

CSI CLUSTERING WITH VARIATIONAL AUTOENCODING

Michael Baur, Michael Würth, Michael Koller, Vlad-Costin Andrei, Wolfgang Utschick

Department of Electrical and Computer Engineering, Technical University of Munich, Germany
Email: {mi.baur, michael.wuerth, michael.koller, vlad.andrei, utschick}@tum.de

ABSTRACT

The model order of a wireless channel plays an important role for a variety of applications in communications engineering, e.g., it represents the number of resolvable incident wavefronts with non-negligible power incident from a transmitter to a receiver. Areas such as direction of arrival estimation leverage the model order to analyze the multipath components of channel state information. In this work, we propose to use a variational autoencoder to group unlabeled channel state information with respect to the model order in the variational autoencoder latent space in an unsupervised manner. We validate our approach with simulated 3GPP channel data. Our results suggest that, in order to learn an appropriate clustering, it is crucial to use a more flexible likelihood model for the variational autoencoder decoder than it is usually the case in standard applications.

Index Terms— Variational autoencoder, generative modeling, latent space, vector channels, clustering

1. INTRODUCTION

Wireless channels are characterized by different multipath components, each comprising individual parameters such as directions of arrival (DoA) at the receiver [1]. Every impinging wavefront at a receiver antenna array from a transmitter represents one of the multiple propagation paths. The number of resolvable propagation paths is typically related to the model order and both play an important role in many fields of communications engineering such as DoA estimation [2] or channel estimation with compressive sensing [3], where the model order or the number of resolvable paths is often assumed to be known. In this work, we investigate the use of generative modeling with variational autoencoders (VAE) for the clustering of unlabeled channel state information (CSI) with respect to its number of resolvable paths.

The idea of clustering wireless channels is an ongoing topic of research. An early work of multipath clustering is performed in [4] by first employing a generalized expectation-maximization (EM) algorithm, which delivers the clusters. Second, the obtained clusters are determined by inspection.

In [5], a Kernel-power-density based algorithm is proposed to solve this clustering problem. The algorithm allows for clustering of the multipath components, e.g. path delay or DoA. Another framework adapts the variational Bayesian EM algorithm towards the multipath clustering task [6]. All of the above methods have in common that clustering is performed on the basis of the (extracted) multipath components, but not with respect to the number of resolvable paths and not directly on realizations of the vector channels.

The task of clustering data, in our case wireless channels, can be interpreted as grouping data that are similar and separating data that are dissimilar with respect to a predefined criterion. Representation learning based on unsupervised machine learning (ML) is a candidate to fulfill this objective as it aims to find a meaningful representation of the data that facilitates the search for beneficial information [7]. ML has already been widely applied in communications engineering and will be important for the development of future wireless systems. The VAE [8] is a suitable tool to generate artificial data from unknown distributions solely represented by training samples. A particular feature of the VAE is the manifestation of a structured representation of the underlying unlabeled data by means of clustering in the latent space. These clusters can be interpreted as representatives of different features of the data and can be used to generate new data with controllable features by the VAE decoder. The clustering capability of VAEs has been demonstrated for multiple image datasets. In this work, we transfer this result to CSI data and demonstrate the capability of the VAE for CSI clustering.

The contribution of this work is as follows. We show that a VAE can cluster CSI data with respect to the number of paths, which we use interchangeably for the model order in this work, although knowing that non-resolvable paths collapse the determinable model order. To this end, we propose a VAE decoder that models a conditional Gaussian posterior with a circular covariance matrix compared to the standard choice which assumes a Gaussian posterior with a scaled identity covariance matrix. We observe that this choice of a more expressive VAE decoder significantly improves the ability of the VAE encoder to produce a better clustered output. In contrast to other methods, our method solely works with realizations of the vector channels and does not require any further preprocessing steps.

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy within the project 6G Future Lab Bavaria.

2. CHANNEL MODEL

A base station with a uniform linear array (ULA) equipped with M antennas is considered, which receives signals from a single antenna mobile terminal. Similar to [9], we assume conditionally Gaussian channels, which are frequency-flat and fast-fading. The channels are distributed according to

$$\mathbf{h} | \boldsymbol{\delta} \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{\mu}_{\boldsymbol{\delta}}, \mathbf{C}_{\boldsymbol{\delta}}), \quad \boldsymbol{\delta} \sim p(\boldsymbol{\delta}) \quad (1)$$

with mean vector $\boldsymbol{\mu}_{\boldsymbol{\delta}} \in \mathbb{C}^M$ and covariance matrix $\mathbf{C}_{\boldsymbol{\delta}} \in \mathbb{C}^{M \times M}$. The random variable $\boldsymbol{\delta}$ with distribution $p(\boldsymbol{\delta})$ describes some abstract prior which can be interpreted, for example, as the multipath components of the channel. For the duration of one coherence interval, the covariance matrix is assumed to be constant. Conditionally Gaussian channels are a common modeling choice in communication scenarios. They appear for example in the stochastic channel models of the 3rd Generation Partnership Project (3GPP) [10]. In particular, the covariance matrices in the 3GPP channel models follow the relation $\mathbf{C}_{\boldsymbol{\delta}} = \int_{-\pi}^{\pi} g(\theta; \boldsymbol{\delta}) \mathbf{a}(\theta) \mathbf{a}(\theta)^H d\theta$, where $g(\theta; \boldsymbol{\delta}) \geq 0$ denotes a power angular spectrum and $\mathbf{a}(\theta)$ the steering vector corresponding to the array geometry.

It is known that a covariance matrix $\mathbf{C}_{\boldsymbol{\delta}}$ of a ULA with half-wavelength spacing exhibits a Toeplitz structure, which can be well approximated by a circular matrix for a large number of antennas [9]. An interesting property of a circular matrix is that it is diagonalizable by the discrete Fourier transform (DFT) matrix $\mathbf{F} \in \mathbb{C}^{M \times M}$, such that $\mathbf{C}_{\boldsymbol{\delta}} = \mathbf{F}^H \text{diag}(\mathbf{c}_{\boldsymbol{\delta}}) \mathbf{F}$, $\mathbf{c}_{\boldsymbol{\delta}} \in \mathbb{R}^M$. Consequently, using DFT, the covariance matrix of the transformed channels $\mathbf{x} = \mathbf{F}\mathbf{h}$ is diagonal and corresponds to $\text{diag}(\mathbf{c}_{\boldsymbol{\delta}})$. In Section 3, we use the circular approximation of the Toeplitz covariance matrix of a ULA in conjunction with Fourier-transformed channel data to design a VAE which is suitable for the task of CSI clustering with respect to the number of resolvable paths.

3. VARIATIONAL AUTOENCODER MODEL

This section reviews the VAE after highlighting some important aspects of variational inference (VI). Subsequently, we present a modification of the VAE framework which enables to learn the circular covariance matrices of the conditional Gaussian channel model defined in Section 2.

3.1. Variational Inference Background

The VAE has its origin in the field of VI [11]. The central term in VI is the evidence lower bound (ELBO)

$$\mathcal{L}(q) = \log p(\mathbf{x}) - \text{D}_{\text{KL}}(q(\mathbf{z}) \| p(\mathbf{z} | \mathbf{x})) \leq \log p(\mathbf{x}) \quad (2)$$

which involves the data log-likelihood $\log p(\mathbf{x})$ and the Kullback-Leibler (KL) divergence between the variational distribution $q(\mathbf{z})$ and the posterior $p(\mathbf{z} | \mathbf{x})$, that is,

$\text{D}_{\text{KL}}(q(\mathbf{z}) \| p(\mathbf{z} | \mathbf{x})) = \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z} | \mathbf{x})} d\mathbf{z}$. The variational distribution $q(\mathbf{z}) \in \mathcal{Q}$ stems from a family of distributions \mathcal{Q} and is supposed to match the posterior $p(\mathbf{z} | \mathbf{x})$ as closely as possible. In the case of equivalence, the KL divergence vanishes and the ELBO is equal to the data log-likelihood. The main problem for finding a suitable $q(\mathbf{z})$ is that the encountered integrals are usually intractable. VI circumvents this problem by turning the task of solving an integral into an optimization problem which reads as

$$\min_{q(\mathbf{z}) \in \mathcal{Q}} \text{D}_{\text{KL}}(q(\mathbf{z}) \| p(\mathbf{z} | \mathbf{x})). \quad (3)$$

A maximization of the ELBO therefore achieves two goals: the data log-likelihood is maximized and the best approximation in \mathcal{Q} to $p(\mathbf{z} | \mathbf{x})$ is found.

3.2. VAE with Scaled Identity Covariance Matrix

The VAE, independently proposed in [8] and [12], makes VI accessible to a broad audience. The reason for this is that a VAE consists of a relatively simple structure and is scalable to large data. This is achieved by using the reparameterization trick and neural networks to learn the parameters of the encountered distributions. In the context of a VAE, the variational distribution is usually explicitly conditioned on the data \mathbf{x} , which yields $q(\mathbf{z}) = q_{\phi}(\mathbf{z} | \mathbf{x})$. For the training of a VAE, the ELBO is often expressed as

$$\mathcal{L}(q) = \mathbb{E}_{q_{\phi}(\mathbf{z} | \mathbf{x})} [\log p_{\theta}(\mathbf{x} | \mathbf{z}) + \log p(\mathbf{z}) - \log q_{\phi}(\mathbf{z} | \mathbf{x})] \quad (4)$$

where $q_{\phi}(\mathbf{z} | \mathbf{x})$ represents the encoder network and $p_{\theta}(\mathbf{x} | \mathbf{z})$ the decoder network with neural network parameters ϕ and θ , respectively [13]. These networks are central components of a VAE and are shown in Fig. 1, which we explain in more detail later in this section. Since calculating the gradient with respect to the neural network parameters is difficult, the reparameterization trick is employed. It assumes that \mathbf{z} , which is also referred to as latent variable, is the result of a differentiable transformation of a random variable $\boldsymbol{\epsilon}$, which belongs to a simpler distribution, e.g., $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. In [13], it is proposed to optimize (4) with a single noise sample $\boldsymbol{\epsilon}$.

If we assume $\mathcal{N}_{\mathbb{C}}(\boldsymbol{\mu}_{\mathbf{x} | \mathbf{z}}, \mathbf{C}_{\mathbf{x} | \mathbf{z}})$, $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{z} | \mathbf{x}}, \text{diag}(\boldsymbol{\sigma}_{\mathbf{z} | \mathbf{x}}^2))$, and $\mathcal{N}(\mathbf{0}, \mathbf{I})$, for $p_{\theta}(\mathbf{x} | \mathbf{z})$, $q_{\phi}(\mathbf{z} | \mathbf{x})$, and $p(\mathbf{z})$, respectively, and omit constant terms we can reformulate the ELBO as

$$\begin{aligned} \mathcal{L}(q) = & -\log \det \mathbf{C}_{\mathbf{x} | \mathbf{z}} - (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x} | \mathbf{z}})^H \mathbf{C}_{\mathbf{x} | \mathbf{z}}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x} | \mathbf{z}}) \\ & + \frac{1}{2} \|\boldsymbol{\epsilon}\|^2 + \mathbf{1}^T \log \boldsymbol{\sigma}_{\mathbf{z} | \mathbf{x}} - \frac{1}{2} \|\mathbf{z}\|^2 \end{aligned} \quad (5)$$

with the all ones vector $\mathbf{1}$. A sketch of a VAE with the previous assumptions is depicted in Fig. 1. A data point is passed through the encoder to receive the vectors $\boldsymbol{\mu}_{\mathbf{z} | \mathbf{x}}$ and $\boldsymbol{\sigma}_{\mathbf{z} | \mathbf{x}}$, which are in turn used to reparameterize \mathbf{z} . The resulting latent vector \mathbf{z} is passed through the decoder to obtain the vector $\boldsymbol{\mu}_{\mathbf{x} | \mathbf{z}}$ and the matrix $\mathbf{C}_{\mathbf{x} | \mathbf{z}}$. These quantities could then

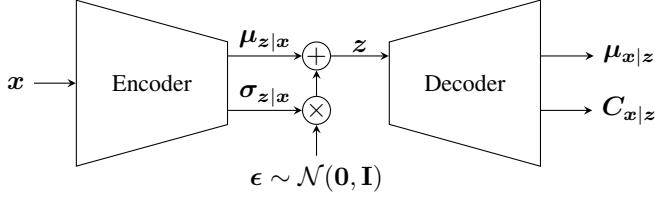


Fig. 1. Sketch of a VAE with Gaussian posteriors.

be used to generate new samples $\tilde{x} \sim \mathcal{N}_{\mathbb{C}}(\mu_{x|z}, C_{x|z})$. More generally, a newly generated sample is supposed to follow the true distribution of the data $p(x)$.

Since the size of outputs of the decoder for $C_{x|z}$ scales with M^2 and the inverse $C_{x|z}^{-1}$ in (5) is expensive to compute, it is uncommon to work with full covariance matrices. Instead, it is often assumed that the covariance matrix is a scaled identity, i.e., $C_{x|z} = \sigma_{x|z}^2 \mathbf{I}$. It is obvious that this approach simplifies the computation of (5) immensely. Its drawback is that the expressiveness of $p_{\theta}(x|z)$ is limited.

3.3. VAE with Diagonal Covariance Matrix

Only few methods exist in the literature to model the covariance matrix $C_{x|z}$ more expressively since the focus is mainly on the optimization of the mean $\mu_{x|z}$ instead of the whole distribution $p_{\theta}(x|z)$. One method uses a Cholesky decomposition of $C_{x|z}$ [13]. Another rarely employed approach uses a diagonal covariance matrix $C_{x|z} = \text{diag}(c_{x|z})$, as described in [8]. It is worth investigating because it increases the expressiveness of $p_{\theta}(x|z)$ and keeps the computation of (5) efficient. With this assumption, we can further simplify (5) to

$$\mathcal{L}(q) = \mathbf{1}^T \left(-\log c_{x|z} - c_{x|z}^{-1} \odot |x - \mu_{x|z}|^2 \right) + \frac{1}{2} \|\epsilon\|^2 + \mathbf{1}^T \log \sigma_{z|x} - \frac{1}{2} \|z\|^2. \quad (6)$$

The vector $c_{x|z}^{-1}$ represents the elementwise reciprocals of the corresponding vector and \odot represents an elementwise multiplication. At this point, we can draw the connection to the channel model presented in Section 2. We would like to learn a complete covariance matrix conditioned on every input channel. Since such a covariance matrix is too computationally expensive to learn, we propose to use its circular approximation in combination with DFT-transformed channels x . This allows us to learn the diagonal covariance matrices of the transformed channels by training a VAE that maximizes Equation (6). Note that a sample x is modeled as conditionally Gaussian with the VAE in Fig. 1, with mean value $\mu_{x|z}$ and diagonal covariance matrix $C_{x|z} = \text{diag}(c_{x|z})$. Also note that for the chosen dataset of conditionally Gaussian 3GPP channels in DFT domain, without loss of generality, the choice of a diagonal covariance matrix at the decoder output of the VAE exactly resembles the structure of the true covariance matrix. This relation further indicates why the proposed model is beneficial for our application as we can

Table 1. MMD hypothesis test TPRs for a comparison of channel data exhibiting the respective number of paths.

	1 path	2 paths	3 paths	4 paths	5 paths
1 path	0.02	0.26	0.65	0.92	0.98
2 paths	-	0.01	0.14	0.40	0.60
3 paths	-	-	0.01	0.10	0.23
4 paths	-	-	-	0.02	0.06
5 paths	-	-	-	-	0.00

approximate the true distribution of our channel data much better than with a scaled identity covariance matrix.

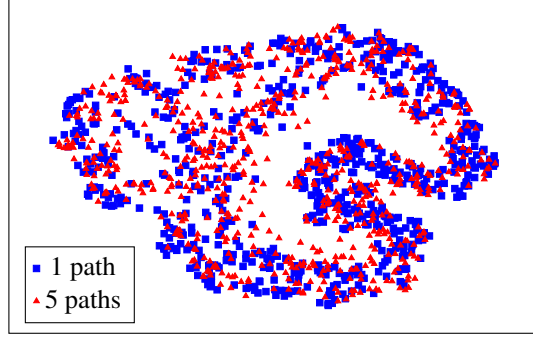
4. RESULTS

In this section, we first give an overview about implementation details. Subsequently, we present the channel clustering results with respect to the number of resolvable paths.

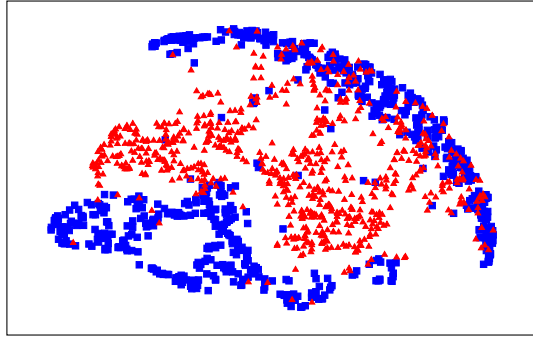
4.1. Dataset Description and Network Architecture

We create a dataset of wireless channels based on the elaborations in Section 2 for the 3GPP urban-macro channel model. To this end, we randomly sample 33 000 channels per model order, each belonging to a different covariance matrix. The number of antennas at the base station is set to 32. The first 30 000 channels are used for training and the remaining 3 000 channels for evaluation purposes. As this work is intended to be a first proof of concept, we create a dataset consisting only of channels with one or five propagation paths. The paths are generated without regard to their resolvability and with different path gains from a uniform distribution in the range of $[0, 1]$ such that the individual path gains add up to one. This leads to the dataset comparison results shown in Table 1. The applied maximum mean discrepancy (MMD) is a kernel based measure for the dissimilarity of two datasets. It can be fully evaluated using samples taken from the respective distributions. For more details we refer the reader to [14]. Based on the MMD, a hypothesis test can be designed to determine the dissimilarity of two datasets [15], which is displayed for channels with different numbers of paths in Table 1. Every table entry represents the true positive rate (TPR) that channels of the corresponding model orders belong to a different dataset. Clearly, within the considered margins, channels with only one path are most dissimilar to channels with five paths, which is the reason for the constitution of our dataset. The investigation of other cases is subject of future work.

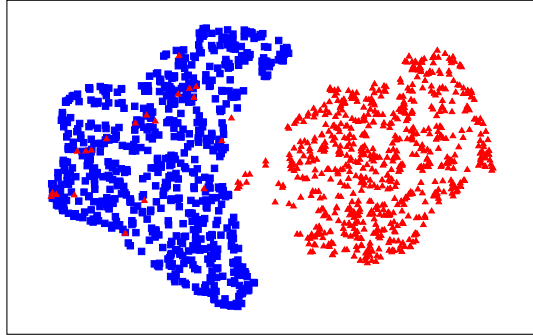
The encoder and decoder are implemented with a symmetric structure. It consists of three layers, where each layer is built of a 1D convolutional layer with kernel size 7 and stride 2, followed by a batch normalization layer and ReLU activation function. In the encoder, the convolutional layers map from 1 to 8 to 32 to 128 convolutional channels. In the decoder, this happens vice versa. In both networks, this composition is followed by a fully connected layer to map to the



(a) UMAP of raw data.



(b) UMAP of latent space of VAE with $C_{x|z} = \sigma_{x|z}^2 \mathbf{I}$.



(c) UMAP of latent space of VAE with $C_{x|z} = \text{diag}(c_{x|z})$.

Fig. 2. UMAP embeddings for different types of data.

correct dimension. The size of the latent space is set to 4. Here, we choose the lowest value for which we can still see distinct clusters in our experiments. Real and imaginary part of the complex valued channels are stacked to a vector, sized two times the amount of antennas. We use the Adam optimizer [16] with a learning rate of 5.3×10^{-4} and the method of free bits for optimization [13].

4.2. Learned CSI Clusterings with VAE

We show different 2D uniform manifold approximation and projection (UMAP) embeddings in this section [17]. It should be noted at this point that UMAP embeddings look different depending on the chosen hyperparameters. Distances between and sizes of clusters are not representative. UMAP em-

beddings are supposed to preserve local and global structures. As they are not descriptive for the embeddings, we omit the axes in the following illustrations. Note that latent space plots were created using previously unseen evaluation data and that data labels were used solely to illustrate results but were not available during unsupervised learning of the VAE.

We plot the UMAP embedding of the input channel data in Fig. 2(a) as a comparison to see the benefit of the learned VAE clustering. Clearly, no clusters are observable in the raw data. In contrast, Fig. 2(b) shows the UMAP embedding of the four-dimensional mean latent vectors $\mu_{z|x}$ corresponding to each input data point. We trained a VAE with scaled identity covariance matrix at the output of the decoder for this embedding, as explained in Section 3.2. The VAE manages more poorly than not to separate the channels that belong to one path from the channels that belong to five paths. While the model orders are somehow separated, the learned VAE encoder embedding is not useful for clustering.

When we used the non-standard VAE with diagonal covariance matrix at the output of the decoder on the same data, as highlighted in Section 3.3, the embedding changes dramatically. The channels with one path and five paths are clearly separated in the latent space in Fig. 2(c). Only a few channels with five paths are projected onto the one path cluster. The most likely reason for these "outliers" is that the corresponding channels have one dominant path and four weak paths, so the channels are more alike to one-path channels. The remaining channels are distinctly assigned to their correct clusters.

In Fig. 2, we see that adjusting the decoder's implicit likelihood model affects the structure of the VAE latent space. By employing a diagonal covariance matrix model at the decoder output, the stochastic model becomes more flexible. This indirectly forces the encoder to produce a more structured latent space to exploit the improved flexibility of the decoder.

5. CONCLUSION

We have shown in this work that it is possible to cluster CSI data with respect to their model order, represented by the number of propagation paths, using unsupervised learning techniques by training a VAE. It turned out that the choice of the implicit likelihood model represented by the decoder is beneficial for clustering in the latent space of the VAE. We achieved this by subjecting the channel data based on a ULA to a DFT matrix, as well as by using a VAE that provides conditional diagonal covariance matrices at the output of the decoder. In our future work, we will address further open questions, e.g., the investigation of different channel models, the extension to MIMO channels, and the model order analysis of measured data from real-world measurement campaigns. Another topic will be the focus on the actual generation of channel data with controllable features using autoencoding; initial results for single antenna channel impulse responses in [18] point the way.

6. REFERENCES

- [1] Mustafa Riza Akdeniz, Yuanpeng Liu, Mathew K. Samimi, Shu Sun, Sundeeep Rangan, Theodore S. Rappaport, and Elza Erkip, "Millimeter Wave Channel Modeling and Cellular Capacity Evaluation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1164–1179, 2014.
- [2] Hamid Krim and Mats Viberg, "Two Decades of Array Signal Processing Research: The Parametric Approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, 1996.
- [3] Christian R. Berger, Zhaohui Wang, Jianzhong Huang, and Shengli Zhou, "Application of Compressive Sensing to Sparse Channel Estimation," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 164–174, 2010.
- [4] Kai Yu, Qinghua Li, David Cheung, and Clifford Prtatie, "On the tap and cluster angular spreads of indoor WLAN channels," in *IEEE 59th Veh. Technol. Conf.*, 2004.
- [5] Ruisi He, Qingyong Li, Bo Ai, Yang Li, Ao Geng, Andreas F. Molisch, Vinod Kristem, Zhangdui Zhong, and Jian Yu, "A Kernel-Power-Density-Based Algorithm for Channel Multipath Components Clustering," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 11, pp. 7138–7151, 2017.
- [6] Yupeng Li, Jianhua Zhang, and Zhanyu Ma, "Clustering Analysis in the Wireless Propagation Channel with a Variational Gaussian Mixture Model," *IEEE Trans. Big Data*, vol. 6, no. 2, pp. 223–232, 2020.
- [7] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [8] Diederik P. Kingma and Max Welling, "Auto-Encoding Variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent.*, 2014.
- [9] David Neumann, Thomas Wiese, and Wolfgang Utschick, "Learning The MMSE Channel Estimator," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2905–2917, 2018.
- [10] 3GPP, "Spatial channel model for Multiple Input Multiple Output (MIMO) simulations (Release 16)," Tech. Rep., 3rd Generation Partnership Project (3GPP), 2020.
- [11] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe, "Variational Inference: A Review for Statisticians," *J. Am. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.
- [12] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014.
- [13] Diederik P. Kingma and Max Welling, "An Introduction to Variational Autoencoders," *Found. Trends® Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.
- [14] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola, "A Kernel Two-Sample Test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, 2012.
- [15] Wolfgang Utschick, Valentina Rizzello, Michael Joham, Zhengxiang Ma, and Leonard Piazzzi, "Learning the CSI Recovery in FDD Systems," *arXiv Prepr. arXiv2104.01322*, 2021.
- [16] Diederik P. Kingma and Jimmy Lei Ba, "Adam: A Method for Stochastic Optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, 2015.
- [17] Leland McInnes, John Healy, and James Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *J. Open Source Softw.*, vol. 3, no. 29, pp. 861, 2018.
- [18] Franz Weisser, Timo Mayer, Bessem Baccouche, and Wolfgang Utschick, "Generative-AI Methods for Channel Impulse Response Generation," in *Proc. 24th Int. ITG Work. Smart Antennas*, Eurecom, French Riviera, 2021.