, we use 3 publicly available real-world datasets: (i) MNIST [24], (ii) CIFAR10 [25], and (iii) CelebA [26]. We compare our method with 2 other methods: (i) *ClusterGANss*: an extension of ClusterGAN [11] to semi-supervised learning, and (ii) *DynAEss*: an extension of DynAE [6] to semi-supervised learning, where DynAE is known to improve over DCN [5]. Compared to ClusterGAN, our ClusterGANss introduces an additional loss term penalizing the cross entropy between its encoder-estimated encodings

and the true one-hot encodings for the subset of training set (where the true one-hot encodings are known). DynAEss uses a similar strategy to improve over DynAE. As per the results in [27], ClusterGAN provided favorable clustering results on CIFAR10, compared to earlier methods. As per the results in [6], DynAE provided favorable clustering results on MNIST, compared to earlier methods. Thus, ClusterGAN and DynAE have the best known performance for generative clustering on the CIFAR10 and MNIST, respectively. For all 3 datasets, we evaluate all 3 methods at varying levels of supervision $\alpha = [0.1, 0.2, \cdots]$. We use 3 metrics for quantitative evaluation of the clustering: (i) accuracy, (ii) adjusted rand index (ARI), and (iii) normalized mutual information (NMI). For each dataset, we choose a random subset of images to evaluate mixture modeling / clustering; to evaluate the variability in the performance resulting from the choice of the chosen subset, we repeat the evaluation 15 times and show error bars for the performance metrics.

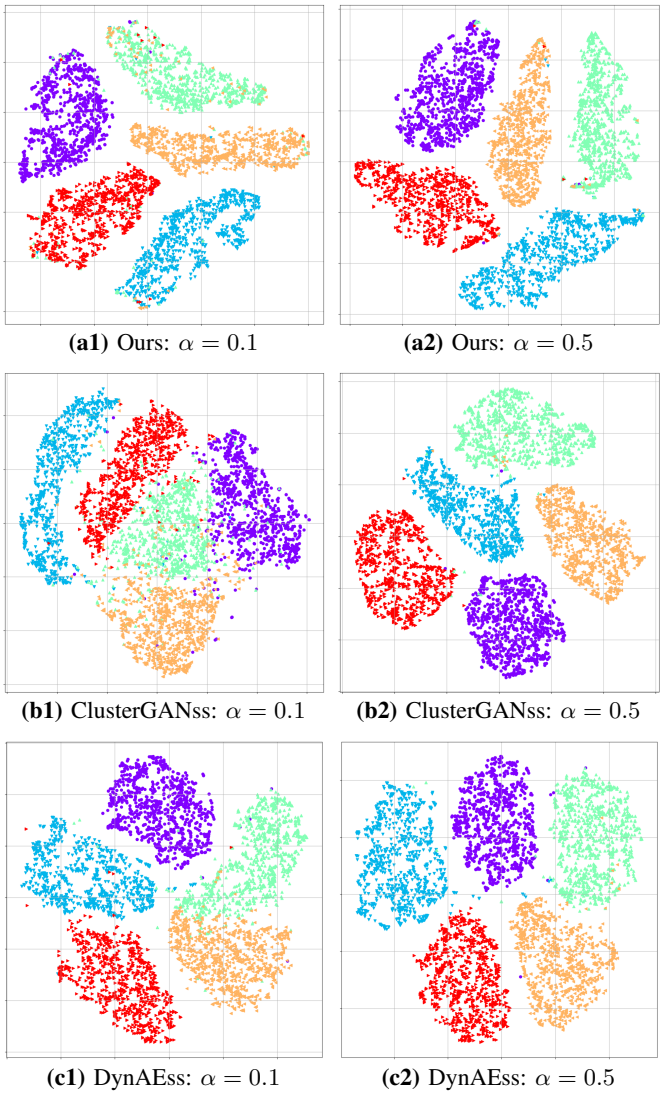


(a1) Accuracy: 5 clusters **(a2)** Accuracy: 7 clusters

(b1) ARI: 5 clusters **(b2)** ARI: 7 clusters

(c1) NMI: 5 clusters **(c2)** NMI: 7 clusters

Fig. 2. **Results: MNIST; Quantitative.** Clustering performance for all methods at varying levels of supervision $\alpha = 0.1, 0.2, \cdots$.



(a1) Ours: $\alpha = 0.1$ **(a2)** Ours: $\alpha = 0.5$

(b1) ClusterGANss: $\alpha = 0.1$ **(b2)** ClusterGANss: $\alpha = 0.5$

(c1) DynAEss: $\alpha = 0.1$ **(c2)** DynAEss: $\alpha = 0.5$

Fig. 3. **Results: MNIST; Visualizations of Latent-Space Encoding PDFs.** t-SNE visualizations of latent-space PDFs for all methods at levels of supervision $\alpha = 0.1, 0.5$.

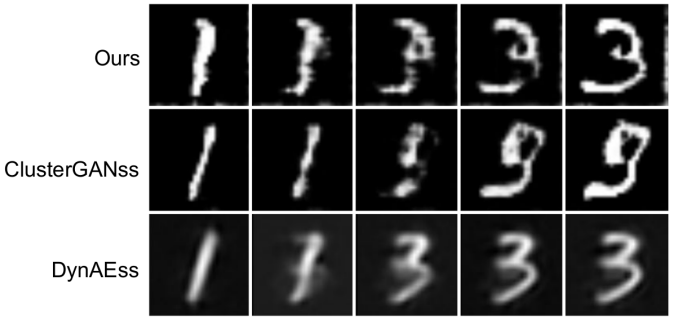
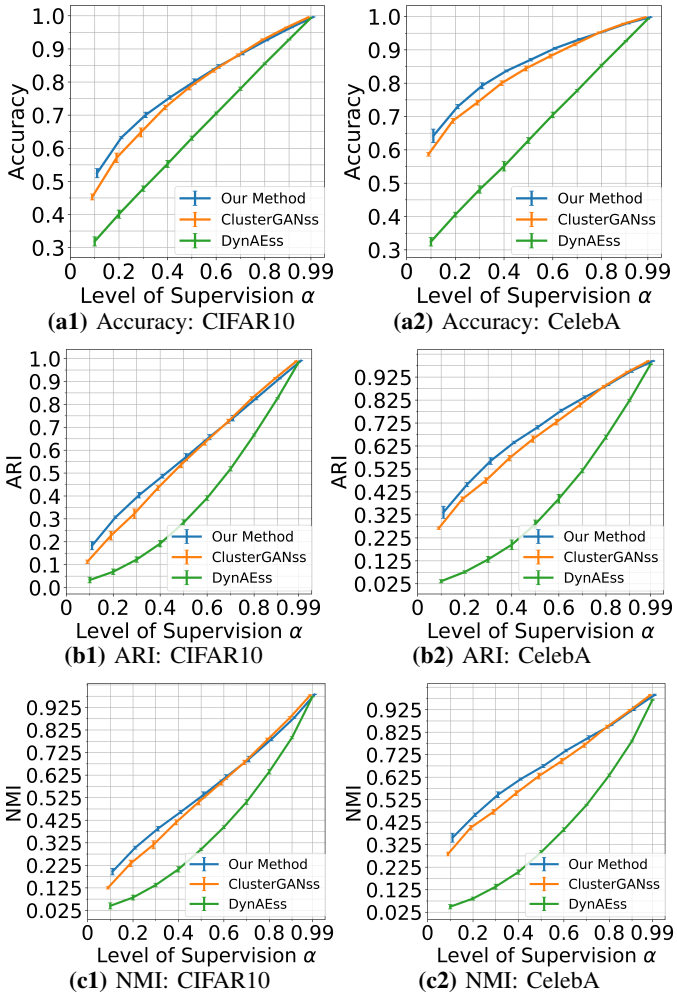


Fig. 4. **Results: MNIST; Interpolation in Latent Space.** At level of supervision $\alpha = 0.1$, generated images using (i) linear interpolation between the mixture-component means (corresponding to digits 1 and 3) in latent space and (ii) subsequent mapping to the image space using the generator.

A. Results: MNIST Dataset

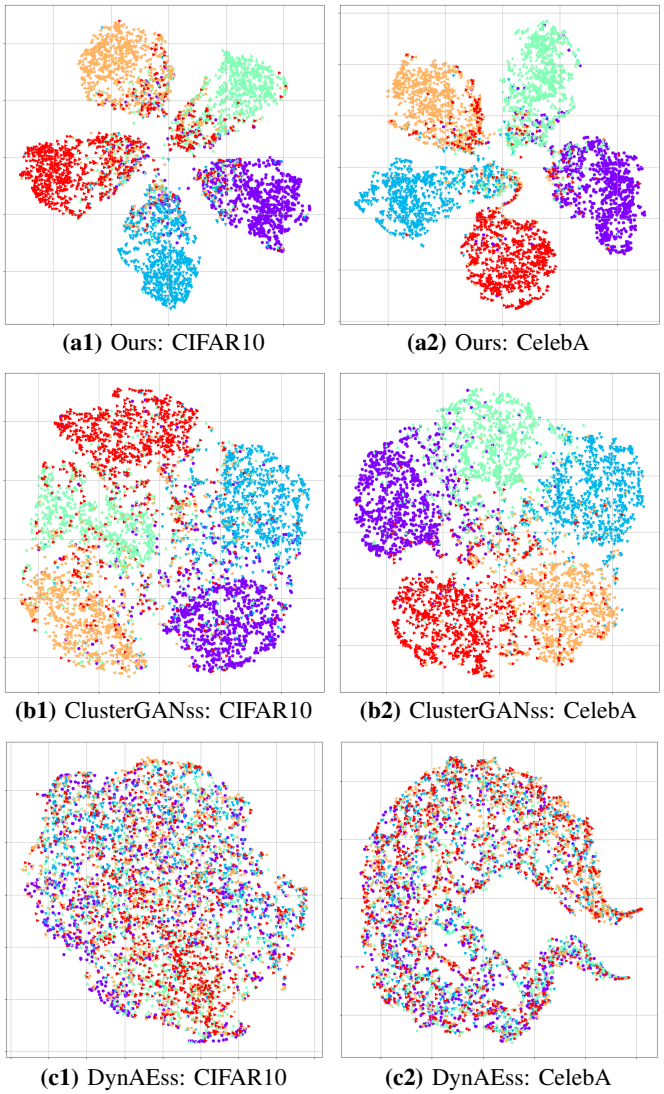
For the MNIST dataset, we evaluate methods on two tasks: (i) mixture modeling 5 classes, i.e., 1000 images of each digit

from 0 to 4 (Figure 2(a1)–(c1)), and (ii) mixture modeling 7 classes, i.e., 1000 images of each digit from 0 to 6 (Figure 2(a1)–(c1)). For both these tasks, at virtually all levels of supervision α, and all 3 clustering-performance metrics, our method outperforms ClusterGANss and DynAEss (Figure 2). While the performance of ClusterGANss and DynAEss deteriorates as the dataset incorporates more number of clusters, our method's performance remains virtually unaffected. For the 5-cluster dataset, DynAEss performs better than ClusterGANss, but ClusterGANss starts to improve over DynAEss as the number of clusters increases (to 7) and the mixture modeling becomes more challenging. The t-SNE visualizations of the latent-space PDFs (Figure 3) clearly indicate that the proposed method produces image encodings for each mixture component with far smaller overlap across components, at both levels of supervision $\alpha = 0.1$ and $\alpha = 0.5$. Consequently, despite the relatively small size of the dataset (1000 images per mixture component), for our method, the images generated by mapping the mixture-component means in latent space resemble actual hand-written digit images quite well (Figure 4). In contrast, the appearance of the digit three generated from ClusterGANss



(a1) Accuracy: CIFAR10 **(a2)** Accuracy: CelebA
(b1) ARI: CIFAR10 **(b2)** ARI: CelebA
(c1) NMI: CIFAR10 **(c2)** NMI: CelebA

Fig. 5. **Results: CIFAR10 and CelebA; Quantitative.** Clustering performance for all methods at varying levels of supervision $\alpha = 0.1, 0.2, \cdots$.



(a1) Ours: CIFAR10 **(a2)** Ours: CelebA
(b1) ClusterGANss: CIFAR10 **(b2)** ClusterGANss: CelebA
(c1) DynAEss: CIFAR10 **(c2)** DynAEss: CelebA

Fig. 6. **Results: CIFAR10 and CelebA; Visualizations of Latent-Space Encoding PDFs.** t-SNE visualizations of latent-space PDFs for all methods at level of supervision $\alpha = 0.5$.

using that cluster representative in latent space is distorted (Figure 4). Latent-space means in DynAEss map to images that are blurrier and dimmer, probably because of (i) the lack of an adversarial component and (ii) the hard-clustering that leads to unweighted averaging of the encodings in the overlapped mixture distributions. Moreover, interpolating between the means in latent space shows a more gradual semantic variation for our method compared to others (Figure 4). For DynAEss, the interpolation leads to an intermediate stage (second image from left) that seems like a blend of the digits 1 and 3.

B. Results: CIFAR10 and CelebA Datasets

For the CIFAR10 dataset, we evaluate all methods on two tasks: (i) mixture modeling 3 classes, i.e., 1000 images from each of the first 3 classes, and (ii) mixture modeling 5 classes, i.e., 1000 images from each of the first 5 classes. To make

the task more challenging, we introduce white noise in the red channel of the RGB images, with a standard deviation of 10% of the intensity range. For the CelebA dataset, we choose 1000 images each from 5 distinct classes corresponding to (i) BlackHair, (ii) BlondeHair, (iii) BrownHair, (iv) GrayHair and (v) Bald. To make the task more challenging, we introduce white noise in the each channel of the RGB images, with a standard deviation of 20% of the intensity range. Both these datasets exhibit complex image PDFs for each class.

Our GMM-based data-likelihood maximizing formulation leads to statistically significantly better performance than ClusterGANss (Figure 5), especially at smaller levels of supervision α, indicating improved robustness to noise, for both CIFAR10 and CelebA datasets. Indeed, for both CIFAR10 and CelebA datasets, t-SNE visualizations of the latent-space PDFs show (Figure 6) that (i) DynAEss is unable to seperate the representations of the mixture components in latent space, (ii) ClusterGANss performs much better then DynAEss, but still leads to a significant overlap between the encodings of multiple mixture components / clusters, and (iii) our method maintains significantly better separability between clusters as well as restricts the overlap between clusters to a significantly smaller region in latent-space.

V. Conclusion

We have proposed a novel statistical framework for a DNN-based mixture modeling using a single GAN. We leverage the generative component in our GAN to *nonlinearly transform* a GMM model in latent space to a nonlinear mixture model in the space of images. We leverage the adversarial component in our GAN to aid the learning of the generator in order to drive the generated-sample PDF towards the observed-data PDF. Our learning formulation proposes a novel data-likelihood term relying on a well-regularized and constrained Gaussian mixture model in the latent space (with only the scaling weights being learned) along with a prior term on the DNN weights. Unlike VAE-based methods, our min-max learning increases the data likelihood using a *tight* variational lower bound using EM. Unlike typical DNN-based mixture models, we leverage our *MinMax+EM* learning scheme for *semi-supervised* learning. 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] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc.*, vol. B, no. 39, pp. 1–38, 1977.

[2] T. Hofmann, B. Scholkopf, and A. Smola, "Kernel methods in machine learning," *Annals of Stat.*, vol. 36, no. 3, pp. 1171–220, 2008.

[3] U. Luxborg, "A tutorial on spectral clustering," *Stat. and Comput.*, vol. 17, p. 395416, 2007.

[4] H. Nguyen, V. Patel, N. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Trans. Imag. Proc.*, vol. 22, no. 12, pp. 5123–35, 2013.

[5] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Int. Conf. Mach. Learn.*, 2017, pp. 3861–70.

[6] N. Mrabah, N. Khan, and R. Ksantini, "Deep clustering with a dynamic autoencoder," arxiv.org/abs/1901.07752, 2019.

[7] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," in *Int. Joint Conf. Artific. Intell.*, 2017, p. 196572.

[8] L. Yang, N. Cheung, J. Li, and J. Fang, "Deep clustering by Gaussian mixture variational autoencoders with graph embedding," in *Int. Conf. Comp. Vis.*, 2019, pp. 6439–48.

[9] 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.

[10] Y. Yu and W.-J. Zhou, "Mixture of GANs for clustering," in *Int. J. Conf. Artific. Intell.*, 2018, p. 304753.

[11] 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.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Adv. Neur. Info. Proc. Sys.*, 2014, pp. 2672–80.

[13] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," abs/1701.00160, 2017.

[14] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *ACM-SIAM Symp. Discr. Algo.*, 2007, p. 102735.

[15] J. Wang, J. Lee, and C. Zhang, "Kernel trick embedded Gaussian mixture model," in *Int. Conf. on Algo. Learn. Th.*, 2003, pp. 159–74.

[16] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE Comp. Vis. Patter. Recog.*, 2010, pp. 3501–8.

[17] S. P. Awate, Y.-Y. Yu, and R. T. Whitaker, "Kernel principal geodesic analysis," in *Proc. Euro. Conf. Machine Learning and Knowledge Discovery in Databases*, vol. 8724, 2014, pp. 82–98.

[18] N. N. Koushik and S. P. Awate, "Robust dictionary learning on the Hilbert sphere in kernel feature space," in *Proc. Euro. Conf. Machine Learning and Knowledge Discovery in Databases*, vol. 9851, 2016, pp. 731–48.

[19] N. Kumar and S. P. Awate, "Semi-supervised robust mixture models in RKHS for abnormality detection in medical images," *IEEE Trans. Imag. Proc.*, vol. 29, pp. 4772–87, 2020.

[20] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39 501–14, 2018.

[21] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Adv. Neur. Info. Proc. Sys.*, 2016, pp. 1–10.

[22] K. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *Int. Conf. Comp. Vis.*, 2017, pp. 5747–56.

[23] D. Kingma and B. Jimmy, "Adam: A method for stochastic optimization," *Int. Conf. Learn. Rep.*, 2015.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–324, 1998.

[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[26] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Int. Conf. Comp. Vis.*, 2015, pp. 3730–8.

[27] 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.