SAR Automatic Target Recognition Based on Supervised Deep Variational Autoencoding Model

DANDAN GUO
BO CHEN, Senior Member, IEEE
MEIXI ZHENG
HONGWEI LIU, Member, IEEE
Xidian University Xi'an, China

Deep learning has been gradually used to solve SAR image classification problems for its desired performance on various recognition problems. A deep variational autoencoding model (DVAEM), that constructs a multi-stochastic-layer generative network (decoder) and variational inference network (encoder), can be employed to build a flexible and interpretable model for the SAR image target recognition task. It is scalable in the training phase and fast in the testing stage, and can extract the hierarchical structured and interpretable features from SAR images. However, the current DVAEM extracts the features of SAR images unsupervisedly, without incorporating the label information, and may fail to extract discriminative representations for the recognition task. In this article, to jointly model SAR images and their corresponding labels, we further propose supervised DVAEM with Euclidean distance restriction (rs-DVAEM), which enhances the discriminative power of latent representations of SAR images. Notably, our proposed rs-DVAEM combines the flexibility of DVAEM in describing the SAR images and the discriminative power of supervised models. Experimental results on the moving and stationary target acquisition and recognition public dataset demonstrate the effectiveness of the proposed method.

Manuscript received September 16, 2020; revised March 12, 2021; released for publication May 31, 2021. Date of publication July 14, 2021; date of current version December 6, 2021.

DOI. No. 10.1109/TAES.2021.3096868

The work of Bo Chen was supported in part by the National Natural Science Foundation of China under Grant 61771361, in part by 111 Project under Grant B18039, and in part by the Program for Oversea Talent by Chinese Central Government. The work of Hongwei Liu was supported in part by the NSFC for Distinguished Young Scholars under Grant 61525105 and in part by Shaanxi Innovation Team Project.

Authors' address: The authors are with National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China, E-mail: (gdd xidian@126.com; bchen@mail.xidian.edu.cn; meixizheng1110@163.com; hwliu@xidian.edu.cn). *(Corresponding author: Bo Chen.)*

I. INTRODUCTION

Synthetic aperture radar (SAR) is widely used in military and civil fields for its advantages of full-time operation, all-weather ability and robustness to environmental conditions. Since the SAR image contains abundant discriminative information, such as shape and scatterer distribution, automatic target recognition (ATR) of SAR images has obtained intensive attention [1]–[10]. However, the unique characteristics of the SAR image formation process, such as specular reflection, typical low-resolution and speckle noise, cause difficulties in the ATR of SAR images. With the development of SAR imagery technology, lots of methods have been applied to the ATR of SAR images. Early work is devoted to obtaining favorable results with different classification criteria, including particle swarm optimization (PSO) with Hausdorff distance (HD) [11] and template matching [12]–[14], etc. These approaches have gained desired recognition results when targets have consistent appearances, but they would fail with highly variable target appearances. In contrast, some methods aim to exhibit the objects of SAR images into another subspace to gain distinctive features, such as principal component analysis (PCA) [15], nonnegative matrix factorization (NMF) [16], [17], sparse representation (SR) [3], [18], [19], and scattering center models [20]. Besides, some statistic approaches, including the hidden Markov model [21], conditional Gaussian method [22] and others, are utilized for SAR ATR. In summary, these conventional approaches have achieved good recognition results for SAR images but still have some limitations. They elaborately designed hand-crafted feature extractors or established complicated classification systems, requiring great effort from human experts [23]. Besides, they ignored the human learning system, which can extract hierarchical representations and be helpful for the interpretation of SAR images [9].

Recently, deep learning algorithms are becoming popular in various areas for their ability to extract multilayer nonlinear features [24]. For example, restricted Boltzmann machine (RBM) [25] and autoencoder (AE) [26] are two commonly used feature extracted methods in deep learning algorithms, and convolutional neural network (CNN) [27] is an end-to-end supervised deep learning framework. Motivated by the successful application of deep learning in optical image recognition, some researchers have introduced deep learning algorithms into problems of remote sensing images [9], [28]. For instance, RBM structure in [29] and CNN with sparse AE in [28] have been utilized to learn features for SAR ATR automatically. Considering the scarcity and concealment of military targets, SAR images for training deep structures are insufficient, which may be prone to severe overfitting [23]. Although the CNN for SAR ATR with domain-specific data augmentation has been explored in [30] to alleviate this problem, researchers expect to improve the algorithm without such man-made augmentation rules. Besides, Ma *et al.* [31] introduced a spatial updated deep AE and a collaborative representation-based classification method. Chen *et al.* [32] presented

all-convolutional networks, which only consist of sparsely connected layers, to reduce the number of free parameters and strengthen the general performance of the network. Deng *et al.* [9] bounded the original AE algorithm with a restriction based on Euclidean distance to extract separable features with the limited training SAR images. Although significant efforts have been made, these commonly used deep models for SAR ATR do not consider the kind of variability observed in highly structured data, which may result in the need for large amounts of SAR images. Besides, the learned neural network representations are generally inaccessible and uninterpretable to humans. Recently, deep probabilistic generative models have been proposed and popular in computer vision and pattern recognition, such as variational autoencoding (VAE) [33], which can learn useful interpretable latent features and generate meaningful data. As a result, VAE can be adopted to extract the features of SAR images due to their ability to learn the distribution inside the input data. Although VAE constructs the variational encoder with deep neural networks, it only utilizes shallow generative models with only a single stochastic hidden layer for data generation as the decoder.

For SAR ATR, to overcome some deep models' need for large amounts of SAR images and move beyond VAE adopting shallow generative models, we focus on developing the deep variational autoencoding model (DAVEM), which is composed of a multi-stochastic-layer generation model and an inference model. Recently, the Poisson gamma belief network (PGBN) [34], a multi-stochastic-layer probabilistic generative model, has been successfully used in text and natural image classification. Different from existing deep models, PGBN factorizes a count matrix under the Poisson likelihood, and factorizes the shape parameters of the gamma-distributed hidden units of each layer into the product of a connection weight matrix and the gamma hidden units of the next layer, which extracts strong nonlinearity and readily interpretable hierarchical latent features. For real-time processing at the test stage, Zhang *et al.* [35] further generalize a novel inference network for the PGBN, which is still limited in the text data or natural images.

This article builds DVAEM for SAR ATR, which adopts PGBN [34] as the generative model (decoder) and the variational inference network [35] as the inference model (encoder). As the unsupervised probabilistic model, DVAEM can extract hierarchical representations from SAR images when performing in the absence of label information. Although the features unsupervisedly learned by DVAEM can be fed into a downstream classifier to realize the SAR target recognition task, it is often more beneficial to incorporate the target label information into the model [36]. To this end, we then develop an end-to-end supervised DVAEM (s-DVAEM) by introducing the label information into DVAEM, which combines the flexibility of DVAEM in describing the SAR images and the discriminative power of supervised models. To promote the mined features with s-DVAEM from the same target to be more similar than those from different classes, we further bind the s-DVAEM with a restriction based on Euclidean distance on the features following [9], referred to as rs-DVAEM. Significantly, the proposed rs-DVEAM not only retains the advantages of DVEAM but also explores the strongly discriminative principle of the classifier with Euclidean distance restriction. The key difference from ours is that Deng *et al.* [9] utilize Euclidean distance restriction based on stacked AE to extract hidden units and train a downstream classifier to categorize the features in a two-stage manner. Hence, the classification results in [9] cannot affect the parameters of the feature extractor. However, our proposed rs-DVAEM learns the hierarchical nonnegative features of SAR images under the supervised deep probabilistic framework, which trains the feature extractor and classifier jointly. Given a new SAR image at the testing stage, our rs-DVSEM can directly predict its label with the trained parameters of the inference model and classifier. Compared with the deterministic mapping in CNN or AE, our proposed DAVEM-based models perform distribution estimation and learn the distribution inside the input data with the probabilistic framework, which brings more robustness to the smaller training set, achieving the acceptable recognition performance even with limited SAR images. For SAR ATR, there are fundamental differences between VAE and our proposed models in terms of the decoder and encoder, not to mention the introduction of label information at the training stage. For our models, gamma distribution rather than Gaussian distribution is utilized to model the latent variables in the decoder and approximated in the encoder, where the gamma distribution density function has the highly desired strong nonlinearity for deep learning.

The experimental results on the MSTAR dataset express the effectiveness of our proposed methods. The contributions of this article include:

1) Adopting a novel deep probabilistic framework (DVAEM) to extract hierarchical structured features from SAR Images.
2) Proposing the supervised models (s-DVAEM and rs-DVAEM) with Euclidean distance restriction and direct supervision on the features across all layers and improving the performance of SAR ATR.
3) Visualizing various aspects of the SAR images from general to specific and revealing how the representations in each layer are related to each other, which is of paramount interest in SAR image interpretation.

The rest of this article is organized as follows. In Section II, we describe the deep variational autoencoding Model. In Section III, we introduce the supervised model for SAR ATR with Euclidean distance restriction. In Section IV, experimental results in comparison with other methods of SAR ATR are displayed. Finally, Section V concludes this article.

II. UNSUPERVISED MODEL

As a deep probabilistic framework, unsupervised DVAEM aims to extract the hierarchical features to represent the data, which can be fed into the downstream tasks.

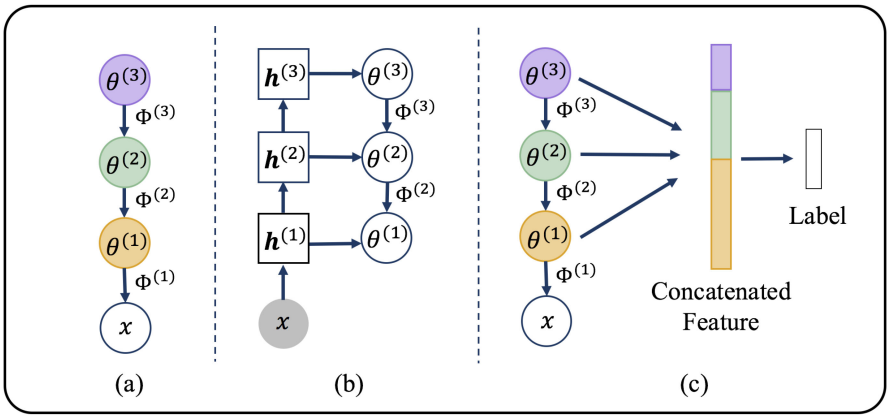


Fig. 1. (a) Generative model of DVAEM. (b) Variational inference model of DVAEM. (c) Generative model of supervised DVAEM (s-DVAEM) and s-DVAEM with Euclidean distance restriction (rs-DVAEM).

In this section, we will first introduce the generative model of DVAEM in Fig. 1(a). Then, we will describe how to infer the unsupervised generative model in Fig. 1(b), based on the idea of variational autoencoder.

A. Generative Model of DVAEM

In order to extract the hierarchical and interpretable features from SAR images, here we generalize the PGBN [34] to DVAEM. Shown in Fig. 1(a) is the graphical representation of a three-hidden-layer generative model of DVAEM. Specifically, we consider $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N \in \mathbb{R}_+^{V\times N}$ as the observed dataset, which consists of N independent and identically distributed (IID) V-dimensional SAR images. Given the deep architecture with L hidden layers, we denote $\boldsymbol{\theta}_n^{(l)} \in \mathbb{R}_+^{K_l\times 1}$ as the latent features of $\boldsymbol{x}_n$ at layer l. The latent features $\boldsymbol{\theta}_n^{(l)}$ of DVAEM, from top to bottom, can be generated as

$$\begin{aligned}
\boldsymbol{\theta}_n^{(L)} &\sim \text{Gam}\left(\boldsymbol{r}, 1/c_n^{(L+1)}\right)\\
&\cdots\\
\boldsymbol{\theta}_n^{(l)} &\sim \text{Gam}\left(\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, 1/c_n^{(l+1)}\right)\\
&\cdots\\
\boldsymbol{\theta}_n^{(1)} &\sim \text{Gam}\left(\boldsymbol{\Phi}^{(2)}\boldsymbol{\theta}_n^{(2)}, 1/c_n^{(2)}\right)
\end{aligned} \tag{1}$$

where $\boldsymbol{\Phi}^{(l)} \in \mathbb{R}_+^{K_{l-1}\times K_l}$ denotes the factor loading matrix or dictionary, K_l the number of hidden units at layer l, respectively, and $K_0 = V$. The generative model factorizes the gamma shape parameter of $\boldsymbol{\theta}_n^{(l)}$ into the product of $\boldsymbol{\Phi}^{(l+1)}$ and gamma distributed hidden units $\boldsymbol{\theta}_n^{(l+1)}$ of layer $l+1$ for $l \in L-1$; at the top layer, the vector $\boldsymbol{r} = \{r_1, \ldots, r_{K_L}\}$ can be shared as the gamma shape parameter of hidden units $\boldsymbol{\theta}_n^{(L)}$; and $\{c_n^{(l+1)}\}_{l=1}^L$ are gamma scale parameters. Following [34], not only for the identifiability of the hidden units but also for the ease of inference and interpretability, the Dirichlet distribution is imposed on each column of $\boldsymbol{\Phi}^{(l)}$, i.e., $\boldsymbol{\phi}_k^{(l)} \sim \text{Dir}(\eta^{(l)}, \ldots, \eta^{(l)})$ for $l \in L$, where $\eta^{(l)}$ denotes the prior of $\boldsymbol{\phi}_k^{(l)}$, and $\phi_{k_1k}^{(l)}$ indicates the connection strength between node (basis vector) k_1 of layer $l-1$ and node k of layer l. We notice that Dirichlet distributed $\boldsymbol{\phi}_k^{(l)}$ is a vector with the simplex-constrained, which means $\phi_{k_1,k}^{(l)} \geqslant 0$ and $\sum_{k_1=1}^{K_{l-1}} \phi_{k_1,k}^{(l)} = 1$. Besides, the Dirichlet prior placed on $\boldsymbol{\phi}_k^{(l)}$ at each layer can promote sparsity due to its nonnegative property.

The conjugacy between the gamma and Poisson distributions makes it convenient to construct the hierarchical Bayesian model amenable to posterior simulation. Therefore, PGBN [37] utilizes the Poisson distribution or its corresponding link functions to model the different data types, including count, binary, and nonnegative real data. For example, if the observations are nonnegative continuous vector $\boldsymbol{x}_n \in \mathbb{R}_+^V$, where the SAR image fits this case, we can factorize it using Poisson randomized gamma (PRG) distribution, formulated as

$$\boldsymbol{x}_n \sim \text{Gam}\left(\boldsymbol{a}_n, 1/b\right), \boldsymbol{a}_n \sim \text{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right) \tag{2}$$

where $b > 0$, $\boldsymbol{a}_n$ denotes the augmentable variable, and we omit more details about the PRG distribution [37]. Despite the desired ability to depict nonnegative real-valued data, the PRG distribution may be time-consuming in real-world applications for the iterative procedure to infer latent counts $\boldsymbol{a}_n$. To improve the computation power in the training stage, we can model the SAR image data with the Poisson distribution by directly discretizing the observed data to counts (i.e., nonnegative integers) before training, which can avoid sampling the latent counts in each iteration. Hence, rather than modeling the SAR image with (2), we assume that

$$\lfloor \mu \boldsymbol{x}_n \rfloor \sim \text{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right) \tag{3}$$

where the scaling factor μ controls the fineness of this discretization as a constant greater than zero. There is a tradeoff between the benefits of the PRG distribution and the restriction of computation resource. We summarize the generative model of rs-DVAEM in Fig. 1(a).

Since the gamma distribution density function has the highly desired strong nonlinearity for deep learning, our proposed model builds the nonlinearity based on the deep probabilistic framework with gamma-distributed hidden units. Therefore, our model can extract the multilayer factor

loading matrices and gamma-distributed features both with strictly nonnegative elements, resulting in a natural sparseness. It is reasonable since the intensity value represented by each SAR image pixel is nonnegative. Based on the deep probabilistic generative framework, we notice that the information of the whole SAR image dataset can be compressed into the inferred network $\{\Phi^{(1)}, \ldots, \Phi^{(L)}\}$. Specifically, the inferred basis vector $\boldsymbol{\phi}_k^{(l)}$ at hidden layer l can be directly visualized as $[\prod_{l=1}^{l-1} \boldsymbol{\Phi}^{(l)}]\boldsymbol{\phi}_k^{(l)}$, which tends to be very basic in the bottom layer and more general with the increase of the layers [34], [35], [38].

B. Inference Model of DVAEM

Under the hierarchical generative model of DVAEM, given the factor loading matrix $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$, the marginal likelihood of the SAR images $\mathbf{X}$ consisting of N IID SAR images is defined as

$$P\left(\mathbf{X}|\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}\right) = \int \prod_{n=1}^{N} p\left(\boldsymbol{x}_n|\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right) \left[\prod_{l=1}^{L} p\left(\boldsymbol{\theta}_n^{l}|\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, 1/c_n^{(l+1)}\right)\right] d\boldsymbol{\theta}_{1:N}^{(1:L)} \quad (4)$$

where $\boldsymbol{\theta}_n^{(1:L)}$ denotes the hierarchical features from the nth SAR image. Conditioned on the SAR images and hyperparameters, the inference task here is to find the weight matrices $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$ and hierarchical features $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1,n=1}^{L,N}$. Although it is difficult to infer the DVAEM due to the coupling of $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1,n=1}^{L,N}$ with $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$, the latent variables of the DVAEM can be trained via the upward–downward Gibbs sampler of [37], with a variety of variable augmentation techniques. However, the Gibbs sampler requires processing all SAR images in each iteration and thus has limited scalability. To make our model scalable to a larger SAR dataset, a topic-layer-adaptive stochastic gradient Riemannian Markov chain Monte Carlo (TLASGR-MCMC) described in [35] and [38] is employed to update the global parameters $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$. In detail, we can sample $\boldsymbol{\phi}_k^{(l)}$, the kth column of the factor loading matrix $\boldsymbol{\Phi}^{(l)}$, as

$$\begin{aligned}\left(\boldsymbol{\phi}_k^{(l)}\right)_{i+1} &= \left[\left(\boldsymbol{\phi}_k^{(l)}\right)_i + \frac{\varepsilon_i}{M_k^{(l)}}\left[\left(\rho\tilde{\boldsymbol{x}}_{:k\cdot}^{(l)} + \eta^{(l)}\right) - \left(\rho\tilde{x}_{\cdot k\cdot}^{(l)} + K_{l-1}\eta^{(l)}\right)\left(\boldsymbol{\phi}_k^{(l)}\right)_i\right] \right. \\ &\quad \left. + \mathcal{N}\left(0, \frac{2\varepsilon_i}{M_k^{(l)}}\mathrm{diag}(\boldsymbol{\phi}_k^{(l)})_i\right)\right]_{\angle} \end{aligned} \quad (5)$$

where ε_i denotes the learning rate at the ith iteration, ρ the ratio of the dataset size N to the mini-batch size, $\eta^{(l)}$ the above-mentioned prior of $\boldsymbol{\phi}_k^{(l)}$, and the symbol "$\cdot$" denotes summing over the corresponding index. Both $\tilde{\boldsymbol{x}}_{:k\cdot}^{(l)}$ and $\tilde{x}_{\cdot k\cdot}^{(l)}$ come from the augmented latent counts $\mathbf{X}^{(l)}$, described in [38, Lemma 3.1]. And $M_k^{(l)}$ is calculated using the estimated Fisher information matrix (FIM), which can be found in (18) and (19) of [38]. Besides, $[.]_{\angle}$ denotes the simplex constraint, whose implementation details can be found in [39, Examples 1–3]. We provide more details about augmented latent counts $\mathbf{X}^{(l)}$, $M_k^{(l)}$, and the implementation of simplex constraint $[.]_{\angle}$ in the Appendix VII for the sake of completeness.

Although the effectiveness, TLASGR-MCMC usually need to perform a sampling-based iterative algorithm inside each iteration at the testing stage, limiting the model for real-time processing. Inspired by variational autoencoder that constructs an inference network to map the observations directly to their latent representations, we introduce a novel variational inference method [35] to efficiently learn the multilayer nonnegative representations $\boldsymbol{\theta}_n^{(1:L)}$ for SAR images, given $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$. Specifically, we employ a factorized distribution $Q = \prod_{n=1}^{N}\prod_{l=1}^{L} q(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)})$ to approximate the Gibbs sampling in [34, Eq. (14)], which can be summarized as $P = \prod_{n=1}^{N}\prod_{l=1}^{L} p(\boldsymbol{\theta}_n^{(l)}|\mathbf{X}^{(l)}, \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)})$. Thus, the objective function becomes the evidence lower bound (ELBO) [35] of the log marginal likelihood in (II-B), formulated as

$$\begin{aligned} L &= \sum_{n=1}^{N} E_{q\left(\boldsymbol{\theta}_n^{(1)}\right)}\left[\ln p\left(\boldsymbol{x}_n|\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right)\right] \\ &\quad - \sum_{n=1}^{N}\sum_{l=1}^{L} E_{q\left(\boldsymbol{\theta}_n^{(l)}\right)}\left[\ln \frac{q\left(\boldsymbol{\theta}_n^{(l)}\right)}{p\left(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, 1/c_n^{(l+1)}\right)}\right] \end{aligned} \quad (6)$$

where the first term encourages a good fit to the observed SAR images, and the second term aims to fit the latent representations of SAR images at each layer, which also represents the Kullback–Leibler (KL) divergence between the approximate posterior and prior distributions. Inspired by the instructive downward information propagation in generative model of Fig. 1(a) and Gibbs sampling in [34], a Weibull distribution based inference network following Zhang *et al*. [35] can be constructed as

$$q\left(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}\right) \sim \mathrm{Weibull}\left(\boldsymbol{k}_n^{(l)} + \boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, \boldsymbol{\lambda}_n^{(l)}\right) \quad (7)$$

where the variational parameters $\boldsymbol{k}_n^{(l)}$ and $\boldsymbol{\lambda}_n^{(l)}$ of $\boldsymbol{\theta}_n^{(l)}$ are both nonlinearly transformed with neural networks from the hidden units $\boldsymbol{h}_n^{(l)}$ of layer l, expressed as

$$\boldsymbol{k}_n^{(l)} = \ln\left[1 + \exp\left(\mathbf{W}_{hk}^{(l)}\boldsymbol{h}_n^{(l)} + \boldsymbol{b}_1^{(l)}\right)\right] \quad (8)$$

$$\boldsymbol{\lambda}_n^{(l)} = \ln\left[1 + \exp\left(\mathbf{W}_{h\lambda}^{(l)}\boldsymbol{h}_n^{(l)} + \boldsymbol{b}_2^{(l)}\right)\right] \quad (9)$$

where hidden units $\boldsymbol{h}_n^{(l)}$ are deterministically nonlinearly transformed from SAR image $\boldsymbol{x}_n$, stated as

$$\boldsymbol{h}_n^{(l)} = \ln\left[1 + \exp\left(\mathbf{W}_{xh}^{(l)}\boldsymbol{h}_n^{(l-1)} + \boldsymbol{b}_3^{(l)}\right)\right] \quad (10)$$

where $\boldsymbol{h}_n^{(0)} = \boldsymbol{x}_n$. Using the reparameterization trick, we can sample the Weibull distributed hidden units in (7) as

$$\boldsymbol{\theta}_n^{(l)} = \boldsymbol{\lambda}_n^{(l)}\left(-\ln(1-\boldsymbol{\epsilon}_n^{(l)})\right)^{1/\boldsymbol{k}_n^{(l)}}, \boldsymbol{\epsilon}_n^{(l)} \sim \mathrm{Uniform}(0, 1). \quad (11)$$

Based on (11), the gradient of the ELBO with respect to parameters of the inference network can be evaluated with low variance; see Appendix IX for more details. The

inference network of DVAEM is depicted in Fig. 1(b). Similar to [35], combining the TLASGR-MCMC and variational inference network described above, we provide a hybrid stochastic-gradient MCMC/autoencoding variational inference for extracting the hierarchical features of SAR images. At the training stage, global parameters $\mathbf{\Phi}^{(1:L)}$ from generative model can be updated via TLASGR-MCMC in (5), and the neural network parameters $\mathbf{\Omega} = \{\mathbf{W}_{xh}^{(l)}, \mathbf{W}_{hk}^{(l)}, \mathbf{W}_{h\lambda}^{(l)}, \boldsymbol{b}_1^{(l)}, \boldsymbol{b}_2^{(l)}, \boldsymbol{b}_3^{(l)}\}_{l=1}^{L}$ from inference model can be updated via SGD by maximizing the ELBO in (6). At the testing stage, with the updated inference network, we can directly obtain the conditional posteriors of features for a new SAR image, rather than performing a large number of iterations.

III. SUPERVISED MODEL

A. Supervised DVAEM

As an unsupervised model, the DVAEM discussed above has the ability to learn the hierarchical representations of SAR images, and can be trained under the condition of no class information. To make label predictions, we can further utilize the multilayer features together to train a downstream classifier. However, it is often beneficial to learn a joint "end-to-end" model that considers both the SAR images and corresponding labels to discover more discriminative representations. Thus, we now introduce a supervised DVAEM (s-DVAEM), learning multilayer features that are good for both SAR image generation and classification. Assume $y_n \in \{1, 2, \ldots, C\}$ is the label of SAR sample $\boldsymbol{x}_n$, where C denotes the total number of classes of targets. By setting p_{cn} as the probability of the SAR image $\boldsymbol{x}_n$ classified to label c and $\sum_{c=1}^{C} p_{cn} = 1$, the label y_n can be generated from a categorical distribution, written as

$$p(y_n|\boldsymbol{x}_n) = \prod_{c=1}^{C} p_{cn}^{\mathrm{I}\{y_n=c\}} \tag{12}$$

where $\mathrm{I}\{y_n = c\}$ indicates whether y_n equals to c.

Considering the introduced model can learn the deep structure from the SAR images, whose inferred weight matrices at different layers reveal different levels of abstraction, we thus enforce direct supervision for the features across all layers. When we concatenate the features into a vector $\mathbf{S}_n = [(\boldsymbol{\theta}_n^{(1)}, \ldots, \boldsymbol{\theta}_n^{(L)})] \in \mathbb{R}^{\sum_{l=1}^{L} K_l}$, the probability p_{cn} is computed with the softmax function, formulated as

$$p_{cn} = \frac{e^{\boldsymbol{w}_c^T \mathbf{S}_n}}{\sum_i^C e^{\boldsymbol{w}_i^T \mathbf{S}_n}} \tag{13}$$

where $\mathbf{W}_{sy} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_C]$ denotes the classifier weight matrix and the label likelihood in (12) can be rewritten as $p(y_n|\mathbf{W}_{sy}, \mathbf{S}_n)$.

We illustrate the generative model for both the observed SAR images and labels in Fig. 1(c). We can train s-DVAEM by maximizing the ELBO of the joint likelihood of the SAR images and labels, expressed as

$$L_s = L + \sum_{n=1}^{N}\sum_{l=1}^{L} E_{q(\boldsymbol{\theta}_n^{(l)})}\left[\ln p(y_n|\mathbf{W}_{sy}, \mathbf{S}_n)\right] \tag{14}$$

where L represents the ELBO of SAR images in (6), and the second term encourages a good fit to the labels of SAR images, denoted as L_y. The inference model parameters $\mathbf{\Omega}$ can be updated by maximizing L_s in (14) and the classifier weight matrix $\mathbf{W}_{sy}$ can be inferred by maximizing the second term in (14), i.e., L_y, whose details are provided in Appendix XI. Because of the concatenation of features across all layers, the supervised model improves the distinctiveness and robustness of the extracted latent representations [40].

B. Euclidean Distance Restriction on Supervised DVAEM

SAR images, whose characteristics vary rapidly with the tiny changes of posture [3], are formed from specular reflections of a coherent source, making SAR images of the same target quite different with different aspect angles. Therefore, to encourage the similarity of inferred features from the same target at different aspect angles, we further introduce the Euclidean distance restriction, motivated by [9]. Specifically, the extracted features from the same target are expected to be more similar than those from different classes. By adding the restriction into the ELBO in (14), we can maximize the loss function of our proposed restricted s-DVAEM (rs-DVAEM), expressed as

$$L_{rs} = L_s - \beta \sum_{l=1}^{L}\sum_{i,j=1}^{N} E_{q\left(\boldsymbol{\theta}_i^{(l)}\right)q\left(\boldsymbol{\theta}_j^{(l)}\right)}\left[\left(||\overline{\boldsymbol{\theta}_i^{(l)}} - \overline{\boldsymbol{\theta}_j^{(l)}}||_2 - S_{ij}\right)^2\right] \tag{15}$$

where β denotes the weight of restriction, $\overline{\boldsymbol{\theta}_i^{(l)}}$ the modular normalization of $\boldsymbol{\theta}_i^{(l)}$, and S_{ij} restricts the Euclidean distance of features between sample $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. If training SAR images $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the same target, we set $S_{ij} = 0$; otherwise $S_{ij} = \alpha$, which is a constant greater than zero. Based on this penalty term, hierarchical features from the same target are more similar because of their shorter distances in feature space, while those from the different targets have bigger difference. Similar with s-DVAEM, we can also illustrate the generative model of rs-DVAEM in Fig. 1(c).

Without training a downstream classifier additionally like in [9], both the feature extractor and classifier in rs-DVAEM are trained simultaneously. Besides, different from [9] where the restricted hidden units are updated with point estimate via SGD, our hierarchical features are sampled from the conditional posterior $q(\boldsymbol{\theta}_n^{(l)})$ using (11), producing the probabilistic uncertainty. We describe the hybrid TLASGR-MCMC/VAE inference algorithm for rs-DVAEM in Algorithm 1. As shown in Fig. 2, we display the overall workflow of our proposed rs-DVAEM in training and testing stages. To sum up, at the training stage, the MSTAR training dataset is employed to learn the parameters of the rs-DVAEM model, including the generative model parameters $\{\mathbf{\Phi}^{(l)}\}_{l=1}^{L}$, classifier weight matrix $\mathbf{W}_{sy}$, and the inference

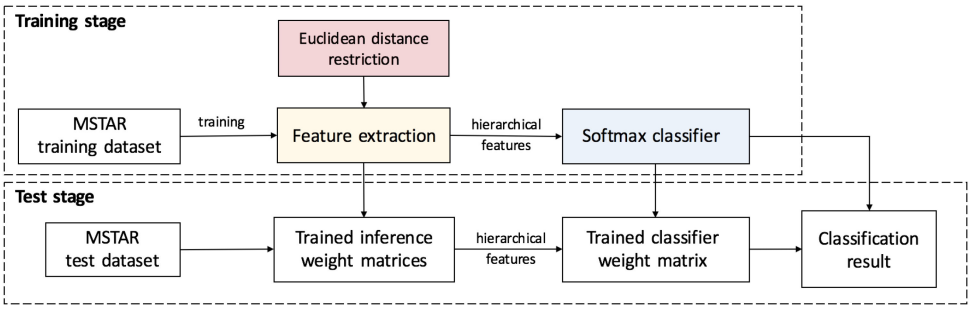


Fig. 2. Overall workflow of training and testing stages.

Algorithm 1: Hybrid TLASGR-MCMC and Variational Inference for rs-DVAEM.

Set mini-batch size M, the number of layer L, the width of layer K_l, and hyperparameters.
Initialize inference model parameters $\mathbf{\Omega}$, generative model parameters $\{\mathbf{\Phi}^{(l)}\}_{l=1}^{L}$, and classifier matrix $\mathbf{W}_{sy}$.
for $iter = 1, 2, \cdots$ **do**
  Randomly select a mini-batch of SAR images and their corresponding labels to form a subset $\mathbf{D} = \{\boldsymbol{x}_m, y_m\}_{1,M}$;
  Draw random noise $\{\boldsymbol{\epsilon}_m^{(l)}\}_{m=1,l=1}^{M,L}$ from uniform distribution (11); sample hidden units $\boldsymbol{\theta}_m^{(l)}$ from (11);
  Compute subgradient $g_{\mathbf{\Omega}} = \nabla_{\mathbf{\Omega}} L_{rs}$ according to (15) and (27) in Appendix B, and update $\mathbf{\Omega}$ using subgradient $g_{\mathbf{\Omega}}$; Compute subgradient $g_{\mathbf{W}_{sy}} = \nabla_{\mathbf{W}_{sy}} L_y$ according to the second term in (14) and (30) in Appendix C, and update $\mathbf{W}_{sy}$ using subgradient $g_{\mathbf{W}_{sy}}$;
  for $l = 1, 2, \ldots, L$ and $k = 1, 2, \ldots, K_l$ **do**
    Update $M_k^{(l)}$ with (21) in Appendix A; then $\boldsymbol{\phi}_k^{(l)}$ with (5);
  end for
end for
Return global parameters $\{\mathbf{\Omega}, \mathbf{W}_{sy}, \{\mathbf{\Phi}^{(l)}\}_{l=1}^{L}\}$.

weight matrices $\mathbf{\Omega}$. Once the rs-DVAEM is trained, the test image $\boldsymbol{x}_n$ is fed into the learned inference network, producing the hierarchical features $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^{L}$ directly without iterative sampling. Then, the extracted features are further introduced into the learned classifier, resulting in the predicted label $\hat{y}_n$.

IV. EXPERIMENT

A. MSTAR Dataset

The MSTAR public dataset [41] with both three-target data and ten-target, containing X-band SAR images with 1 ft × 1 ft resolution, is utilized to evaluate the performance of the proposed method. The three-target dataset contains BMP2 infantry combat vehicle, BTR70 armored personnel carrier, and T72 tank, where the BMP2 and T72 have variants with different serial numbers. Following the suggested experiment settings in [9], [42], [43], 698 images captured at an depression angle of 17° from serial number C21 for BMP2 and 132 for T72 are used for training and 1365 images at 15° from all the serial numbers are used for testing. Such experiment setting aims to verify the generalization of the proposed model, namely evaluating the classification system when recognizing objects with a 2° difference in elevation angle and also recognizing variants that are not in the training samples. The details of the three-target data in our experiments are summarized in Table I. In addition to the three target data, the ten-target dataset also contains the data from another seven ground vehicles. Similarly, the SAR images at depression angle 17° without variants are utilized as the training data, while all images at depression angle 15° as the testing data. We list the details of the ten-target data in Table II.

TABLE I
Detailed Type and Number Used for Three-Target MSTAR Dataset

Target	BMP2			BTR70	T72		
Type	C21	9563	9566	C71	132	812	S7
Train samples	233	0	0	233	232	0	0
Test samples	196	195	196	196	196	195	191

The original SAR image of 128 × 128 pixels is cropped around the center of the image to 64 × 64 pixels, which contains the target and a few backgrounds, reducing the computational complexity. Since the amplitudes of all images differ a lot in the MSTAR dataset, which may conceal their differences between the targets [9], we then adopt the normalization preprocessing method for training and test sets. Each SAR image in both training and test sets is discretized with the scaling factor μ and reshaped to a column vector.

B. Experimental Setup

We consider a three-layer rs-DVAEM (original DVAEM + supervised learning + Euclidean distance restriction), where we set DVAEM and s-DVAEM (original DVAEM + supervised learning) as additional variants. For the inference network used in our model, the elements of all weight matrices are initialized with Gaussian distributions whose

TABLE II
Detailed Type and Number Used for Ten-Target MSTAR Dataset

Target	BMP2	BTR70	T72	BTR60	2S1	BRDM2	D7	T62	ZIL131	ZSU234
Train samples	233(C21)	233	232(132)	255	299	298	299	299	299	299
Test samples	587(C21,9566,9563)	196	582(132,S7,812)	195	274	274	274	273	274	274

TABLE III
Variation of the Recognition Rates With Hidden Dimension Size Via PCA, NMF, VAE and Our Single-Layer DVAEM

Methods	**Size**				
	K=100	K=200	K=300	K=400	K=500
PCA	83.31	84.61	85.57	85.78	86.95
NMF	81.83	86.45	84.47	86.48	86.86
VAE	83.47	84.52	85.41	86.76	87.13
Single-layer DAVEM	84.54	86.80	87.47	87.62	**88.28**

TABLE IV
Accuracies of Different Methods on the Three-Target Dataset

Models	**Size**	**Accuracy%**
CNN	32-32 (kernel 3 ×3)	89.06
	32-64-128(kernel 6 ×6)	**96.41**
	64-128-256 (kernel 6 ×6)	95.89
MLP	1000-400	94.94
	500-450	**95.48**
	500-450-50	95.04
SAE	1000-400	88.43
	500-450	**88.71**
	500-450-50	86.32
SAEED (cited from [9])	1000-400	**94.14**
	1000-400-200	83.15
	400-400-200	88.50
DVAEM	500	88.28
	500-450	88.51
	500-450-50	**89.16**
s-DVAEM	500	94.13
	500-450	95.23
	500-450-50	**95.53**
rs-DVAEM	500	95.16
	500-450	96.48
	500-450-50	**96.92**

standard deviations are set to 0.1, and all bias terms are initialized to 0. We set the mini-batch size M as 5. The first 150 epochs are used to train rs-DVAEM without the label information, and then another 50 epochs are used to train rs-DVAEM with the label information. The Adam optimizer [44] with learning rate 10^{-4} is utilized for optimization. For Bayesian generative model, we would like to ensure the corresponding random variables drawn from noninformative priors and all the posteriors learned from data. Thus, we directly set hyper-parameters in DVAEM as $\{\eta^{(l)} = 0.1, c_n^{(l)} = 1, \boldsymbol{r} = \mathbf{1}\}$. When training samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from different targets, we set $S_{ij} = \alpha$ in (15) and $\alpha = 3$. The weight of restriction in (15) is set as $\beta = 1$. The scaling factor in (3) is set as $\mu = 80$.

C. Recognition Results

1) *Three-Target Experiments:* For the three-target MSTAR dataset, we first evaluate the recognition rates of some unsupervised single-layer feature extraction methods followed by linear SVM [45], including PCA, NMF, VAE, and our single-layer DVAEM. The recognition rates of different models varying with the hidden dimension size are listed in Table III. Moving beyond NMF, which extracts the nonnegative hidden units of SAR images, our single-layer DVAEM constructs a probabilistic model to extract the nonnegative features, facilitating more powerful modeling. Compared with VAE with Gaussian distributed latent variables, our proposed model achieves better accuracy, proving the efficiency of gamma distributed nonnegative hidden units for SAR ATR.

To demonstrate the advantages of constructing deep systems and introducing the label information, we evaluate the results of our proposed DVAEM-based models. Here, we set the hidden dimension size of single-layer DVAEM-based models as 500, two-layer DVAEM-based models as $500 - 450$, and three-layer DVAEM-based models as $500 - 450 - 50$. We further compare our models with stacked autoencoder with Euclidean distance restriction (referred to as SAEED) [9]. We report the recognition results of SAEED with two hidden layers (hidden dimension is $1000 - 400$) and three hidden layers ($1000 - 400 - 200$ or $400 - 400 - 200$), respectively, cited from [9]. Besides, some commonly used deep models, including convolutional neural network (CNN) [46], multilayer perceptron (MLP) [47] and stacked autoencoder (SAE), are also compared with our proposed models, where we set the different network structures for them for a fair comparison. Summarized in Table IV are the recognition results of different methods on the three-target MSTAR dataset.

There are several observations drawn from the results. With the development of the layer, we can see a clear recognition performance improvement for all DVAEM-based models. We find the performance of s-DVAEM is higher than that of DVAEM, which confirms the validity of incorporating the label information into the modeling process in s-DVAEM. Besides, rs-DVAEM achieves better performance over s-DVAEM, which indicates the efficiency of enforcing the Euclidean distance restriction on the hidden features. Compared with the SAE, which extracts the deep features of SAR images unsupervisedly, our DVAEM achieves better accuracy. The reason behind this might be that the DVAEM is a deep probabilistic model that learns the distribution inside the SAR image data. We then compare the recognition rates of proposed rs-DVAEM with those of commonly used supervised deep methods, including CNN, MLP, and SAE. Although these deep models perform well in some particular datasets, we find that their recognition results are inferior to our proposed rs-DVAEM. Besides, for MLP and SAE, their two-layer structures can obtain higher accuracies than the three-layer structures; for CNN, the small sizes of the three-layer structures outperform

s-DVAEM	BMP2	BTR70	T72
BMP2	0.928	0.016	0.056
BTR70	0.000	0.995	0.005
T72	0.005	0.012	0.983

(a)

rs-DVAEM	BMP2	BTR70	T72
BMP2	0.960	0.023	0.017
BTR70	0.000	0.995	0.005
T72	0.011	0.020	0.969

(b)

Fig. 3. Shown in (a) and (b) are confusion matrices for three-layer s-DVAEM and rs-DVAEM on three-class data, respectively.

the large sizes. It is not surprising as these deep models ignore learning the kind of variability in highly structured data and only provide a point estimate for the global parameters. Hence, their performance may be significantly affected by the larger structures or sizes due to the limited SAR image training samples. Compared with SAEED adopting Euclidean distance restriction based on SAE to extract hidden units [9], our proposed rs-DVAEM with the two-layer or even single-layer structure still obtains better performance. This observation also proves the effectiveness of our probabilistic framework for SAR ATR. In addition, different from SAEED in [9] employing a downstream classifier, our proposed rs-DVAEM is an end-to-end model, which trains both the feature extractor and classifier jointly. In summary, compared with other deep models for SAR ATR, our proposed rs-DVAEM can extract discriminative features with a small amount of SAR images due to the supervised probabilistic model with the Euclidean distance restriction.

As shown in Fig. 3, we demonstrate the confusion matrices of our proposed three-layer s-DVAEM and rs-DVAEM on different targets. Although each row of the confusion matrix is supposed to be the number of samples in a specific predicted class, for the sake of comparison, we replace it with the predicted ratio. There are several observations drawn from different aspects. As can be seen from the matrix, the proposed method has difficulties in distinguishing between BMP2 and T72, but owns the ability to make the distinction between BTR70 and other types of targets pretty well. The reason is that the appearance of BTR70 is relatively more distinctive than that of other targets. Moreover, the classification accuracy of rs-DVAEM is better than that of s-DVAEM, implying the effectiveness of the learned hierarchical features based on Euclidean distance restriction.

2) *Ten-Target Experiments:* For the ten-target MSTAR dataset, we also compare the rs-DVAEM with its corresponding variants, i.e., DVAEM and r-DVAEM. Here, we set the hidden dimension size of single-layer DVAEM-based models as 1300, two-layer DVAEM-based models as $1300-1100$, and three-layer DVAEM-based models as $1300-1100-600$. Some commonly used deep models, including SAE, MLP and CNN, are also included for comparison, where we also evaluate these models with several different network structures. In addition to the basic CNN, we also compare the results of our proposed methods with CNN based on data augmentation (referred to as CNN+aug) [30], which investigates the capability of a CNN combined with three types of data augmentation operations in SAR ATR. We further compare our models with stacked autoencoder with Euclidean distance (referred to as SAEED) [9]. Here, CNN+aug [30] reports the recognition result of the ten-target MSTAR dataset with the structure of $96-256$ (kernel size is 3×3 in each layer), achieving the performance of 94.29%, and SAEED [9] achieves the recognition accuracy of 91.29% with the structure of $1200-800$ and the accuracy of 91.07% with the structure of $800-600-400$. We list the recognition results of different methods on the ten-target MSTAR dataset in Table V .

TABLE V
Accuracies of Different Methods on the Ten-Target Dataset

Models	Size	Accuracy%
CNN	96-256 (kernel 3 ×3)	89.14
	64-128-256 (kernel 3 ×3)	**94.15**
	128-256-512(kernel 6 ×6)	93.73
CNN+aug (cited from [30])	96-256 (kernel 3 ×3)	94.29
MLP	1200-800	**92.85**
	1300-1100	92.53
	1300-1100-600	92.16
SAE	800-600	**86.76**
	1300-1100	85.98
	1300-1100-600	84.88
SAEED (cited from [9])	1200-800	**91.29**
	800-600-400	91.07
DVAEM	1300	88.44
	1300-1100	89.60
	1300-1100-600	**89.82**
s-DVAEM	1300	94.16
	1300-1100	94.19
	1300-1100-600	**94.87**
rs-DVAEM	1300	94.31
	1300-1100	94.94
	1300-1100-600	**95.47**

We find the classification accuracy of s-DVAEM is higher than the DVAEM because of the direct supervision for the features across all layers, and the accuracy of rs-DVAEM is higher than the s-DVAEM by reason of the Euclidean distance restriction. Similar to the results of three-target data, our proposed unsupervised DVAEM is still superior to the SAE on the ten-target data. Besides, the accuracy of CNN+aug is much higher than CNN, demonstrating the effectiveness of data augmentation in alleviating the overfitting of CNN in SAR ATR. We notice that all the results of these deep models, including

rs-DVAEM	BMP2	BTR70	T72	BTR60	2S1	BRDM2	D7	T62	ZIL131	ZSU234
BMP2	0.939	0.017	0.032	0.005	0.003	0.000	0.000	0.002	0.000	0.002
BTR70	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
T72	0.007	0.016	0.907	0.000	0.017	0.000	0.000	0.051	0.002	0.000
BTR60	0.010	0.000	0.000	0.959	0.000	0.005	0.000	0.005	0.000	0.021
2S1	0.004	0.000	0.000	0.000	0.960	0.011	0.000	0.025	0.000	0.000
BRDM2	0.003	0.000	0.004	0.000	0.040	0.938	0.000	0.011	0.000	0.004
D7	0.000	0.000	0.000	0.000	0.008	0.000	0.985	0.000	0.000	0.007
T62	0.000	0.000	0.004	0.000	0.015	0.000	0.003	0.971	0.007	0.000
ZIL131	0.000	0.000	0.000	0.000	0.004	0.000	0.000	0.000	0.992	0.004
ZSU234	0.000	0.000	0.000	0.000	0.000	0.000	0.018	0.000	0.000	0.982

Fig. 4. Confusion matrix for three-layer rs-DVAEM on ten-class data.

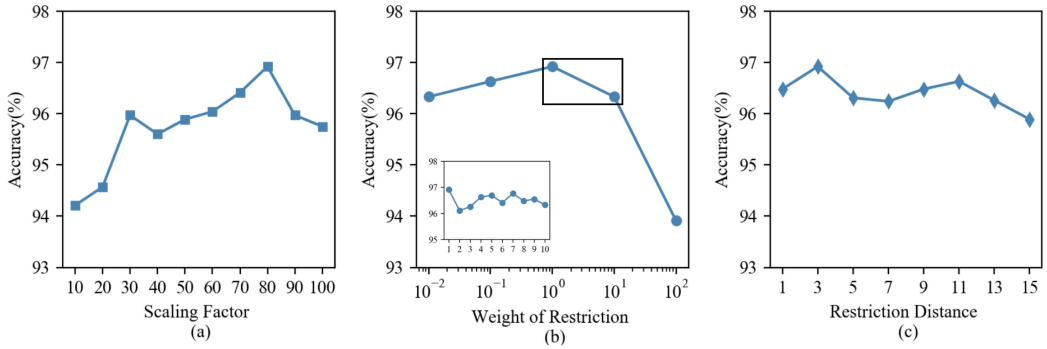


Fig. 5. Shown in (a)–(c) are the recognition results for three-layer rs-DVAEM on three-target data varying with the scaling parameter μ, weight of restriction β, and restriction distance α, respectively.

CNN+aug, CNN, MLP, and SAEED, are still lower than our proposed rs-DVAEM. This phenomenon is similar to the results observed on the three-target data, verifying our model's superiority again. In summary, in comparison with these deep learning methods, rs-DVAEM retains the advantages of the deep probabilistic model and explores the strongly discriminative principle of classifier because of the Euclidean distance restriction and aggregation of multilayer features. Fig. 4 shows the confusion matrix of three-layer rs-DVAEM on ten-class data.

D. Analysis of Our Proposed Model

1) *Analysis of Parameters:* As shown in previous studies [9], it is important to analyze how the recognition result is influenced by the model parameters, which for the proposed models include the scaling parameter μ, the restriction distance α, and the weight of the restriction β.

SAR image data can be modeled with Poisson distribution in (3), by directly quantizing the SAR images dataset to discrete counts before training and testing. Thus, the scaling factor μ controls the fineness of this quantization. In Fig. 5(a), we show the variation of the classification accuracies of rs-DVAEM model with the value of the scaling factor. Clearly, while improving μ usually can increase the performance of the rs-DVAEM, the performance gain would quickly diminish once μ becomes sufficiently large.

In Fig. 5(b), we compare the performance of rs-DVAEM varying with different weights of restriction $\beta \in [10^{-2}, 10^{2}]$, a regularization parameter balancing the Euclidean distance restriction and the log-likelihood of SAR images and their corresponding labels. We can see that if β is too small, the model cannot provide enough distance restriction, leading to poor discriminative features. However, if β gets larger, rs-DVAEM may lead to a worse estimation of data log-likelihood, which is not good for extracting hierarchical features. Nevertheless, $\beta = 1$ is quite a good tradeoff between the two components.

When training samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from different targets, we set $S_{ij} = \alpha$ in (15). Fig. 5(c) shows the variation of the classification results of rs-DVAEM with the value of distance $\alpha \in [1, 15]$. As α decreases, we get the weaker restriction for features from different targets, thus degrading the separability of features. On the contrary, a large α allows more strong restriction on the features to be learned, but a

Fig. 6. Shown in (a)–(c) are the recognition results on the three-target data for single-layer, two-layer, three-layer s-DVAEM, respectively, as a function of the number of hidden units.

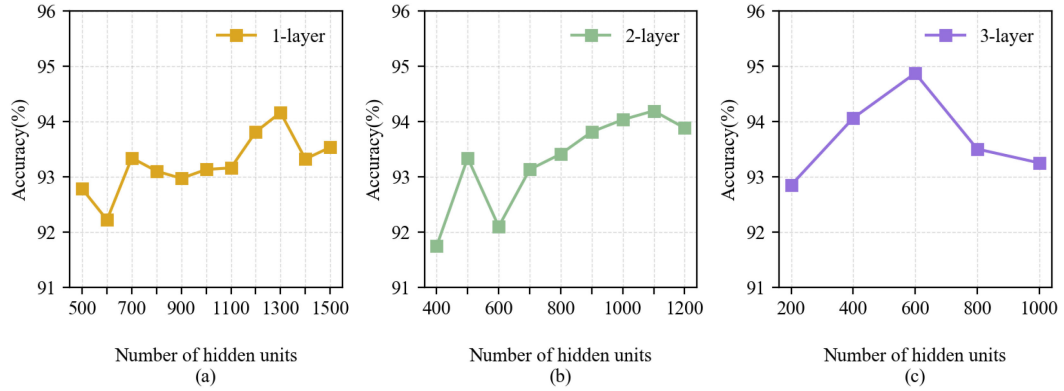


Fig. 7. Shown in (a)–(c) are the recognition results on the ten-target data for single-layer, two-layer, three-layer s-DVAEM, respectively, as a function of the number of hidden units.

too large one results in the overfitting and also decreases the accuracy. Hence, $\alpha = 3$ is a suitable distance restriction.

2) *Analysis of Network Structures:* The structures of deep learning have a significant effect on recognition results. Thus, it is important to analyze how the recognition performance is influenced by the network structures. Here, we present a detailed analysis of the sizes of our proposed method. For brevity, we only analyze the performance of different network structures on s-DVAEM, and set the same structure for rs-DVAEM. For MSTAR three-target data, in Fig. 6(a), we show the variation of the accuracies of one-layer s-DVAEM with the number of hidden units. Since the 500 hidden units can describe the SAR images well, we set the number of hidden units in one-layer s-DVAEM as 500. For two-layer s-DVAEM, we set the hidden dimension as 500 at the first layer and investigate the influence of the hidden dimension in the second layer on the recognition results in Fig. 6(b), where 450 hidden units can obtain leading performance. Fixing the number of hidden units in the first and second layers as 500 and 450, respectively, we then vary the hidden dimension in the third layer to show the corresponding performance of three-layer s-DVAEM in Fig. 6(c). We find the third layer with 50 hidden units is suitable, and a too large number of hidden units will lead to overfitting and degrade the recognition performance of our proposed method. For MSTAR ten-target data, we also show the classification accuracies of s-DVAEM varying with the number of hidden units in Fig. 7(a)–(c). Since the size of the ten-target dataset is larger and the recognition task is more complicated, we use wider structures than that of three-target data to perform SAR ATR. We can find that the $1300 - 1100 - 600$ structure is suitable for describing the data.

3) *Robustness to the Smaller Training Set:* Many algorithms, especially deep learning models, can obtain good results in a particular dataset; however, the performance of the algorithms may be significantly affected if the size of training set is limited [23]. It is worth pointing that a practical SAR ATR system should provide acceptable recognition result even with a few SAR training images [9], [23]. To analyze rs-DVAEM's sensitivity to the number of training samples, we show how the recognition results of rs-DVAEM and other models vary with the number of training samples on the three-target MSTAR dataset. For a fair comparison, we choose the best-performing structures of different models in Table IV, including CNN, SAEED, SAE, and MLP. Here the training datasets in this experiment are randomly sampled from the all three-target training data and the testing data used here remain unchanged. As shown in Fig. 8, when the number of training SAR images decreases, the performance degradation of rs-DVAEM is relatively smooth. Compared with CNN, SAE, SAEED, and MLP,

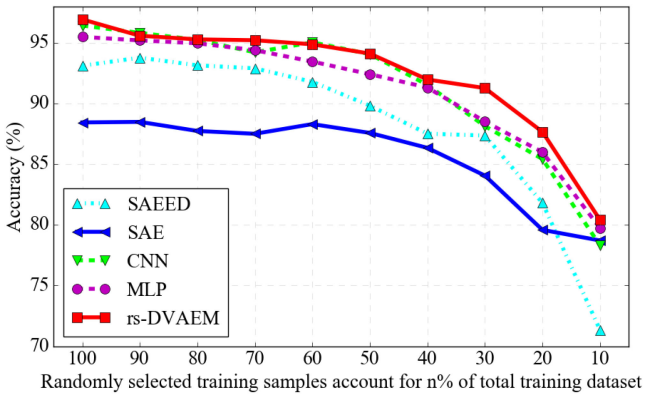


Fig. 8. Comparison of the recognition performance between various methods with different three-target MSTAR training set sizes.

our proposed model can still achieve better recognition result using only dozens of SAR images. The reason behind this phenomenon might be that our proposed rs-DVAEM constructs a probabilistic model to perform distribution estimation and fuses multi-stochastic-layer features under the Euclidean distance restriction to enhance robustness. Besides, the results of SAEED drop more sharply than that of other models with a relatively small training dataset, which may be attributed to the overfitting by restricted supervised learning without applying a probabilistic framework.

4) *Visualization of Features:* In this section, we discuss the effectiveness of our proposed method. We first visualize high-dimensional features learned by three-layer s-DVAEM and rs-DVAEM, by mapping them to the 2-D PCA subspace, where the features across all layers are concatenated as vectors. In Fig. 9, each dot represents a SAR image and each color-shape pair denotes a category. We visually illustrate that the features learned by the proposed rs-DVAEM are more discriminative than those by s-DVAEM, proving the benefits from learning a supervised hierarchical probabilistic model that introduces the Euclidean distance restriction.

5) *Visualization of Dictionary Elements:* To prove that our proposed model considers the human learning system, we visualize the extracted hierarchical representations of SAR images with three-layer rs-DVAEM. For exhibiting the inferred hierarchical structure, it is straightforward to project the dictionary elements $\boldsymbol{\phi}_k^{(l)}$ of layer l to the observed space, which can be depicted in Fig. 10. We find that the dictionary elements at layer one are basic, fundamental, and look similar. And the dictionary elements at layer two are the part-based structures used to constitute the SAR images. Further, the dictionary elements at layer three are more informative than those at layers one and two, so we can find several parts of SAR images, such as target, shadow, and noise. It reveals that the dictionary elements, from the bottom to top layers, gradually become less fundamental and more specialized. Since the learned weight matrices at different layers reveal different abstraction levels, it is reasonable to enforce direct supervision for the features across all layers.

6) *Visualization of the Hierarchical Tree:* Although the specific and general aspects of SAR images have been displayed by exploring the dictionaries of different layers as shown in Fig. 10, it would be more informative to further show the relationships between the dictionary elements of different layers. Therefore, we build a tree to visualize the rs-DVAEM. We first choose a SAR image as the root node, then move down the node to include dictionary components at layer three that are connected to SAR image with sufficiently large weights. By further moving down the network to pick any lower-layer dictionary vectors with the same rule, we can complete the hierarchical tree. As shown in Fig. 11, we find that some of the nodes seem to be closely related to each other, which may be helpful for interpreting and understanding SAR images. For example, the three main vectors of the dictionary $\boldsymbol{\Phi}^{(3)}$ contain several parts of the SAR image, which are quite different. While the vectors of dictionary $\boldsymbol{\Phi}^{(2)}$ also have several parts of the upper dictionary $\boldsymbol{\Phi}^{(3)}$, the differences between the six nodes of layer two decrease. At the bottom of this tree, the main vectors of dictionary $\boldsymbol{\Phi}^{(1)}$ used to constitute $\boldsymbol{\Phi}^{(2)}$ are very basic and belong to the pixel-level information. Therefore, we can get the same conclusion that the dictionary elements learned by rs-DVAEM become more and more fundamental when moving down the tree from top to bottom. The usage of dictionary elements in each layer also proves the point discussed here, shown in Fig 12.

7) *Sparseness Analysis:* The sparse constraint, reducing the redundant information and improving the accuracy, has been playing an important role, particularly in the area of part-based dictionary learning [16], [17], [24]. We present the usage frequency of the dictionary elements in Fig. 12, where we plot the top 50 dictionary elements at each layer. We find that the dictionary elements are becoming more specialized hence more sparsely utilized with the increasing layer. Without enforcing sparse constraint, rs-DVAEM still owns favorable and natural sparseness for the product of Dirichlet distributed dictionary and gamma distributed nonnegative hidden units in each layer.

V. CONCLUSION

A novel SAR image ATR method based on the supervised deep generative model under the Euclidean distance restriction, rs-DVAEM, is proposed. The proposed method has the ability to learn highly discriminative hierarchical structured representations of SAR images. The visualization of the dictionary elements in each layer shows that the proposed model can extract the hierarchical and interpretable information of the targets and enhance the comprehension of the SAR images. Experimental results on MSTAR data demonstrate that the proposed method has a better performance than other related methods.

APPENDIX A
DETAILS OF TLASGR-MCMC

Here, we will provide more details about $\mathbf{X}^{(l)}$, $M_k^{(l)}$, and $[.]_{\angle}$. For the sake of completeness, we rewrite the update

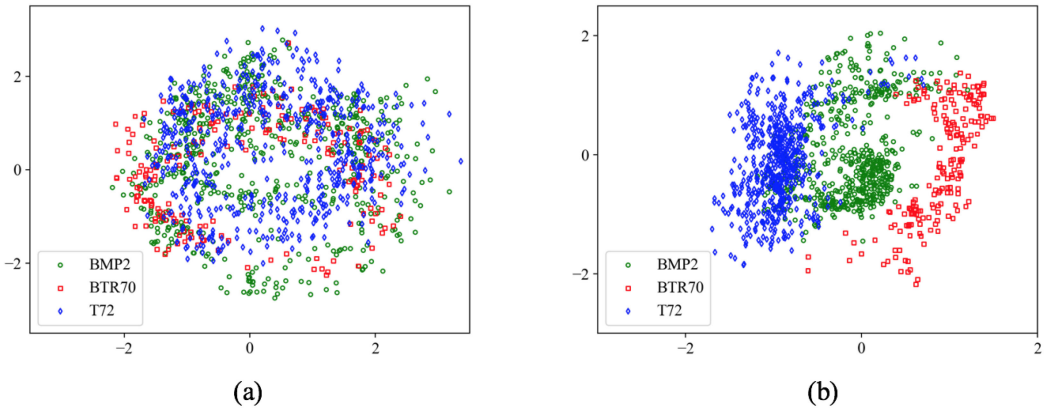


Fig. 9. Two-dimensional PCA projection of features extracted from the test SAR images. Shown in (a) and (b) are learned features by s-DVAEM and rs-DVAEM, respectively.

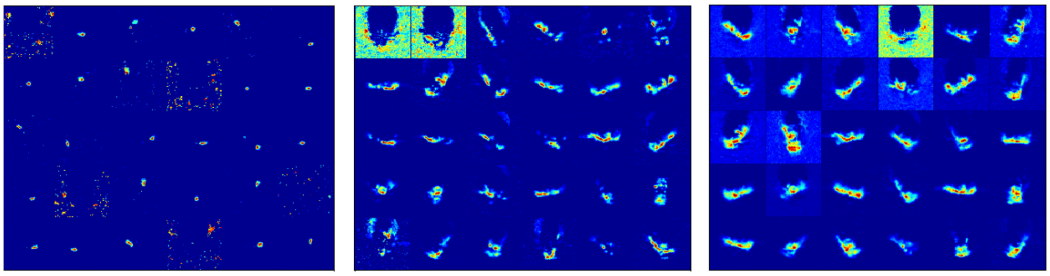


Fig. 10. Dictionary for the MSTAR three-target data based on a three-layer rs-DVAEM. Shown in (a)–(c) are dictionary elements for layers one, two, and three, respectively.

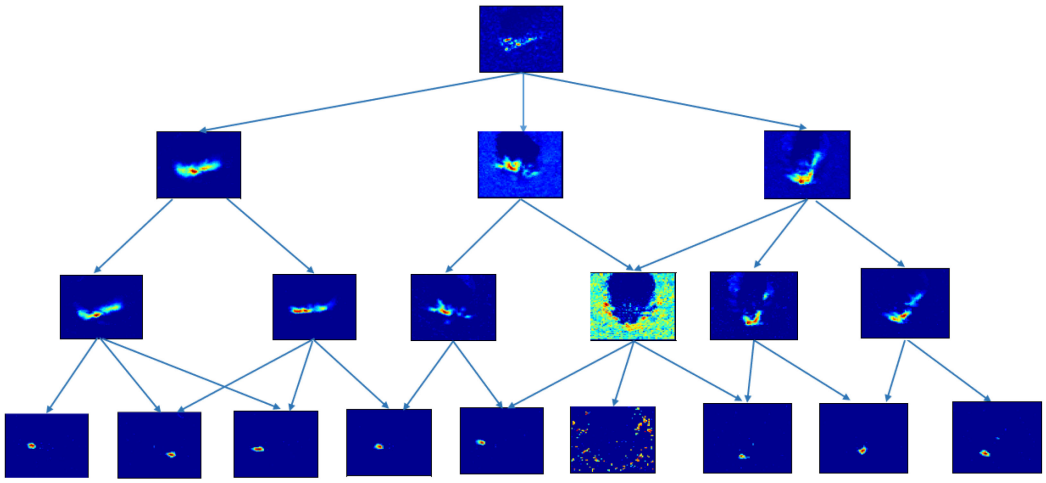


Fig. 11. Correlation between SAR image and dictionary elements across all layers.

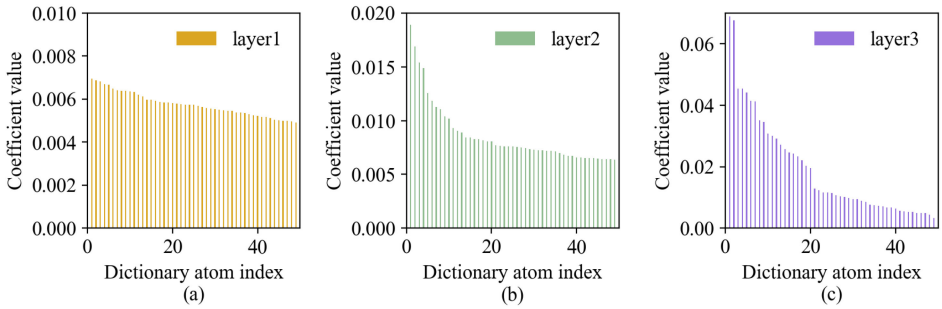


Fig. 12. Statistics of three-layer rs-DVAEM coefficients on MSTAR three-target data. Shown in (a)–(c) are usage of dictionary elements for layers one, two, and three, respectively.

equation for $\boldsymbol{\phi}_k^{(l)}$ as

$$\left(\boldsymbol{\phi}_k^{(l)}\right)_{i+1} = \left[\left(\boldsymbol{\phi}_k^{(l)}\right)_i + \frac{\varepsilon_i}{M_k^{(l)}}\left[\left(\rho\tilde{\boldsymbol{x}}_{:k\cdot}^{(l)} + \eta^{(l)}\right) - \left(\rho\tilde{x}_{\cdot k\cdot}^{(l)} + K_{l-1}\eta^{(l)}\right)\left(\boldsymbol{\phi}_k^{(l)}\right)_i\right] + \mathcal{N}\left(0, \frac{2\varepsilon_i}{M_k^{(l)}}\mathrm{diag}(\boldsymbol{\phi}_k^{(l)})_i\right)\right]_{\angle} \quad (16)$$

where both $\tilde{\boldsymbol{x}}_{:k\cdot}^{(l)}$ and $\tilde{x}_{\cdot k\cdot}^{(l)}$ come from the augmented latent counts $\mathbf{X}^{(l)}$, $M_k^{(l)}$ is calculated using the estimated FIM, and $[.]_{\angle}$ denotes the simplex constraint.

Sample the augmented auxiliary counts $\mathbf{X}^{(l)}$: Given mini-batch $\{\boldsymbol{x}_m\}_{m=1}^M$ in the training set and their corresponding latent features $\{\boldsymbol{\theta}_m^{(1:L)}\}_{m=1}^M$, we aim to update the global parameters $\boldsymbol{\Phi}^{(1:L)}$ with TLASGR-MCMC. To model the SAR image with Poisson likelihood, $\boldsymbol{x}_m$ can be discretized into the count vector $\lfloor \mu \boldsymbol{x}_m \rfloor$ with the deterministic scaling factor μ before training. Below $\boldsymbol{x}_m$ will be used to represent the discretized count vector for brevity. Let us denote $\mathbf{X}^{(l)} \in \mathbb{Z}^{K_{l-1}\times K_l \times M}$ as the latent counts, $X_{\cdot km}^{(l)} = \sum_{k=1}^{K_{l-1}} X_{kk'm}^{l}$, $x_{km}^{(l)}$ with $k \in K_{l-1}$ and $x_{km}^{(1)} = x_{vm}$. Working upward for $l = 1, \ldots, L$, we draw

$$\left(X_{kk'm}^{(l)}\right)_{k'=1,K_l} \sim \mathrm{Multi}\left(x_{km}^{(l)}; \frac{(\phi_{kk'}^{(l)}\theta_{k'm}^{(l)})_{k'=1,K_l}}{\sum_{k_l=1}^{K_l}\phi_{vk_l}^{(l)}\theta_{k_lm}^{(l)}}\right) \quad (17)$$

$$x_{km}^{(l+1)} \sim \mathrm{CRT}\left(X_{\cdot km}^{(l)}, \sum_{k'=1}^{K_{l+1}}\phi_{kk'}^{(l+1)}\theta_{k'm}^{(l+1)}\right). \quad (18)$$

Given the augmented latent count variables $\mathbf{X}^{(l)}$, we can compute $\tilde{\boldsymbol{x}}_{:k\cdot}^{(l)}$ and $\tilde{x}_{\cdot k\cdot}^{(l)}$ used to sample the $\boldsymbol{\phi}_k^{(l)}$ in (16), expressed as

$$\tilde{x}_{k'k\cdot}^{(l)} = \sum_{m=1}^{M} X_{k'km}^{(l)}, \tilde{\boldsymbol{x}}_{:k\cdot}^{l} = \{\tilde{x}_{1k\cdot}^{l}, \ldots, \tilde{x}_{K_{l-1}k\cdot}^{l}\}^T \quad (19)$$

$$\tilde{x}_{\cdot k\cdot}^{(l)} = \sum_{k'=1}^{K_{l-1}}\tilde{\boldsymbol{x}}_{k'k\cdot}^{(l)}. \quad (20)$$

Compute $\boldsymbol{M_k^{(l)}}$: Note that $M_k^{(l)}$ for $l \in \{1, \ldots, L\}$, appearing as the denominator in (16), can be calculated using the estimated FIM. Following (18), (19) of [38], we can update $M_k^{(l)}$ as

$$M_k^{(l)} = (1-\varepsilon_i)M_k^{(l)} + \varepsilon_i \rho E\left[\tilde{x}_{\cdot k\cdot}^{(l)}\right] \quad (21)$$

where $E[\cdot]$ denotes averaging over the collected MCMC samples. Note that as in (16), instead of having a single learning rate for all layers and weight vectors, a common practice due to the difficulty to adapt the step sizes to different layers and/or weight vectors, the TLASGR MCMC employs topic-layer-adaptive learning rates as $\varepsilon_i/M_k^{(l)}$, adapting a single step size ε_t to different weight vectors and layers by multiplying it with the weights $1/M_k^{(l)}$ for $l \in \{1, \ldots, L\}$ and $k \in \{1, \ldots, K_l\}$.

Implement the simplex constraint: The simplex constraint $[.]_{\angle}$ means $\phi_{k'k}^{(l)} \geqslant 0$ and $\sum_{k'=1}^{K_{l-1}}\phi_{k'k}^{(l)} = 1$. The first line in (16) represents the mean of $\boldsymbol{\phi}_k^{(l)}$, denoted as $\boldsymbol{\mu}(\boldsymbol{\phi}_k^{(l)})$. Following [39, Examples 1–3], the simulation from (16) can be approximately but rapidly realized as

1) sample $\boldsymbol{y} \sim \mathcal{N}[\boldsymbol{\mu}(\boldsymbol{\phi}_k^{(l)}), \frac{2\varepsilon_i}{M_k^{(l)}}\mathrm{diag}(\boldsymbol{\phi}_k^{(l)})_i]$
2) calculate $\boldsymbol{z} = \boldsymbol{y} + (1 - \mathbf{1}^T\boldsymbol{y})\boldsymbol{\phi}_t$
3) if $\boldsymbol{z}$ satisfies the simplex constraint, return $(\boldsymbol{\phi}_k^{(l)})_{i+1} = (z_1, \ldots, z_{K_{l-1}})^T$; else calculate $\boldsymbol{d} = \max(\gamma, \boldsymbol{z})$ with a small constant $\gamma \geq 0$, let $\boldsymbol{e} = \boldsymbol{d}/\sum_{i=1}^{K_{l-1}} d_i$, and return $(\boldsymbol{\phi}_k^{(l)})_{i+1} = (e_1, \ldots, e_{K_{l-1}})^T$.

We refer the readers to Cong *et al.* [38] for more details on the derivation for the TLASGR MCMC algorithm.

APPENDIX B
GRADIENTS OF THE NEURAL NETWORK PARAMETERS ABOUT THE ELBO IN DVAEM

Here, we aim to update the neural network parameters $\boldsymbol{\Omega} = \{\mathbf{W}_{xh}, \mathbf{W}_{hk}^{(l)}, \mathbf{W}_{h\lambda}^{(l)}, \boldsymbol{b}_1^{(l)}, \boldsymbol{b}_2^{(l)}, \boldsymbol{b}_3^{(l)}\}$ from the inference model. For convenience, we rewrite the evidence lower bound (ELBO) of the log marginal likelihood in (II-B), formulated as

$$\begin{aligned} L &= \sum_{n=1}^{N} E_{q(\boldsymbol{\theta}_n^{(1)})}\left[\ln p\left(\boldsymbol{x}_n|\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right)\right] \\ &\quad - \sum_{n=1}^{N}\sum_{l=1}^{L} E_{q(\boldsymbol{\theta}_n^{(l)})}\left[\ln\frac{q\left(\boldsymbol{\theta}_n^{(l)}\right)}{p\left(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, 1/c_n^{(l+1)}\right)}\right] \\ &= \sum_{n=1}^{N} L1\left(q\left(\boldsymbol{\theta}_n^{(1)}\right)\right) + \sum_{n=1}^{N}\sum_{l=1}^{L} L2\left(q\left(\boldsymbol{\theta}_n^{(l)}\right)\right) \end{aligned} \quad (22)$$

where the first term encourages a good fit to the observations, and the second term aims to fit the latent representations at each layer. The second term $L2(q(\boldsymbol{\theta}_n^{(l)}))$ in ELBO denotes the Kullback–Leibler (KL) divergence between $q(\boldsymbol{\theta}_n^{(l)})$ and $p(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, 1/c_n^{(l+1)})$, represented as $\mathrm{KL}(q(\boldsymbol{\theta}_n^{(l)})\|p(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{\alpha}_n^{(l+1)}, 1/c_n^{(l+1)}))$, where $\boldsymbol{\alpha}_n^{(l+1)} = \boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}$ denotes the shape parameter of $\boldsymbol{\theta}_n^{(l)}$ for brevity. Since we choose the Weibull distribution to construct the inference network, i.e., $q(\boldsymbol{\theta}_n^{(l)})$ the KL divergence between Weibull distribution and gamma distribution has a closed-form expression (for ELBO computation), whose general derivation can be expressed as

$$\begin{aligned} &\mathrm{KL}(\text{Weibull}\ (k, \lambda)\|\ \text{Gamma}(\alpha, 1/\beta)) \\ &= -\alpha\ln\lambda + \frac{\gamma\alpha}{k} + \ln k + \beta\lambda\Gamma\left(1 + \frac{1}{k}\right) \\ &\quad - \gamma - 1 - \alpha\ln\beta + \ln\Gamma(\alpha) \end{aligned} \quad (23)$$

where Γ is the gamma function, γ denotes the Euler's constant, and we set $\gamma = 0.57721$. By replacing the parameters $\{k, \lambda, \alpha, 1/\beta\}$ with the specific parameters from Weibull distribution $q(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{k}_n^{(l)} + \boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, \boldsymbol{\lambda}_n^{(l)})$ in (7), denoted as $q(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{k}_n^{(l)} + \boldsymbol{\alpha}_n^{(l+1)}, \boldsymbol{\lambda}_n^{(l)})$, and the parameters from gamma

distribution $p(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{\alpha}, 1/c_n^{(l+1)})$, we can rewrite the closed-form expression of the KL divergence as

$$\begin{aligned} L2\left(q\left(\boldsymbol{\theta}_n^{(l)}\right)\right) &= \mathrm{KL}\left(q\left(\boldsymbol{\theta}_n^{(l)}\right) \| p\left(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{\alpha}_n^{(l+1)}, 1/c_n^{(l+1)}\right)\right) \\ &= -\boldsymbol{\alpha}_n^{(l+1)} \ln \boldsymbol{\lambda}_n^{(l)} + \frac{\gamma \boldsymbol{\alpha}_n^{(l+1)}}{\boldsymbol{k}_n^{(l)} + \boldsymbol{\alpha}_n^{(l+1)}} + \ln\left(\boldsymbol{k}_n^{(l)} + \boldsymbol{\alpha}_n^{(l+1)}\right) \\ &\quad + \frac{1}{c_n^{(l+1)}} \boldsymbol{\lambda}_n^{(l)} \Gamma\left(1 + \frac{1}{\boldsymbol{k}_n^{(l)} + \boldsymbol{\alpha}_n^{(l+1)}}\right) \\ &\quad - \gamma - 1 - \boldsymbol{\alpha}_n^{(l+1)} \ln \frac{1}{c_n^{(l+1)}} + \ln \Gamma(\boldsymbol{\alpha}_n^{(l+1)}) \end{aligned} \tag{24}$$

where the analytic KL divergence results in small variance and hence stable evaluation of the gradient. Since $\boldsymbol{x}_n$ follows the Poisson distribution with parameter $\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}$, we can rewrite the first term in ELBO as

$$\begin{aligned} L1\left(q\left(\boldsymbol{\theta}_n^{(1)}\right)\right) &= E_{q\left(\boldsymbol{\theta}_n^{(1)}\right)}\left[\ln p\left(\boldsymbol{x}_n|\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right)\right] \\ &= E_{q\left(\boldsymbol{\theta}_n^{(1)}\right)}\left[\boldsymbol{x}_n \ln\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right) - \boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)} - \ln(\boldsymbol{x}_n!)\right] \end{aligned} \tag{25}$$

where the deterministic constant $\ln(\boldsymbol{x}_n!)$ can be ignored when computing the gradient of parameters $\boldsymbol{\Omega}$. Due to simple reparameterization of the Weibull distribution, the gradient of the first term of the ELBO with respect to the parameters $\boldsymbol{\Omega}$ can have a low variance, by reparametrizing $\boldsymbol{\theta}_n^{(1)}(\boldsymbol{\epsilon}_n^{(1)}) = \boldsymbol{\lambda}_n^{(1)}(-\ln(1-\boldsymbol{\epsilon}_n^{(1)}))^{1/\boldsymbol{k}_n^{(1)}}$ and rewriting

$$\begin{aligned} L1\left(q\left(\boldsymbol{\theta}_n^{(1)}\right)\right) &= E_{q\left(\boldsymbol{\theta}_n^{(1)}\right)}\left[\boldsymbol{x}_n \ln\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right) - \boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right] \\ &= E_{q\left(\boldsymbol{\epsilon}_n^{(1)}\right)}\left[\boldsymbol{x}_n \ln\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\left(\boldsymbol{\epsilon}_n^{(1)}\right)\right) - \boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\left(\boldsymbol{\epsilon}_n^{(1)}\right)\right] \end{aligned} \tag{26}$$

where $\boldsymbol{\epsilon}_n^{(1)}$ is a vector of standard uniform variable and usually a single Monte Carlo sample can be used to estimate the gradient and achieve satisfactory performance. Therefore, the ELBO can be represented as the sum of $\sum_{n=1}^{N} L1(q(\boldsymbol{\theta}_n^{(1)}))$ from (26) and $\sum_{n=1}^{N}\sum_{l=1}^{L} L2(q(\boldsymbol{\theta}_n^{(l)}))$ from (24). By maximizing the ELBO, the inference model parameters $\boldsymbol{\Omega}$ can be trained through standard backpropagation technique for stochastic gradient descent. Taking the parameter $\mathbf{W}_{h\lambda}^{(1)}$ in $\boldsymbol{\Omega}$ as the example, we can compute its gradient w.r.t. to the ELBO as

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{W}_{h\lambda}^{(1)}} &= \sum_{n=1}^{N}\left(\frac{\partial L1\left(q\left(\boldsymbol{\theta}_n^{(1)}\right)\right)}{\partial \boldsymbol{\lambda}_n^{(1)}} + \frac{\partial L2\left(q\left(\boldsymbol{\theta}_n^{(1)}\right)\right)}{\partial \boldsymbol{\lambda}_n^{(1)}}\right)\frac{\partial \boldsymbol{\lambda}_n^{(1)}}{\partial \mathbf{W}_{h\lambda}^{(1)}} \\ &= \sum_{n=1}^{N}\left(\frac{\partial L1\left(q\left(\boldsymbol{\theta}_n^{(1)}\right)\right)}{\partial \boldsymbol{\theta}_n^{(1)}\left(\boldsymbol{\epsilon}_n^{(1)}\right)}\frac{\partial \boldsymbol{\theta}_n^{(1)}\left(\boldsymbol{\epsilon}_n^{(1)}\right)}{\partial \boldsymbol{\lambda}_n^{(1)}} + \frac{\partial L2\left(q\left(\boldsymbol{\theta}_n^{(1)}\right)\right)}{\partial \boldsymbol{\lambda}_n^{(1)}}\right) \\ &\quad \times \frac{\partial \boldsymbol{\lambda}_n^{(1)}}{\partial \mathbf{W}_{h\lambda}^{(1)}} \end{aligned} \tag{27}$$

where the gradient $\frac{\partial \boldsymbol{\lambda}_n^{(1)}}{\partial \mathbf{W}_{h\lambda}^{(1)}}$ can be computed by following the mapping function in (9). The gradients of neural network parameters $\boldsymbol{\Omega}$ from the inference model in s-DVAEM or rs-DVAEM can also be inferred in a similar manner by maximizing the ELBO in (14) or (15).

APPENDIX C
THE GRADIENT OF THE CLASSIFIER WEIGHT MATRIX IN S-DVAEM OR RS-DVAEM

Here, we discuss how to update the classifier weight matrix $\mathbf{W}_{sy}$ from our supervised models, including s-DVAEM and rs-DVAEM. Due to simple reparameterization of the Weibull distribution, we rewrite the log-likelihood of the label information in (14) as

$$\begin{aligned} L_y &= \sum_{n=1}^{N}\sum_{l=1}^{L} E_{q\left(\boldsymbol{\theta}_n^{(l)}\right)}\left[\ln p(y_n|\mathbf{W}_{sy}, \mathbf{S}_n)\right] \\ &= \sum_{n=1}^{N}\sum_{l=1}^{L} E_{q\left(\boldsymbol{\epsilon}_n^{(l)}\right)}\left[\ln p(y_n|\mathbf{W}_{sy}, \mathbf{S}_n)\right] \\ &\approx \sum_{n=1}^{N}\left[\ln p\left(y_n|\mathbf{W}_{sy}, \mathbf{S}_n\left(\boldsymbol{\epsilon}_n^{(1:L)}\right)\right)\right] \\ &= \sum_{n=1}^{N}\sum_{c=1}^{C} \mathrm{I}\{y_n = c\} \ln p_{cn} \end{aligned} \tag{28}$$

where $\mathbf{S}_n(\boldsymbol{\epsilon}_n^{(1:L)})$ denotes the concatenation of the features, i.e., $\mathbf{S}_n = [\boldsymbol{\theta}_n^{(1)}(\boldsymbol{\epsilon}_n^{(1)}), \ldots, \boldsymbol{\theta}_n^{(L)}(\boldsymbol{\epsilon}_n^{(L)})] \in \mathbb{R}^{\sum_{l=1}^{L} K_l}$. Since the probability p_{cn} of the SAR image $\boldsymbol{x}_n$ classified to label c is computed with the softmax function in (13), we can further formulate L_y as

$$L_y = \sum_{n=1}^{N}\sum_{c=1}^{C} \mathrm{I}\{y_n = c\} \ln \frac{e^{\boldsymbol{w}_c^T \mathbf{S}_n}}{\sum_i^C e^{\boldsymbol{w}_i^T \mathbf{S}_n}}. \tag{29}$$

Thus, the gradient of $\boldsymbol{w}_c$ in $\mathbf{W}_{sy} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_c, \ldots, \boldsymbol{w}_C]$ can be computed as

$$\begin{aligned} \nabla_{\boldsymbol{w}_c} L_y &= \sum_{n=1}^{N}\left(\mathrm{I}\{y_n = c\} \times \left(1 - \frac{e^{\boldsymbol{w}_c^T \mathbf{S}_n}}{\sum_{i=1}^{C} e^{\boldsymbol{w}_i^T \mathbf{S}_n}}\right)\right. \\ &\quad \left. - \left(\frac{e^{\boldsymbol{w}_c^T \mathbf{S}_n}}{\sum_{i=1}^{C} e^{\boldsymbol{w}_i^T \mathbf{S}_n}}\right)\left(1 - \mathrm{I}\{y_n = c\}\right)\right)\mathbf{S}_n \\ &= \sum_{n=1}^{N}\left(\mathrm{I}\{y_n = c\} - \left(\frac{e^{\boldsymbol{w}_c^T \mathbf{S}_n}}{\sum_{i=1}^{C} e^{\boldsymbol{w}_i^T x^{(n)}}}\right)\right)\mathbf{S}_n \\ &= \sum_{n=1}^{N}\left[\mathrm{I}\{y_n = c\} - p\left(y_n = c \mid \mathbf{S}_n; \mathbf{W}_{sy}\right)\right]\mathbf{S}_n. \end{aligned} \tag{30}$$

Then, $\boldsymbol{w}_c$ can be updated via SGD using the Adam optimizer.

REFERENCES

[1] L. M. Novak, G. J. Owirka, and A. L. Weaver
Automatic target recognition using enhanced resolution SAR data
IEEE Trans. Aerosp. Electron. Syst., vol. 35, no. 1, pp. 157–175, Jan. 1999.

[2] F. Sadjadi
Improved target classification using optimum polarimetric SAR signatures
IEEE Trans. Aerosp. Electron. Syst., vol. 38, no. 1, pp. 38–49, Jan. 2002.

[3] H. Zhang, N. M. Nasrabadi, Y. Zhang, and T. S. Huang
Multi-view automatic target recognition using joint sparse representation
IEEE Trans. Aerosp. Electron. Syst., vol. 48, no. 3, pp. 2481–2497, Jul. 2012.

[4] U. Srinivas, V. Monga, and R. G. Raj
SAR automatic target recognition using discriminative graphical models
IEEE Trans. Aerosp. Electron. Syst., vol. 50, no. 1, pp. 591–606, Jan. 2014.

[5] T. Li and L. Du
Target discrimination for SAR ATR based on scattering center feature and K-center one-class classification
IEEE Sensors J., vol. 18, no. 6, pp. 2453–2461, Mar. 2018.

[6] E. E. Laubie, B. D. Rigling, and R. P. Penno
Decreased probability of error in template-matching classification using aspect-diverse bistatic SAR
IEEE Trans. Aerosp. Electron. Syst., vol. 54, no. 4, pp. 1862–1870, Aug. 2018.

[7] Y. Sun, L. Du, Y. Wang, Y. Wang, and J. Hu
SAR automatic target recognition based on dictionary learning and joint dynamic sparse representation
IEEE Geosci. Remote Sens. Lett., vol. 13, no. 12, pp. 1777–1781, Dec. 2016.

[8] Y. Wang and H. Liu
SAR target discrimination based on BOW model with sample-reweighted category-specific and shared dictionary learning
IEEE Geosci. Remote Sens. Lett., vol. 14, no. 11, pp. 2097–2101, Nov. 2017.

[9] S. Deng, L. Du, C. Li, J. Ding, and H. Liu
SAR automatic target recognition based on euclidean distance restricted autoencoder
IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 10, no. 7, pp. 3323–3333, Jul. 2017.

[10] O. Kechagias-Stamatis and N. Aouf
Fusing deep learning and sparse coding for SAR ATR
IEEE Trans. Aerosp. Electron. Syst., vol. 55, no. 2, pp. 785–797, Apr. 2019.

[11] K. E. Dungan
Feature-based vehicle classification in wide-angle synthetic aperture radar
Dissertations Thesis, The Ohio State University, 2010.

[12] K. Ikeuchi, T. Shakunaga, M. D. Wheeler, and T. Yamazaki
Invariant histograms and deformable template matching for SAR target recognition
In *Proc. CVPR IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 1996, pp. 100–105.

[13] T. D. Ross, S. W. Worrell, V. J. Velten, J. C. Mossing, and M. L. Bryant
Standard SAR ATR evaluation experiments using the MSTAR public release data set in Algorithms for Synthetic Aperture Radar Imagery V
Int. Soc. Opt. Photon., vol. 3370, pp. 566–573, 1998.

[14] L. M. Novak, G. J. Owirka, and W. S. Brower
Performance of 10- and 20-target MSE classifiers
IEEE Trans. Aerosp. Electron. Syst., vol. 36, no. 4, pp. 1279–1289, Oct. 2000.

[15] Y. Wang, P. Han, X. Lu, R. Wu, and J. Huang
The performance comparison of adaboost and SVM applied to SAR ATR
In *Proc. CIE Int. Conf. Radar*, 2006, pp. 1–4.

[16] D. D. Lee and H. S. Seung
Learning the parts of objects by non-negative matrix factorization
Nature, vol. 401, no. 6755, pp. 788–791, 1999.

[17] P. O. Hoyer
Non-negative matrix factorization with sparseness constraints
J. Mach. Learn. Res., vol. 5, pp. 1457–1469, 2004.

[18] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma
Robust face recognition via sparse representation
IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 2, pp. 210–227, Feb. 2009.

[19] G. Dong, G. Kuang, N. Wang, L. Zhao, and J. Lu
SAR target recognition via joint sparse representation of monogenic signal
IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 8, no. 7, pp. 3316–3328, Jul. 2015.

[20] Z. Jianxiong, S. Zhiguang, C. Xiao, and F. Qiang
Automatic target recognition of SAR images based on global scattering center model
IEEE Trans. Geosci. Remote Sens., vol. 49, no. 10, pp. 3713–3729, Oct. 2011.

[21] C. Nilubol, Q. H. Pham, R. M. Mersereau, M. J. Smith, and M. A. Clements
Hidden markov modelling for SAR automatic target recognition
In *Proc. IEEE Int. Conf. Acoust., Speech., Signal Process., (Cat. No 98CH36181)*, vol. 2, 1998, pp. 1061–1064.

[22] J. A. O'Sullivan, M. D. DeVore, V. Kedia, and M. I. Miller
SAR ATR performance using a conditionally gaussian model
IEEE Trans. Aerosp. Electron. Syst., vol. 37, no. 1, pp. 91–108, Jan. 2001.

[23] R. Shang, J. Wang, L. Jiao, R. Stolkin, B. Hou, and Y. Li
SAR targets classification based on deep memory convolution neural networks and transfer parameters
IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 11, no. 8, pp. 2834–2846, Aug. 2018.

[24] B. Chen, G. Polatkan, G. Sapiro, D. M. Blei, D. B. Dunson, and L. Carin
Deep learning with hierarchical convolutional factor analysis
IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1887–1901, Aug. 2013.

[25] G. E. Hinton and R. Salakhutdinov
Reducing the dimensionality of data with neural networks
Science, vol. 313, no. 5786, pp. 504–507, 2006.

[26] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol
Extracting and composing robust features with denoising autoencoders
In *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[27] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng
Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations
In *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 609–616.

[28] S. Chen and H. Wang
SAR target recognition based on deep learning
In *Proc. Int. Conf. Data Sci. Adv. Analytics*, 2014, pp. 541–547.

[29] J. C. Ni and Y. L. Xu
SAR automatic target recognition based on a visual cortical system
In *Proc. 6th Int. Congr. Image Signal Process.*, vol. 2, 2013, pp. 778–782.

[30] J. Ding, B. Chen, H. Liu, and M. Huang
Convolutional neural network with data augmentation for SAR target recognition
IEEE Geosci. Remote Sens. Lett., vol. 13, no. 3, pp. 364–368, Mar. 2016.

[31] X. Ma, H. Wang, and J. Geng
Spectral-spatial classification of hyperspectral image based on deep auto-encoder
IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 9, pp. 4073–4085, Sep. 2016.

[32] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin
Target classification using the deep convolutional networks for SAR images
IEEE Trans. Geosci. Remote Sens., vol. 54, no. 8, pp. 4806–4817, Aug. 2016.

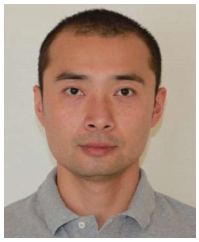
[33] D. P. Kingma and M. Welling
Auto-encoding variational Bayes
In *Proc. Int. Conf. Learn. Representations*, 2014.

[34] M. Zhou, Y. Cong, and B. Chen
The Poisson gamma belief network

In *Proc. Neural Inf. Process. Syst.*, 2015, pp. 3043–3051.
[35] H. Zhang, B. Chen, D. Guo, and M. Zhou, "WHAI
: Weibull hybrid autoencoding inference for deep topic modeling
In *Proc. Int. Conf. Learn. Representations*, 2018.
[36] C. Li, J. Zhu, T. Shi, and B. Zhang
Max-margin deep generative models
In *Proc. Neural Inf. Process. Syst.*, 2015, pp. 1837–1845.
[37] M. Zhou, Y. Cong, and B. Chen
Augmentable gamma belief networks
J. Mach. Learn. Res., vol. 17, no. 163, pp. 1–44, 2016.
[38] Y. Cong, B. Chen, H. Liu, and M. Zhou
Deep latent Dirichlet allocation with topic-layer-adaptive stochastic gradient Riemannian MCMC
In *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 864–873.
[39] Y. Cong, B. Chen, and M. Zhou
Fast simulation of hyperplane-truncated multivariate normal distributions
Bayesian Anal., vol. 12, no. 4, pp. 1017–1037, 2017.
[40] C. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu
Deeply-supervised nets
In *Proc. Int. Conf. Artif. Intell. Statist.*, 2015, pp. 562–570.
[41] T. D. Ross, S. W. Worrell, V. J. Velten, J. C. Mossing, and M. L. Bryant
Standard SAR ATR evaluation experiments using the MSTAR public release data set
Proc. SPIE, vol. 3370, 1998, pp. 566–573.
[42] Q. Zhao and J. C. Principe
Support vector machines for SAR automatic target recognition
IEEE Trans. Aerosp. Electron. Syst.., vol. 37, no. 2, pp. 643–654, Apr. 2001.
[43] Z. Wang, Y. Wang, H. Liu, and H. Zhang
Structured kernel dictionary learning with correlation constraint for object recognition
IEEE Trans. Image Process., vol. 26, no. 9, pp. 4578–4590, Sep. 2017.
[44] D. P. Kingma and J. Ba
Adam: A method for stochastic optimization
In *Proc. Int. Conf. Learn. Representations*, 2015.
[45] C. C. Chang and C. J. Lin
LIBSVM: A library for support vector machines
ACM Trans. Intell. Syst. Technol., vol. 2, pp. 27: 1–27:27, 2011, software. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm
[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton
Imagenet classification with deep convolutional neural networks
In *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[47] S. Agatonovic-Kustrin and R. Beresford
Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research
J. Pharmaceut. Biomed. Anal., vol. 22, no. 5, pp. 717–727, 2000.

Dandan Guo received the B.S. degree in optical information science and technology from North University of China, Tai Yuai, China, in 2014. She received the Ph.D. degree in signal processing from Xidian University, Xi'an, China, in 2020.

She is currently working as a Postdoctoral Researcher with Chinese University of Hong Kong (Shenzhen), Shenzhen, China. Her current research interests include radar automatic target recognition, statistical machine learning, including hierarchical models and statistical inference.



Bo Chen (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 2003, 2006, and 2008, respectively, all in electronic engineering.

He became a Postdoctoral Fellow, a Research Scientist, and a Senior Research Scientist with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA, from 2008 to 2012. From 2013, he has been a Professor with the National Laboratory for Radar Signal Processing, Xidian University. He received the Honorable Mention for 2010 National Excellent Doctoral Dissertation Award and is selected into Oversea Talent by Chinese Central Government in 2014. His current research interests include statistical machine learning, statistical signal processing and radar automatic target detection and recognition.

Meixi Zheng received the B.S. degree in electronic engineering from Xidian University, Xian, China, in 2019. She is currently working toward the M.S. degree in signal processing at the National Laboratory of Radar Signal Processing, Xidian University.

Her current research interests include radar automatic target recognition and statistical machine learning.

Hongwei Liu (Member, IEEE) received the M.S. and Ph.D. degrees in electronic engineering from Xidian University, Xi'an, China, in 1995 and 1999, respectively.

From 2001 to 2002, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. He is currently a Professor with the National Laboratory of Radar Signal Processing, Xidian University. His research interests include radar automatic target recognition, radar signal processing, and adaptive signal processing.