Probabilistic Models with Invariance

Alexander Gepperth[0000−0003−2216−7808]

Fulda University of Applied Sciences
Applied Computer Science Department, Leipzigerstr.123, 36037 Fulda, Germany
alexander.gepperth@cs.hs-fulda.de
www.gepperth.net/alexander

Abstract. This proof-of-concept work generalizes the concept of invariance, as used in contrastive learning, to fully probabilistic models (such as, e.g., mixture models) that explicitly describe data distributions in an interpretable fashion, and whose main applications are density estimation (e.g., outlier detection), sampling and tractable inference. Invariance allows allows probabilistic models to operate at a lower effective model complexity, and therefore to deal with more complex (image) data. In this article, we propose iGMM, a Gaussian Mixture Model (GMM) that explicitly incorporates invariance into its loss, which is a generalization of the conventional GMM log-likelihood. When constructing hierarchies of conventional GMM and iGMM instances, we obtain invariance properties that are reminiscent of simple and complex cells in the mammalian visual cortex. We show, by experiments on the MNIST and FashionMNIST dataset, that GMM-iGMM hierarchies can faithfully sample from learned data distributions even if the iGMM is invariant to some aspects of the data, and demonstrate that outlier detection performance is strongly enhanced in GMM-iGMM hierarchies.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

Recent years have seen tremendous advances in what is now called *generative models*, whose primary function after training is to generate (i.e., sample) images that are similar to training images. Most prominent representatives of generative models are doubtlessly GANs and VAEs.

In contrast to this, we term models that aim at directly capturing the data density $p(\boldsymbol{x})$ *fully probabilistic models*. Important examples are mixture models such as Gaussian Mixture Models (GMMs). Probabilistic models offer efficient sampling, outlier detection as well as tractable inference, although model complexity needs to be significantly higher for realistic data (e.g., images) to produce meaningful results[10].

1.1 Motivation: complex image data for CL

There are several ways to motivate interest in more powerful probabilistic models. Besides conceptual reasons, probabilistic models are of particular interest

in the field of continual learning (CL, [13]), i.e., machine learning from non-stationary data distributions. An obvious concern is, e.g., the detection of non-stationarity onsets via outlier detection. Furthermore, efficient sampling is relevant for replay methods in CL where there is a dedicated "generator" for remembering "past" samples instead of storing them. Lastly, inference is required when querying for samples that are similar to the ones that are currently processed. Probabilistic model such as sum-product networks (SPNs, see [1,15,8]) are unfortunately not yet able to process complex datasets that contain significant structural variations such as color shifts or translations. DCGMMs have had limited success in handling SVHN [2], although the color variations in particular pose significant challenges.

Probabilistic models have a drawback: they aim to model the full data distribution, including aspects that are completely irrelevant to downstream tasks. For example, in a digit recognition problem, the polarity or foreground/background color of digits is irrelevant, yet, e.g., a GMM will spend considerable resource capturing these aspects. If probabilistic models such as GMMs could be trained to ignore certain aspects of the input distribution (e.g., shifts, background color, rotation, ...), this kind of *invariance* would free up significant resources for modeling the task-relevant aspects of a data distribution. In a digit recognition task, such GMMs with invariance (iGMMs) would mainly model aspects related to shape. Please see fig. 1(left) for a visualization of the concept.

1.2 Related work

To the best of our knowledge, no works that integrate invariance into probabilistic models have been proposed. In the following text, we shall discuss models that cover some aspects of probabilistic models, which we understand to include sampling, outlier detection and probabilistic inference:

VAEs and GANs GANS and VAEs do not achieve their considerable successes by drawing samples from an explicit learned representation $p(\boldsymbol{x})$ of the data density. VAEs in particular rather model the prior probability for latent variables $p(\boldsymbol{z})$, the conditional decoder density $p(\boldsymbol{x}|\boldsymbol{z})$ and the conditional encoder density $p(\boldsymbol{z}|\boldsymbol{x})$. While VAEs cannot perform tractable inference, they can use importance sampling to approximate $p(\boldsymbol{x})$, although at considerable computational cost. GANs cannot perform density estimation, outlier detection or tractable inference since they are not formulated in terms of probabilities. A key issue with GANs and VAEs is that they can use the standard tools of DNNs, such as max-pooling layers, to ensure invariance at least to positional shifts or rotations.

Flow-based models Flow-based models [9,6,4] try to transform a known distribution into the observed data distribution by means of adaptive invertible transformations, usually mediated by special deep neural networks. Indeed, competitive sampling has been demonstrated for complex image data, whereas outlier detection suffers from from problems that are not fully understood [17]. Since the overall distribution is still intractable, inference seems to difficult as well. As the overall transformation mediated by any flow-based model is supposed to be

invertible, standard DNN mechanisms like max-pooling layers are not applicable for ensuring invariance.

Sum-product networks Sum-product networks (sometimes termed probabilistic circuits or Einsum networks) [1,8,15] are directed trees whose leaves represent tractable probability densities, and whose structure aims at efficiently expressing hierarchical mixture models. They are thus a true probabilistic model and can perform all associated functions. However, they are lacking in expressiveness when modeling real-world images: even for MNIST and similar datasets, generated samples do not seem overly realistic, while more complex problems like SVHN are currently out of scope. We know of no works that attempt to incorporate invariance into SPNs or related models.

Mixture models and DCGMMs The most well-known mixture models are Gaussian Mixture Models (GMMs). They implement a "flat" hierarchy with just a single layer of computation that is global in the sense that the whole image is processed at once. There have been several attempts to create hierarchical GMMs, see., e.g., [14,12,11], which have however only been applied to synthetic data. Mixture of Factor Analyzers (MFAs) are an extension of GMMs that has been applied to model the CelebA face dataset [10], although using extensive pretraining. To the best of our knowledge, the only attempt to create a hierarchical *and* convolutional GMM is that of [2], with application to the SVHN dataset.

1.3 GMMs with invariance (iGMMs)

As a key concept, we propose an invariance-enforcing extension to vanilla GMMs that we term iGMM. iGMM instances are trained with augmented data and encourage the best-matching GMM component to be identical for all augmented versions of the same sample. Since iGMMs are based on Mahalanobis distances, the best-matching component will automatically get more *dissimilar* to other samples by consequence. Instead of applying iGMMs to image content directly, it is advantageous to use them as layers in a hierarchy together with a vanilla GMM layer, see fig. 1.

1.4 Contributions

This article is a proof-of-concept work operating on simple datasets. The conceptual contribution is a proposal for an invariance-enforcing generalization of the GMM log-likelihood that can be used for SGD training. In addition, we describe a simple way to sample from a trained GMM-iGMM hierarchy. Experimental contributions include a training procedure based on GMM-iGMM hierarchies for the efficient learning of invariances, as well as an analysis of invariance, sampling and outlier detection properties of such hierarchies.

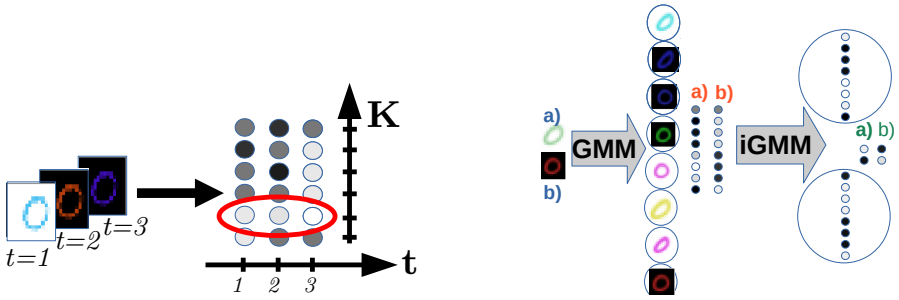

Fig. 1. Left: visualization of iGMM invariance, shown for $A = 3$ augmented versions of a single sample passed through an iGMM with $K = 6$ components, resulting in $AK = 18$ component activities (log-probabilities) color-coded as filled circles. Invariance to augmentations translates into the requirement that one component (red ellipse) should have consistently higher activity for all augmented versions of a sample. The iGMM loss is formulated such that the component with maximal average (taken over $t = 1, 2, 3$) activity is adapted. **Right:** response to different stimuli (blue letters)) in a GMM-iGMM cascade. A GMM with 6 components (centroids shown inside large circles) is directly trained on color-augmented MNIST digits, followed by an iGMM with two components (centroids again in large circles). GMM responses (marked by red letters, strength indicated by brightness) show selectivity to slant, color and polarity. iGMM responses (indicated by green letters) on the other hand just react to different polarities but are invariant to other image properties.

2 Methods

2.1 Data augmentation in general

We assume the existence of several parameterized transformations $\tau_i(\boldsymbol{p})$ such that a training example $\boldsymbol{x}_\nu$ can be augmented as $T_\mathcal{V}(\boldsymbol{x}_\nu) \equiv \tau_1(\boldsymbol{v}_1) \circ \tau_2(\boldsymbol{v}_2) \circ \ldots(\boldsymbol{x}_\nu), \boldsymbol{v}_i \in \mathcal{V}$ according to the set of parameter vectors $\mathcal{V}$. By creating a fixed number A of augmented samples for every $\boldsymbol{x}_\nu$ in a mini-batch of N samples, we create multi-viewed mini-batches of AN samples for training. The indices of all augmented version of sample $\boldsymbol{x}_\nu$ are given by $P(\nu)$, with $|P(\nu)| = A$. Conversely, the source index of an augmented sample $\boldsymbol{x}_n$ is given by $\nu(n)$. The set of parameter vectors is chosen randomly and uniformly from the ranges defined for a particular transformation.

2.2 Transformations for data augmentation

Channel swap either exchanges two random RGB channel in the whole image or does nothing, with each of these four possibilities chosen with equal probability. The parameter vector is either $\boldsymbol{v} = (a, b)^T$ for a channel swap, or $\boldsymbol{v} = (-1, -1)^T$ for no swap.

Brightness inversion is a deterministic transformation which simply performs the operation $x \to 255 - x$ on all channels simultaneously. The parameter vector is a single number indicating whether or not to perform this transformation.

Random image shifts are performed with shifts $\Delta_x, \Delta_y \in [-3, 3]$ using cyclic boundary conditions. The parameter vector is simply $\boldsymbol{v} = (\Delta_x, \Delta_y)^T$.

2.3 Data and preprocessing

All experiments are based on the MNIST and FashionMNIST datasets.
MNIST [7] consists of 60 000 28×28 gray scale images of handwritten digits (0-9). We promote all images to RGB by repeating the single channel 3 times, and multiplying channels by normalized random factors in the $[0, 1]$ interval. Prior to normalization, one randomly chosen factor is multiplied by a random number between 3 and 6 so as to induce stronger colors.
Fashion-MNIST [16] consists of 60.000 images of clothes in 10 categories and is structured like MNIST.

Both datasets are transformed into colored versions of themselves by setting all non-black pixels to a randomly selected shade of red, green or blue. For each sample, we create $A = 5$ augmented versions using channel swap, image shift and brightness inversion operations.

2.4 GMMS with invariance: iGMMs

Vanilla GMMs try to represent the target distribution as a weighted sum of multi-variate normal distributions with centroids $\boldsymbol{\mu}$ and covariance matrices $\boldsymbol{\Sigma}$: $p(\boldsymbol{x}) = \sum_k \pi_k \mathcal{N}_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. A trained GMM outputs the posterior probability $\gamma_k(\boldsymbol{x}_i) = \frac{\pi_k \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$ for each image $\boldsymbol{x}_i$ and GMM component k.

Generalizing the conventional GMM log-likelihood loss, the iGMM loss pushes the best-matching component k^* to be identical for all augmented versions of the same sample $\boldsymbol{x}_\nu$, and to be different for other samples as much as possible:

$$\mathcal{L} = \sum_n \log \sum_k \left(\pi_k \mathcal{N}_k(\boldsymbol{x}_n)\right) \approx \sum_n \log \max_k \left(\pi_k \mathcal{N}_k(\boldsymbol{x}_n)\right) = \sum_n \max_k \log \left(\pi_k \mathcal{N}_k(\boldsymbol{x}_n)\right) \tag{1}$$

$$\rightarrow \sum_\nu \max_k \sum_{n \in A(\nu)} \log(\pi_k \mathcal{N}_k(\boldsymbol{x}_n)). \tag{2}$$

The first expression of eq. (1) represents the incomplete-data loss for a vanilla GMM. Subsequently, we employ the max-component approximation of the sum as outlined in [2], which removes exponentials and brings the max operation in front of the log. In the second row, we generalize to the case of multiple augmented versions of a single sample $\boldsymbol{x}_\nu$. The mean over all augmented versions $\boldsymbol{x}_n$ of the sample $\boldsymbol{x}_\nu$ will have high values if the same component consistently shows high log-probabilities. This is similar to constrastive learning [5], except that we do not need to enforce dissimilarity to other samples due to the distance-based nature of the loss eq. (1).

An iGMM can be applied to image data directly, in which case components tend to averages over augmented samples. This leads to blurred or unrecognizable

sampling results. A better ways of using iGMMs is to apply them to sparse data, such as posterior probabilities computed by a vanilla GMM. This allows to construct a GMM-iGMM sampling chain of hierarchical priors as outlined in section 2.5 on the one hand, and on the other hand ensures that learned iGMM components remain sparse as well, which facilitates interpretability and sampling, please see also fig. 2.

2.5 GMM-iGMM hierarchies

GMM-iGMM hierarchies are trained using a separate loss for the GMM and iGMM layer. In contrast to standard procedure for, e.g., DNNs, both losses are not added to form a single overall loss, but optimized in complete independence. Otherwise, adaptation of GMM parameters would be impacted by the iGMM loss as well, which has proven to prevent convergence. Both "layers" are linked through the GMM posterior probabilities which form the input to the iGMM. Posterior probabilities or responsibilities are computed as $\gamma_k(\boldsymbol{x}) = \frac{p_k(\boldsymbol{x})}{\sum_j p_j(\boldsymbol{x})}$ and are therefore normalized and bounded in the $[0, 1]$ range. Both GMM and iGMM are trained by SGD according to [3], and iGMM adaptation is prevented until the GMM annealing parameter $\sigma(t)$ has reached a value of 0.8, which means that a certain convergence has already been achieved.

Sampling in a GMM-iGMM hierarchy is performed by generating a sample in the iGMM layer, which is used as a prior over component selection probabilities when sampling from the GMM layer. When sampling from a certain GMM component directly, this hierarchical prior mechanism is disabled.

3 Experiments

We conduct the experiments with GMMs, iGMMs and GMM-iGMM hierarchies according to the following principles: most importantly, all optimization is carried out by SGD (not Expectation-Maximization or similar schemes). Image data are converted to vectorial form by simple flattening (e.g., a 28x28x3 images becomes a 2352-dimensional vector). When image augmentation as described in section 2.1 is just performed for training, whereas only unaugmented colored samples are used during evaluation. In hierarchies, the forward pass for one samples (regardless of training or testing) consist always of passing the vector of GMM responsibilities as input to the iGMM. Furthermore, the GMM and iGMM loss are optimized in parallel, and adaptation is inhibited for the iGMM until the GMM has converged sufficiently.

3.1 Preliminary experiment: naive application of iGMMs

For this experiment, we directly train an iGMM instance on MNIST data augmented as described in section 2.3. To illustrate the developed component selectivities, we sample from certain components of this iGMM and display the resulting images. see fig. 2. We can observe that components tend to get selective

Fig. 2. Component selectivities of an iGMM instance that has been directly trained on augmented data, visualized by sampling from these components.

to "average samples", which are mainly blank since polarity-inversed samples will cancel each other out in the average.

3.2 Formation of simple/complex cells in GMM-iGMM hierarchies

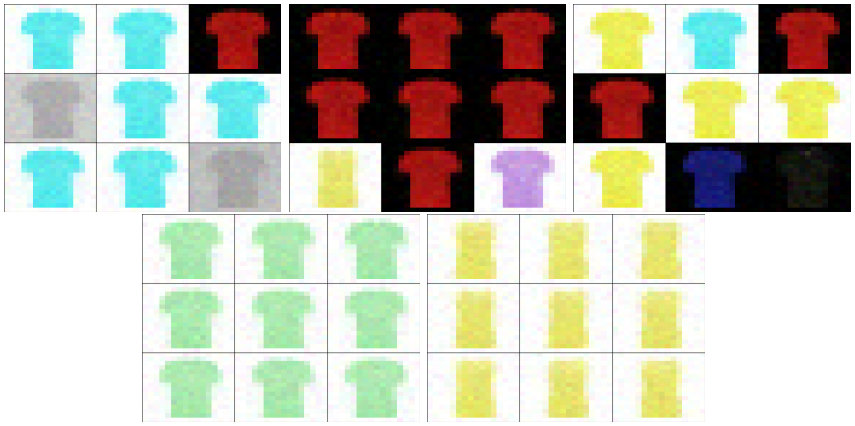

Fig. 3. Component selectivities in a GMM-iGMM hierarchy visualized by sampling from different iGMM (upper row) and GMM (lower row) components after training on colored FashionMNIST. We can observe invariance, to different degrees, for iGMM components but none for GMM components.

Motivated by the negative outcome of section 3.1, we perform a similar experiment with a GMM-iGMM hierarchy as described in section 2.5. Again, we determine component selectivities through selectively sampling from those components, both for the GMM and the iGMM "layer". Sampling from the GMM layer is performed in the standard GMM way, whereas sampling from the iGMM layer proceeds in two steps: first of all, the iGMM is queried for a sample, again in the standard way. This sample is then taken to represent a multinomial distribution over GMM components, from which we draw a sample that identifies

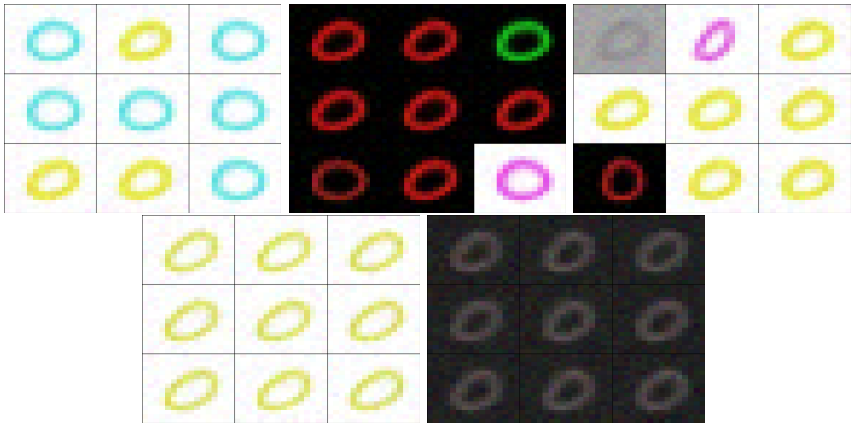

Fig. 4. Component selectivities in a GMM-iGMM hierarchy visualized by sampling from different iGMM (upper row) and GMM (lower row) components after training on colored MNIST. We can observe invariance, to different degrees, for iGMM components but none for GMM components.

a GMM component. This component will produce the final sampling result from the multivariate Gaussian distribution it represents.

In fig. 3 and fig. 4, we observe very interesting behavior: iGMM components are selective to stimuli of multiple polarities and colors, whereas GMM components are selective just for one given configuration w.r.t. these axes. It is furthermore discernible that, although iGMM components do have preferences for, e.g., certain colors or polarities, they always show some degree of invariance. This is reminiscent of the behavior of simple/complex cells in the visual cortex: whereas simple cells respond only to test stimuli of a given polarity, complex cells respond to the same stimuli but with inversed polarity. We find it noteworthy that such behavior arises naturally through unsupervised learning.

3.3 Explicitly measuring invariance

In this section, we introduce am explicit measure of invariance for assessing the differences between GMM and iGMM "layer" in a GMM-iGMM hierarchy. The measure is recorded over the course of training and evaluated on the test set as well. The invariance measure we propose essentially measures the maximal number of times any component is best-matching unit (BMU) for all augmented version of a given sample, averaged over all distinct samples in the the test set. We define the invariance measure based on the posterior probabilities $\gamma(\boldsymbol{x})$:

$$\chi_k(\boldsymbol{x}) = \begin{cases} 1 & k = \arg\max_j \gamma_k(\boldsymbol{x}) \\ 0 & \text{else} \end{cases} \tag{3}$$

Using the notation form section 2.1, we can count the number of times any component was BMU for all augmented version of a single sample as $c_k(\nu) =$

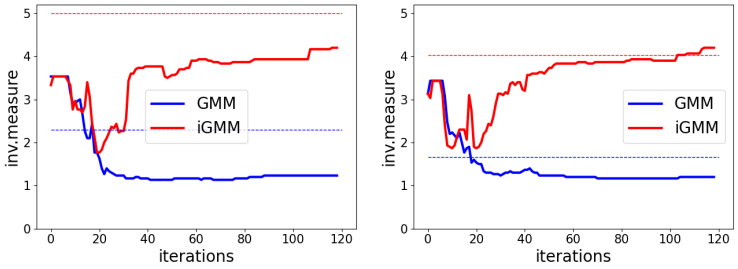

Fig. 5. Invariance measure C for the GMM andf iGMM layer plotted over the course of 120 training epochs for MNIST(left) and FashionMNIST(right). Invariance measures plotted on the respective test sets after epoch 120 are shown as dashed lines. We observe that the iGMM layer always shows higher invariance.

$\sum_{\boldsymbol{x}_n \in P(\nu)} \chi_k(x_{n(\nu)})$. The desired invariance measure C is then found as: $C = \mathbb{E}_\nu \max_k c_k(\nu)$. When training a GMM-iGMM hierarchy on MNIST and FashionMNIST, we obtain the curves depicted in fig. 5, from which we can observe a markedly higher invariance for the iGMM layer.

4 Discussion

From the experiments of the previous section, we conclude that GMM-iGMM hierarchies are capable of capturing simple invariances, and to nevertheless perform faithful sampling in the input domain. The modification of the iGMM loss to reward invariance appears to be beneficial for describing abstract concept with fewer components, as shown by the success of the outlier detection experiment. Some specific topics seem worth highlighting here:

Choice of datasets Since this is a proof.of-concept work, we do not perform experiment on complex real-world datasets such as CIFAR since these are hard to model even for dedicated generators such as VAEs. When moving to deep convolutional hierarchies as outlined in [2], such datasets will within the range of capabilities of our probabilistic models.

Assessment of selectivities through sampling Since probabilistic models always have the capacity to sample, we consider this an interesting and natural tool for visualizing component selectivities. This is in stark contrast to deep neural networks, where one has to resort to complex procedure like deep dreaming to analyze selectivities of specific neurons.

Link to constrastive supervised learning The approach to training iGMMs is similar to the one taken by constrastive learning approaches. Certainly, data augmentation is a common property, but also the explicit modeling of invariances. We believe that by generalizing GMM-iGMM hierarchies to deeper convolutional hierarchies, we can model invariances at a local image level and learn feature compared to those obtained by contrastive learning.

Reproducibility We will release TF2-based Python code to reproduce all experiments.

References

1. Butz, C.J., Oliveira, J.S., dos Santos, A.E., Teixeira, A.L.: Deep convolutional sum-product networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3248–3255 (2019)
2. Gepperth, A.: A new perspective on probabilistic image modeling. In: International Joint Conference on Neural Networks(IJCNN) (2022)
3. Gepperth, A., Pfülb, B.: Gradient-based training of gaussian mixture models in high-dimensional spaces (2020)
4. Grathwohl, W., Chen, R.T., Bettencourt, J., Sutskever, I., Duvenaud, D.: Ffjord: Free-form continuous dynamics for scalable reversible generative models. arXiv preprint arXiv:1810.01367 (2018)
5. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Advances in neural information processing systems **33**, 18661–18673 (2020)
6. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018)
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
8. Peharz, R., Lang, S., Vergari, A., Stelzner, K., Molina, A., Trapp, M., Van den Broeck, G., Kersting, K.: Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In: International Conference on Machine Learning. pp. 7563–7574. PMLR (2020)
9. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: International conference on machine learning. pp. 1530–1538. PMLR (2015)
10. Richardson, E., Weiss, Y.: On GANs and GMMs. Advances in Neural Information Processing Systems **2018-December**(NeurIPS), 5847–5858 (2018)
11. Tang, Y., Salakhutdinov, R., Hinton, G.: Deep mixtures of factor analysers. Proceedings of the 29th International Conference on Machine Learning, ICML 2012 **1**, 505–512 (2012)
12. Van Den Oord, A., Schrauwen, B.: Factoring variations in natural images with deep Gaussian mixture models. Advances in Neural Information Processing Systems **4**(January), 3518–3526 (2014)
13. Van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. arXiv preprint arXiv:1904.07734 (2019)
14. Viroli, C., McLachlan, G.J.: Deep Gaussian mixture models. Statistics and Computing **29**(1), 43–51 (2019). https://doi.org/10.1007/s11222-017-9793-z
15. Wolfshaar, J., Pronobis, A.: Deep generalized convolutional sum-product networks. In: International Conference on Probabilistic Graphical Models. pp. 533–544. PMLR (2020)
16. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms pp. 1–6 (2017), http://arxiv.org/abs/1708.07747
17. Zhang, L., Goldstein, M., Ranganath, R.: Understanding failures in out-of-distribution detection with deep generative models. In: International Conference on Machine Learning. pp. 12427–12436. PMLR (2021)