Weakly Supervised Complex Radio Scene Analysis Using Deep Variational Autoencoder

John Vorhies and Seung-Jun Kim*
Department of Computer Science & Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250
E-mail: {johnv3,sjkim}@umbc.edu

***Abstract*—Complex radio scene analysis methods are proposed to address scenarios in which multiple emitters coexist and share the spectrum. A passive receiver equipped with a multiantenna array is employed to detect, extract, and classify constituent signals from their mixtures by exploiting spatial separation and the unique time–frequency patterns of individual transmissions. We address this problem using machine learning algorithms trained on datasets of mixture signal samples paired with their class labels. In particular, a weakly supervised learning approach is adopted: the labels do not specify which component signals belong to which classes but only indicate the set of signal types present in each mixture. This greatly reduces the practical burden of dataset preparation while increasing the complexity of the learning task. To solve this challenge, we develop efficient algorithms based on deep variational inference with a successive signal estimation architecture. Numerical experiments demonstrate the effectiveness of the proposed methods.**

***Index Terms*—RF signal analysis, RF signal classification, variational autoencoder, weakly supervised machine learning.**

I. Introduction

The electromagnetic spectrum is often contested by various radio frequency (RF) emitters. They may coexist in the same frequency band, sharing the spectrum via temporal and spatial multiplexing, possibly employing different waveforms and modulations without central coordination. Analyzing such a complex radio scene using a passive observer to gain situational awareness of the spectrum is a challenging task. However, it is very useful for different applications including dynamic spectrum access, interference mitigation, radio map estimation, geolocation, unmanned aerial vehicle (UAV) operation, and RF fingerprinting [1]–[3].

Complex radio scene analysis typically involves critical subtasks such as signal detection, channel/direction estimation, signal separation, and signal classification. Recently, considerable attention has been devoted to data-driven machine learning methods for accomplishing such tasks. Dictionary learning was employed to detect and classify constituent signals in complex radios scenes [4]. Deep neural networks (DNNs) were used to estimate the direction of arrival (DoA) of signals of interest in [5], [6]. Beamforming weights were predicted via convolutional neural networks (CNNs) for spatial separation of component signals [7]. A transfer learning approach was taken to obtain a blind beamformer, after a classifier was first trained on component signals [8]. Unsupervised learning was employed for signal detection and channel estimation [9], [10]. In [11], DoA estimation was treated as a spatial compressed sensing problem tackled through unsupervised learning.

In this paper, we propose a machine learning framework for weakly supervised complex radio scene analysis. The learning algorithms are provided with a dataset containing mixtures of RF emissions as well as the signal class labels for the mixtures. It is weakly supervised in the sense that the labels do not reveal which individual emitters belong to which signal class, but only gives a multiset of signal types present in the mixture. Thus, the burden of dataset preparation is significantly reduced; however, the algorithm must handle a range of challenges, including component channel estimation, signal separation, and label association during training.

Our framework jointly tackles these challenges through a deep generative model of the mixture signals, where the detailed spectro-temporal patterns of the component signals are captured using DNNs. Two different models are proposed in this work depending on whether the labels are directly incorporated as latents. Then, approximate variational inference algorithms are developed to obtain the posterior distribution of the model parameters, including the latent representations and the spatial channels of the component signals. We build upon the idea in [10], where a deep successive signal extraction and cancellation architecture has been proposed for model efficiency. The architecture employs initial rough spatial separation of component signals followed by DNN-assisted spatio-temporal-spectral filtering for cleaner extraction. We significantly expand the model by incorporating weak labels and deeper generative processes. Our numerical tests show that the proposed methods achieves excellent signal separation, channel estimation, and signal classification performance, and compare favorably to the existing benchmarks.

The layout of the paper is as follows. In Sec. II, the signal model and problem statement are provided. In Sec. III, the proposed methods are derived in a variational inference framework using a DNN-based successive estimation architecture. Numerical experiments are conducted to verify the proposed methods in Sec. IV. Finally, the conclusion and future work are discussed in Sec. V.

This work was supported by the US National Science Foundation grant 2242412. Code available at https://github.com/johnvorhies/rf-dvae.
*Corresponding author.

II. Signal Model and Problem Statement

Let $\mathbf{x}(t) \in \mathbb{C}^M$, $t = 1, ..., T$, be a radio frequency (RF) signal converted to baseband and sampled by an M-element antenna array receiver. Assuming that there are J emitters in the environment, each transmitting with a single antenna over a flat-fading channel, $\mathbf{x}(t)$ can be expressed as

$$\mathbf{x}(t) = \sum_{j=1}^{J} \mathbf{h}_j s_j(t) + \mathbf{b}(t) \tag{1}$$

where $\mathbf{h}_j \in \mathbb{C}^M$ is the channel between the j-th emitter and the receiver array, $s_j(t)$ the j-th transmitted signal, and $\mathbf{b}(t) \in \mathbb{C}^M$ additive noise. One can rewrite (1) as

$$\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t) + \mathbf{b}(t), \quad t = 1, ..., T \tag{2}$$

where $\mathbf{H} := [\mathbf{h}_1, ..., \mathbf{h}_j]$, $\mathbf{s}(t) := [s_1(t), ..., s_J(t)]^\top$, and $\cdot^\top$ denotes transposition. Each transmission $s_j(t)$ belongs to one of C signal types or *classes*, such as Wi-Fi or Bluetooth, represented by $y_j \in \mathcal{C} := \{1, 2, \ldots, C\}$.

Our goal is to estimate $\{s_j(t)\}$ and $\{\mathbf{h}_j\}$ from mixture $\mathbf{x}(t)$ and predict the component signal types $\{y_j\}$ via a machine learning technique that can learn from examples the salient patterns of the component signals. An important practical constraint is that the examples are also in the form of mixture signals, and the learning algorithm must learn to separate them. Furthermore, the labels are given in mixtures as well. That is, instead of revealing which component belongs to which class, only the multiset $\mathcal{Y} = \{y_1, y_2, \ldots, y_J\}$ is given, which we refer to as a *weak* label. Naturally, providing only mixture signal samples and weak labels in the dataset significantly reduces the burden of dataset preparation, but also adds to the challenges of the learning process.

Let us consider the short-time Fourier transforms (STFTs) $\mathbf{x}(n, f)$, $\mathbf{s}(n, f)$ and $\mathbf{b}(n, f)$ of $\mathbf{x}(t)$, $\mathbf{s}(t)$ and $\mathbf{b}(t)$, respectively, with the time index $n = 1, ..., N$ and the frequency bin index $f = 1, ..., F$. It is assumed that $\mathbf{s}(n, f)$ is zero-mean complex Gaussian with covariance

$$\mathbf{R}_\mathbf{s}(n, f) = \text{diag}\{v_1(n, f), \ldots, v_J(n, f)\} \tag{3}$$

where $\text{diag}\{\cdot\}$ is a diagonal matrix with nonzero entries $\{\cdot\}$. It is also assumed that $s_j(n, f)$ is statistically independent from $s_{j'}(n', f')$ unless $j = j'$, $n = n'$, and $f = f'$, conditioned on $v_j(n, f)$ [12]. Then, it can be shown that $\mathbf{x}(n, f)$ is independent across n and f, and has a zero-mean complex Gaussian distribution with covariance

$$\mathbf{R}_\mathbf{x}(n, f) = \mathbf{H}\mathbf{R}_\mathbf{s}(n, f)\mathbf{H}^\mathcal{H} + \mathbf{R}_\mathbf{b} \tag{4}$$

where $\mathbf{R}_\mathbf{b}$ is the covariance of $\mathbf{b}(n, f)$, which is assumed to be stationary across n and f, and $\cdot^\mathcal{H}$ denotes Hermitian transpose.

To characterize the time-frequency (TF) patterns in the signal, deep generative models are adopted. Inspired by [13], we derive two variants depending on whether the class variables $\{y_j\}$ are included in the generative process.

A. Model (m1): Latent-Feature Discriminative Model

The TF pattern $\mathbf{V}_j := \{v_j(n, f)\}_{n,f} \in \mathbb{R}_+^{N \times F}$ for the j-th signal is modeled as

$$\mathbf{V}_j = f_{\boldsymbol{\theta}}(\mathbf{z}_j^{(1)}) \tag{5}$$

where $\mathbf{z}_j^{(1)} \in \mathbb{R}^K$ is a latent representation with $K \ll NF$, and $f_{\boldsymbol{\theta}}(\cdot)$ is a decoder described by a DNN, parameterized by $\boldsymbol{\theta}$. Upon defining $\mathbf{Z}^{(1)} := \{\mathbf{z}_j^{(1)}\}_{j=1}^J$ and $\boldsymbol{\zeta} := \{\mathbf{Z}^{(1)}, \mathbf{H}, \mathbf{R}_\mathbf{b}\}$, the likelihood of $\mathbf{X} := \{\mathbf{x}(n, f)\}_{n,f}$ with latents $\boldsymbol{\zeta}$ is given as

$$p_{\boldsymbol{\theta}}(\mathbf{X}|\boldsymbol{\zeta}) = \prod\nolimits_{n,f} \mathcal{N}\big(\mathbf{x}(n, f); \mathbf{0}, \mathbf{R}_\mathbf{x}(n, f)\big) \tag{6}$$

with a prior on $\boldsymbol{\zeta}$ given by

$$p(\boldsymbol{\zeta}) = \prod\nolimits_j p(\mathbf{z}_j^{(1)}) p(\mathbf{H}) p(\mathbf{R}_\mathbf{b}) \tag{7}$$

where $p(\mathbf{z}_j^{(1)})$ is the standard multivariate Gaussian distribution and $p(\mathbf{H})$ and $p(\mathbf{R}_\mathbf{b})$ are uniformly distributed.

The classification of $\{s_j(t)\}$ is done using the latents $\{\mathbf{z}_j^{(1)}\}$ sampled approximately from the posterior $p(\mathbf{Z}^{(1)}|\mathbf{X})$ as input features to the classifier, which improves performance over using $\{s_j(t)\}$ (or $\{s_j(n, f)\}$) directly due to the reduced dimension. We train a DNN model for $p_{\boldsymbol{\psi}}(y_j|\mathbf{z}_j^{(1)})$ with parameters $\boldsymbol{\psi}$.

B. Model (m2): Stacked Generative Weakly Supervised Model

Here, instead of (7), $\mathbf{Z}^{(1)}$ is modeled through another level of latents $\mathbf{Z}^{(2)}$ as well as the class variables $\mathbf{y} := \{y_j\}_j$ [13]. Specifically,

$$p_{\boldsymbol{\theta}}(\mathbf{X}, \boldsymbol{\zeta}, \mathbf{y}, \mathbf{Z}^{(2)}) = p_{\boldsymbol{\theta}}(\mathbf{X}|\boldsymbol{\zeta}) p_{\boldsymbol{\theta}}(\mathbf{Z}^{(1)}|\mathbf{y}, \mathbf{Z}^{(2)}) p(\mathbf{y}, \mathbf{Z}^{(2)}, \mathbf{H}, \mathbf{R}_\mathbf{b}) \tag{8}$$

$$p_{\boldsymbol{\theta}}(\mathbf{Z}^{(1)} \mid \mathbf{y}, \mathbf{Z}^{(2)}) = \prod\nolimits_j \mathcal{N}(\mathbf{z}_j^{(1)}; \boldsymbol{\mu}_{j,\boldsymbol{\theta}}^{(1)}, \text{diag}\{\boldsymbol{\sigma}_{j,\boldsymbol{\theta}}^{2(1)}\}) \tag{9}$$

$$p(\mathbf{y}, \mathbf{Z}^{(2)}, \mathbf{H}, \mathbf{R}_\mathbf{b}) = \prod\nolimits_j p(y_j) p(\mathbf{z}_j^{(2)}) p(\mathbf{H}) p(\mathbf{R}_\mathbf{b}) \tag{10}$$

with $p(y_j)$ a uniform distribution and $p(\mathbf{z}_j^{(2)})$ the standard multivariate Gaussian distribution.

Note that here $\mathbf{y}$ is treated as a latent variable similar to the unsupervised case in [13]. While the availability of the class label $\mathcal{Y}$ would normally allow supervised learning, the weak nature of the label prevents its straightforward application, especially with our successive architecture to be detailed in Sec. III-B. Our pragmatic approach is to treat $\mathbf{y}$ as an unobserved latent and only use the weak label during training to ensure it plays the role of class labels.

Since both $\mathbf{y}$ and $\mathbf{Z}^{(2)}$ are used as latents, class-shared traits can be captured in $\mathbf{Z}^{(2)}$ while class-specific variations are encoded in $\mathbf{y}$. Furthermore, stacking $\mathbf{Z}^{(1)}$ on top of $\mathbf{y}, \mathbf{Z}^{(2)}$ as in (9) yields representations in $\mathbf{Z}^{(1)}$ richer than those from **(m1)**.

III. Weakly Supervised Radio Scene Analysis

A. Variational Inference

For both proposed models, direct computation of the posterior distributions is intractable. Therefore, the variational inference approach is taken to derive approximate posteriors. Specifically, the *evidence lower bound* (ELBO) of the marginal log likelihood of $\mathbf{X}$ is maximized as a training objective for DNNs.

For **(m1)**, the ELBO is derived as [10]

$$\log p_{\boldsymbol{\theta}}(\mathbf{X}) \geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\zeta}|\mathbf{X})}\big[\log p_{\boldsymbol{\theta}}(\mathbf{X}|\boldsymbol{\zeta})\big] - \mathcal{D}(q_{\boldsymbol{\phi}}(\boldsymbol{\zeta}|\mathbf{X})\|p(\boldsymbol{\zeta})) \tag{11}$$

where $\mathcal{D}(\cdot\|\cdot)$ is the Kullback-Leibler (KL) divergence. The true posterior is approximated by an encoder model with parameters $\boldsymbol{\phi}$:

$$\begin{aligned} q_{\boldsymbol{\phi}}(\boldsymbol{\zeta}|\mathbf{X}) &= q_{\boldsymbol{\phi}}(\mathbf{Z}^{(1)}|\mathbf{H},\mathbf{R}_\mathbf{b},\mathbf{X})q_{\boldsymbol{\phi}}(\mathbf{H}|\mathbf{X})q_{\boldsymbol{\phi}}(\mathbf{R}_\mathbf{b}|\mathbf{X}) \\ &= \prod\nolimits_j q_{\boldsymbol{\phi}}(\mathbf{z}_j^{(1)}|\mathbf{H},\mathbf{R}_\mathbf{b},\mathbf{X})q_{\boldsymbol{\phi}}(\mathbf{H}|\mathbf{X})q_{\boldsymbol{\phi}}(\mathbf{R}_\mathbf{b}|\mathbf{X}) \end{aligned} \tag{12}$$

where

$$q_{\boldsymbol{\phi}}(\mathbf{z}_j^{(1)}|\mathbf{H},\mathbf{R}_\mathbf{b},\mathbf{X}) = \mathcal{N}(\mathbf{z}_j^{(1)};\boldsymbol{\mu}_{j,\boldsymbol{\phi}}^{(1)},\mathrm{diag}\{\boldsymbol{\sigma}_{j,\boldsymbol{\phi}}^{2(1)}\}) \tag{13}$$

$$q_{\boldsymbol{\phi}}(\mathbf{H}|\mathbf{X}) = \delta(\mathbf{H}-\hat{\mathbf{H}}) \tag{14}$$

$$q_{\boldsymbol{\phi}}(\mathbf{R}_\mathbf{b}|\mathbf{X}) = \delta(\mathbf{R}_\mathbf{b}-\hat{\mathbf{R}}_\mathbf{b}) \tag{15}$$

with point estimates $\hat{\mathbf{H}}$ and $\hat{\mathbf{R}}_\mathbf{b}$, and the Dirac delta function δ. Note that the KL divergence in (11) has a closed-form expression for Gaussian distributions [10] and the expectation can be approximately evaluated in practice using sample averages.

Similarly, one can derive the ELBO for **(m2)** as

$$\begin{aligned} \log p_{\boldsymbol{\theta}}(\mathbf{X}) \geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\zeta},\mathbf{y},\mathbf{Z}^{(2)}|\mathbf{X})}[&\log p_{\boldsymbol{\theta}}(\mathbf{X},\boldsymbol{\zeta},\mathbf{y},\mathbf{Z}^{(2)}) \\ &- \log q_{\boldsymbol{\phi}}(\boldsymbol{\zeta},\mathbf{y},\mathbf{Z}^{(2)}|\mathbf{X})] \end{aligned} \tag{16}$$

whose posterior is approximated by the encoder model

$$q_{\boldsymbol{\phi}}(\boldsymbol{\zeta},\mathbf{y},\mathbf{Z}^{(2)}|\mathbf{X}) = q_{\boldsymbol{\phi}}(\mathbf{Z}^{(2)}|\mathbf{y},\mathbf{Z}^{(1)})q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{Z}^{(1)})q_{\boldsymbol{\phi}}(\boldsymbol{\zeta}|\mathbf{X}) \tag{17}$$

where

$$q_{\boldsymbol{\phi}}(\mathbf{Z}^{(2)}|\mathbf{y},\mathbf{Z}^{(1)}) = \prod\nolimits_j q_{\boldsymbol{\phi}}(\mathbf{z}_j^{(2)}|\, y_j,\mathbf{z}_j^{(1)}) \tag{18}$$

$$q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{Z}^{(1)}) = \prod\nolimits_j q_{\boldsymbol{\phi}}(y_j|\mathbf{z}_j^{(1)}) \tag{19}$$

with

$$q_{\boldsymbol{\phi}}(\mathbf{z}_j^{(2)}|y_j,\mathbf{z}_j^{(1)}) = \mathcal{N}(\mathbf{z}_j^{(2)};\boldsymbol{\mu}_{j,\boldsymbol{\phi}}^{(2)},\mathrm{diag}\{\boldsymbol{\sigma}_{j,\boldsymbol{\phi}}^{2(2)}\}) \tag{20}$$

$$q_{\boldsymbol{\phi}}(y_j|\mathbf{z}_j^{(1)}) = \mathrm{Cat}(y_j;\boldsymbol{\gamma}_{j,\boldsymbol{\phi}}) \tag{21}$$

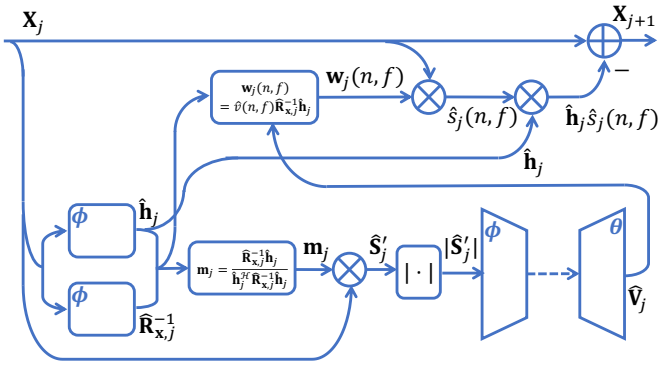


Fig. 1. Successive estimation architecture.

for the categorical distribution $\mathrm{Cat}(y_j;\boldsymbol{\gamma}_{j,\boldsymbol{\phi}})$ with parameters $\boldsymbol{\gamma}_{j,\boldsymbol{\phi}} \in \mathbb{R}^C$. It can be verified that the ELBO in (16) is equal to

$$\begin{aligned} &\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{Z}^{(1)})}\Big\{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}^{(1)}|\hat{\mathbf{H}},\hat{\mathbf{R}}_\mathbf{b},\mathbf{X})}\Big[\\ &\log p_{\boldsymbol{\theta}}(\mathbf{X}|\hat{\mathbf{R}}_\mathbf{b},\hat{\mathbf{H}},\mathbf{Z}^{(1)}) - \mathcal{D}\left(q_{\boldsymbol{\phi}}(\mathbf{Z}^{(2)}|\mathbf{y},\mathbf{Z}^{(1)})\|p(\mathbf{Z}^{(2)})\right)\Big] \\ &-\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}^{(2)}|\mathbf{y},\mathbf{Z}^{(1)})}\left[\mathcal{D}\left(q_{\boldsymbol{\phi}}(\mathbf{Z}^{(1)}|\hat{\mathbf{H}},\hat{\mathbf{R}}_\mathbf{b},\mathbf{X})\|p_{\boldsymbol{\theta}}(\mathbf{Z}^{(1)}|\mathbf{y},\mathbf{Z}^{(2)})\right)\right]\Big\} \\ &+\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}^{(1)}|\hat{\mathbf{H}},\hat{\mathbf{R}}_\mathbf{b},\mathbf{X})}\left[\mathcal{H}\left(q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{Z}^{(1)})\right)\right] + \mathrm{const.} \end{aligned} \tag{22}$$

where $\mathcal{H}(\cdot)$ denotes the entropy and const. represents terms which are constant with respect to (w.r.t) $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ and therefore have no effect on the gradient computation.

B. Successive Estimation Architecture

DNNs are employed to infer distributional parameters of the latents such as $\{\boldsymbol{\mu}_{j,\boldsymbol{\theta}}^{(1)},\boldsymbol{\sigma}_{j,\boldsymbol{\theta}}^{2(1)}\}$, $\{\boldsymbol{\mu}_{j,\boldsymbol{\phi}}^{(1)},\boldsymbol{\sigma}_{j,\boldsymbol{\phi}}^{2(1)}\}$, $\{\boldsymbol{\mu}_{j,\boldsymbol{\phi}}^{(2)},\boldsymbol{\sigma}_{j,\boldsymbol{\phi}}^{2(2)}\}$, $\{\boldsymbol{\gamma}_{j,\boldsymbol{\phi}}\}$, as well as $\hat{\mathbf{H}}$ (for simplicity, a prespecified constant is used for $\hat{\mathbf{R}}_{\boldsymbol{b}}$ in this work). In principle, one could train a large DNN that maps the input to the desired parameters. However, such a black box approach would require a large dataset. For efficiency and scalability, a successive estimation architecture is employed as in [10], which can process the input component-by-component in J stages. In particular, rough spatial separation of the mixture is attempted, with algorithmic structure inspired by array signal processing methods.

First, rather than estimating $\mathbf{h}_j$ for all j at once, a DNN is used to map the j-th stage input $\mathbf{X}_j$ to a single component's channel $\hat{\mathbf{h}}_j$. In the first stage, $\mathbf{X}_1 = \mathbf{X}$. In addition, a DNN is trained to obtain an estimate of the inverse of the signal covariance matrix $\hat{\mathbf{R}}_{\mathbf{x},j}^{-1}$. Then, filtering in the spatial dimension is done via a minimum-variance distortion-less response (MVDR) beamformer

$$\mathbf{m}_j := \frac{\hat{\mathbf{R}}_{\mathbf{x},j}^{-1}\hat{\mathbf{h}}_j}{\hat{\mathbf{h}}_j^{\mathcal{H}}\mathbf{R}_{\mathbf{x},j}^{-1}\hat{\mathbf{h}}_j} \tag{23}$$

to acquire a rough estimate of the j-th component signal as

$$\hat{s}'_j(n,f) = \mathbf{m}_j^{\mathcal{H}}\mathbf{x}_j(n,f). \tag{24}$$

One can view this step as a spatial attention mechanism to significantly reduce the dimension of the VAE input [14].

Next, the magnitude of $\hat{s}'_j(n,f)$ (or $|\hat{\mathbf{S}}'_j| := \{|\hat{s}'_j(n,f)|\}_{n,f}$) is mapped to $\boldsymbol{\mu}^{(1)}_{j,\boldsymbol{\phi}}, \boldsymbol{\sigma}^{2(1)}_{j,\boldsymbol{\phi}}$. Then, from sample $\hat{\mathbf{z}}^{(1)}_j \sim \mathcal{N}(\mathbf{z}^{(1)}_j; \boldsymbol{\mu}^{(1)}_{j,\boldsymbol{\phi}}, \boldsymbol{\sigma}^{2(1)}_{j,\boldsymbol{\phi}})$, $\hat{\mathbf{V}}_j$ is estimated per (5). Now with $\{\hat{v}_j(n,f)\}_{n,f} = \hat{\mathbf{V}}_j$, one can get a refined estimate of $s_j(n,f)$ by filtering in space, time, and frequency, similar to multichannel Wiener filtering [15]:

$$\mathbf{w}_j(n,f) = \hat{v}_j(n,f)\hat{\mathbf{R}}^{-1}_{\mathbf{x}_j}\hat{\mathbf{h}}_j \tag{25}$$

$$\hat{s}_j(n,f) = \mathbf{w}_j(n,f)^{\mathcal{H}}\mathbf{x}_j(n,f). \tag{26}$$

The estimated component signal $\hat{s}_j(n,f)$ is then subtracted from $\mathbf{X}_j$ to yield the next stage input $\mathbf{X}_{j+1}$ as

$$\mathbf{x}_{j+1}(n,f) = \mathbf{x}_j(n,f) - \hat{\mathbf{h}}_j\hat{s}_j(n,f), \quad \forall\, n,f. \tag{27}$$

The stages are repeated until all J components are estimated. Fig. 1 depicts the overall successive estimation process.

C. DNN Training

The training loss is a combination of the negative ELBO and the cross-entropy loss for the classifier. To use the weak labels $\mathcal{Y} = \{y_1, \ldots, y_J\}$ for training, one must determine which of the y_j's each stage's extracted component signal corresponds to. This can be done by finding a permutation $\pi : \{1,\ldots,J\} \to \{1,\ldots,J\}$ that minimizes the classifier training loss [16], [17]. For model **(m1)**, the training loss function is given by

$$\mathcal{L}_1(\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}) = -\mathcal{E}_1(\boldsymbol{\theta},\boldsymbol{\phi}) + \lambda_1 \min_{\pi}\left[-\sum_{j=1}^{J}\log(p_{\boldsymbol{\psi}}(y_j|\mathbf{z}^{(1)}_{\pi(j)}))\right] \tag{28}$$

where $\mathcal{E}_1$ is the ELBO in (11), $\lambda_1 > 0$ is a prespecified weight, and the expression inside the brackets in (28) is the cross-entropy loss for classifier $p_{\boldsymbol{\psi}}$. The optimal permutation can be found by solving a linear assignment problem using efficient algorithms such as the Hungarian method [18]. The training is done by minimizing the value of $\mathcal{L}_1$ averaged over the dataset w.r.t. $\boldsymbol{\theta}, \boldsymbol{\phi}$ and $\boldsymbol{\psi}$. To facilitate the computation of the gradient of $\mathcal{E}_1$ w.r.t. $\boldsymbol{\phi}$, which involves samples taken from the Gaussian distribution parameterized by $\boldsymbol{\phi}$, one can use the standard reparameterization trick [19].

For model **(m2)**, the loss function is similarly given as

$$\mathcal{L}_2(\boldsymbol{\phi},\boldsymbol{\theta}) = -\mathcal{E}_2(\boldsymbol{\theta},\boldsymbol{\phi}) + \lambda_2 \min_{\pi}\left[-\sum_{j=1}^{J}\log(q_{\boldsymbol{\phi}}(y_j|\mathbf{z}^{(1)}_{\pi(j)}))\right] \tag{29}$$

where $\mathcal{E}_2$ is the ELBO in (16). Note that in **(m2)** the classifier $q_{\boldsymbol{\phi}}(y|\mathbf{z}^{(1)})$ is automatically learned as part of the encoder model.

One issue encountered with the gradient computation of $\mathcal{E}_2$ w.r.t. $\boldsymbol{\phi}$ is that $\{\hat{y}_j\}$ sampled from $\prod_j q_{\boldsymbol{\phi}}(y_j|\mathbf{z}^{(1)}_j)$ are categorical random variables and thus are not differentiable. One could analytically evaluate the expectation w.r.t. $\{y_j\}$ as in [13] but this increases the computational complexity significantly. Instead, we adopt Gumbel-softmax reparameterization [20]. Note first that $\hat{y}_j \in \mathcal{C}$ is converted to a one-hot vector $\mathbf{e}_{\hat{y}_j} \in \mathbb{R}^C$, which is then used as input to the DNN for $q_{\boldsymbol{\phi}}(\mathbf{z}^{(2)}_j|y_j, \mathbf{z}^{(1)}_j)$. Gumbel-softmax approximates $\mathbf{e}_{\hat{y}_j}$ with $\hat{\mathbf{e}}_{\hat{y}_j}$ whose c-th element is given by

$$[\hat{\mathbf{e}}_{\hat{y}_j}]_c = \frac{\exp((\log([\boldsymbol{\gamma}_{j,\boldsymbol{\phi}}]_c) + g_c)/\tau)}{\sum_{c'=1}^{C}\exp((\log([\boldsymbol{\gamma}_{j,\boldsymbol{\phi}}]_{c'}) + g_{c'})/\tau)}, \; c \in \mathcal{C} \tag{30}$$

where $\{g_c\}_{c=1}^{C}$ are i.i.d. samples drawn from Gumbel$(0,1)$, $[\boldsymbol{\gamma}]_c$ represents the c-th entry of vector $\boldsymbol{\gamma}$, and $\tau > 0$ is a temperature parameter. Note that $\hat{\mathbf{e}}_{\hat{y}_j}$ is differentiable w.r.t. $\boldsymbol{\gamma}_{j,\boldsymbol{\phi}}$ and approaches $\mathbf{e}_{\hat{y}_j}$ as $\tau \to 0$. During training, the value of τ is annealed from 1.0 to 0.1 with an exponential decay.

IV. Numerical Experiments

A. Test Setup

The proposed algorithms were tested using RF signals collected at 2.4 GHz using a software-defined radio [4]. Six types of signals were collected: Wi-Fi with high occupancy, Wi-Fi with low occupancy, Bluetooth, Bluetooth Low Energy (BLE), and two different proprietary frequency-hopping spread spectrum (FHSS) transmissions. A total of $2,000$ STFT snapshots with $F = 64$ and $N = 66$ were obtained, each corresponding to a bandwidth of 40 MHz and a time duration of 100 ms. The signals were collected using a single receive antenna. The multi-antenna mixture signals were then synthesized by using steering vectors of a uniform linear array with random incident angles. The number J of the component signals was also varied randomly in $\{1,\ldots,6\}$. It was assumed in this work that the mixtures contain no more than one component per signal type but the developed algorithms can also accommodate the case with multiple component signals belonging to the same class.

In this experiment, neural networks are implemented with a combination of convolutional neural networks and multi-layer perceptrons (MLP). Full details of the implementation and training parameters are available in the supplementary code repository. In short, we utilize the same architecture as in [10], with MLPs added for the classifier and the encoder/decoder for $\mathbf{z}^{(2)}_j$. These networks are small as they operate in the latent space, ensuring fair comparison to the unsupervised VAE with only a marginal increase in model capacity.

The algorithms were trained at the signal-to-noise power ratio (SNR) of 40 dB and tested at varying SNR levels between 0 and 40 dB. Different SNR values were simulated by adding Gaussian noise to the mixtures. For simplicity, $\hat{\mathbf{R}}_{\mathbf{b}}$ is set to the true $\mathbf{R}_{\mathbf{b}} = N_0\mathbf{I}$, which is assumed known. The model was supplied with the correct number J of component signals in the mixture. The estimation of J from the input is left for future work.

B. Results

To evaluate the performance of the proposed models, the fidelity of the estimated component signals and channel vectors, as well as the classification accuracy were assessed. The

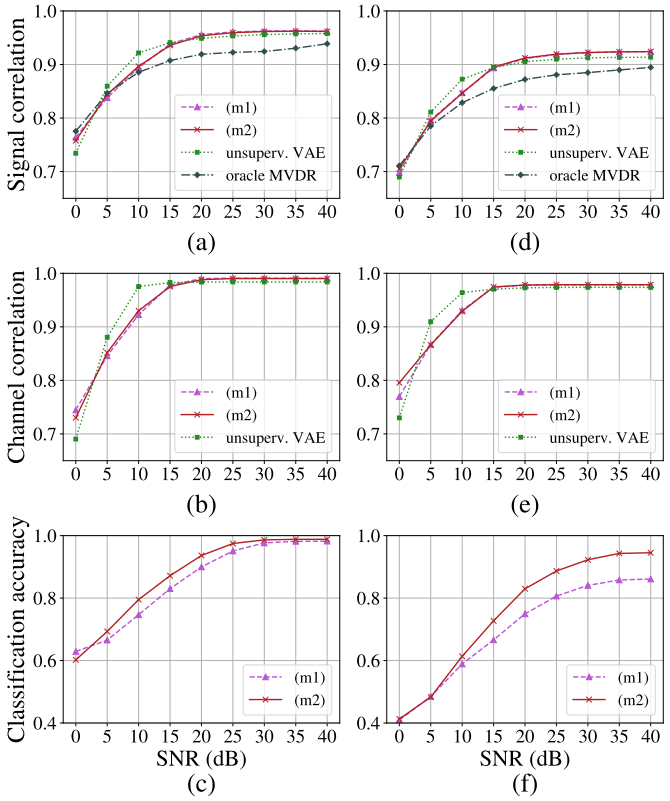


Fig. 2. Performance comparison. (a)–(c) $M = 6$ antennas. (d)–(f) $M = 4$ antennas.

final component signals $\check{\mathbf{s}}(n,f) := [\check{s}_1(n,f),\ldots,\check{s}_J(n,f)]^\top$ were obtained through multichannel Wiener filtering as

$$\check{\mathbf{s}}(n,f) = \hat{\mathbf{R}}_s(n,f)\hat{\mathbf{H}}^{\mathcal{H}}\hat{\mathbf{R}}_{\mathbf{x}}^{-1}(n,f)\mathbf{x}(n,f) \tag{31}$$

where $\hat{\mathbf{R}}_{\mathbf{x}}(n,f)$ is computed from (4) with $\mathbf{H}$, $\mathbf{R}_{\mathbf{s}}(n,f)$ and $\mathbf{R}_{\mathbf{b}}$ replaced with $\hat{\mathbf{H}}$, $\hat{\mathbf{R}}_{\mathbf{s}}(n,f) := \mathrm{diag}\{\hat{v}_j(n,f)\}_j$, and $\hat{\mathbf{R}}_{\mathbf{b}}$, respectively. Then, the correlation coefficients between true component STFTs $\{s_j(n,f)\}$ and the estimates $\check{s}_j(n,f)\}$ were computed and averaged. Likewise, the correlation coefficients between true $\mathbf{H}$ and estimated $\hat{\mathbf{H}}$ were computed.

Figs. 2(a)–(c) show the performance when the number of antenna elements is $M = 6$, which is the same as the maximum value that J can assume. Fig. 2(a) depicts the signal correlation performance, (b) the channel correlation, and (c) the classification accuracy. For the signal and channel correlations, the unsupervised VAE-based algorithm in [10] was also employed for comparison. The algorithm is based on a model very close to **(m1)** in this paper, but does not exploit the (weak) labels. We also compare the signal correlation performance due to an oracle MVDR beamformer, which uses the true channels $\mathbf{H}$ to separate the signals spatially, but does not employ DNN-based nonlinear processing. It can be seen that both the **(m1)** and **(m2)** models outperform the unsupervised VAE in terms of the signal and channel estimation performance at high SNRs ($\geq$ 20 dB), indicating that incorporating labels helps component extraction and channel estimation. However, in the low SNRs, the proposed algorithms are seen more sensitive than the unsupervised VAE to the SNR mismatch between the training and the test setups. At SNR $\geq$ 10 dB, the oracle MVDR beamformer is outperformed by the deep models in terms of the signal separation performance. While models **(m1)** and **(m2)** show very similar performances in Figs. 2(a) and (b), it can be observed in Fig. 2(c) that **(m2)** yields improved classification performance over **(m1)**, thanks to the extra level of latents which incorporate the signal classes.

Figs. 2(d)–(f) depict the performance when $M = 4$, highlighting the ability of the DNN models when there may be more component signals than the antennas. The performance curves show that the proposed methods can effectively mitigate the significant leakages in the spatial filtering step with the help of the VAE based spectro-temporal processing. It can be seen from Figs. 2(d) and (e) that the proposed models outperform the unsupervised VAE for SNR $\geq$ 15 dB in this case. Furthermore, the superiority of model **(m2)** to **(m1)** in classification is more evident in Fig. 2(f) in this challenging scenario.

An example mixture signal spectrogram is shown in Fig. 3

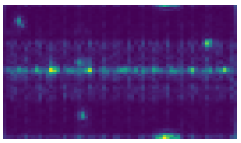


Fig. 3. A mixture signal example for $J = 6$ component signals with SNR = 20 dB.

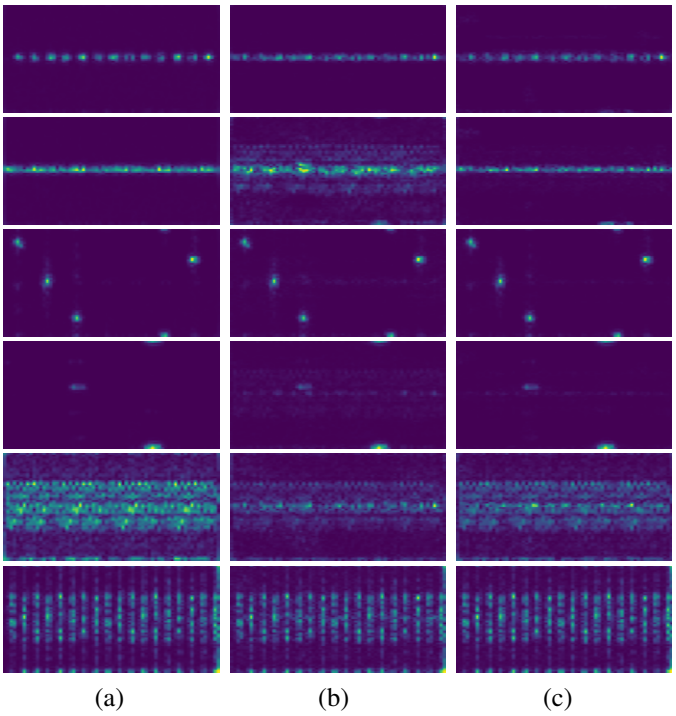


Fig. 4. Comparison of ground truth component signals (a), and estimated component signals for $M = 4$ antennas (b) and $M = 6$ antennas (c) using **(m2)**. Average correlation is 0.8936 for $M = 4$, 0.9557 for $M = 6$.

with $J = 6$ component signals. This mixture contains a component from each signal class, and has complete overlap in time and frequency with various levels of component signal power. Noise is added to the mixture, resulting in SNR = 20 dB. The component signals are shown in Fig. 4 using **(m2)**. The ground truth component signals in Fig. 4(a) are compared to the estimated component signals for $M = 4$ and $M = 6$ antennas in Fig. 4(b) and Fig. 4(c), respectively. The method achieves an average correlation between the ground truth components and the estimated components of 0.8936 for $M = 4$, and 0.9557 for $M = 6$.

V. Conclusion

Weakly-supervised deep generative models have been proposed for complex RF scene analysis. A successive estimation architecture has been employed for efficient mixture signal separation and classification. Learning occurs with weak labels of the signal classes present in the mixture, greatly reducing the overhead of dataset collection. The methods showed improved performance exploiting the class labels especially in the challenging cases where the number of antennas is smaller than the number of emitters. Future work includes estimation of the number of component signals, detection of unknown emitters, and equalization in frequency-selective channels.

References

[1] R. K. Saha and J. M. Cioffi, "Dynamic spectrum sharing for 5G NR and 4G LTE coexistence - a comprehensive review," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 795–835, 2024.

[2] E. Widdison and D. G. Long, "A review of linear multilateration techniques and applications," *IEEE Access*, vol. 12, pp. 26 251–26 266, 2024.

[3] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A review of radio frequency fingerprinting techniques," *IEEE J. Radio Freq. Identif.*, vol. 4, pp. 222–233, 2020.

[4] H. Chen and S.-J. Kim, "Robust RF mixture signal recognition using discriminative dictionary learning," *IEEE Access*, vol. 9, pp. 141 107–141 120, Oct. 2021.

[5] G. K. Papageorgiou, M. Sellathurai, and Y. C. Eldar, "Deep networks for direction-of-arrival estimation in low SNR," *IEEE Trans. Signal Process.*, vol. 69, pp. 3714–3729, 2021.

[6] J. Ma, M. Wang, Y. Chen, and H. Wang, "Deep convolutional network-assisted multiple direction-of-arrival estimation," *IEEE Signal Process. Lett.*, vol. 31, pp. 576–580, 2024.

[7] Z. Liao, K. Duan, J. He, Z. Qiu, and B. Li, "Robust adaptive beamforming based on a convolutional neural network," *Electronics*, vol. 12, no. 12, 2023.

[8] M. Wentz, J. Capper, B. Kurien, K. Forsythe, and K. Chowdhury, "Blind beamforming via deep learning-based signal classification and transfer learning," *IEEE Trans. on Cogn. Commun. Netw.*, 2025, to appear.

[9] H. Chen and S.-J. Kim, "Unsupervised radio scene analysis using neural expectation maximization," in *Proc. of the IEEE Military Comm. Conf. (MILCOM)*, Rockville, MD, Nov. 2022, pp. 368–373.

[10] ——, "Unsupervised radio scene analysis using variational autoencoder," in *Proc. of the IEEE Int. Conf. Commun. (ICC)*, Rome, Italy, May 2023, pp. 4591–4596.

[11] Z. Zhang, X. Qu, W. Li, H. Miao, and F. Liu, "DOA estimation method based on unsupervised learning network with threshold Capon spectrum weighted penalty," *IEEE Signal Process. Lett.*, vol. 31, pp. 701–705, 2024.

[12] C. Fevotte and J.-F. Cardoso, "Maximum likelihood approach for blind audio source separation using time-frequency Gaussian source models," in *Proc. of the IEEE Wksp. Appl. Sig. Process. Audio Acoust. (ASPAA)*, Oct. 2005.

[13] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 27, 2014.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[15] S. Braun, A. Kuklasiński, O. Schwartz, O. Thiergart, E. A. P. Habets, S. Gannot, S. Doclo, and J. Jensen, "Evaluation and comparison of late reverberation power spectral density estimators," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, pp. 1056–1071, 2018.

[16] Y. Zhang, J. Hare, and A. Prugel-Bennett, "Deep set prediction networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019.

[17] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-centric learning with slot attention," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 11 525–11 538.

[18] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, Mar. 1995.

[19] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014.

[20] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.