Leveraging Variational Autoencoders for Parameterized MMSE Estimation

Michael Baur, *Graduate Student Member, IEEE*, Benedikt Fesl, *Graduate Student Member, IEEE*, and Wolfgang Utschick, *Fellow, IEEE*

***Abstract*—In this manuscript, we propose to use a variational autoencoder-based framework for parameterizing a conditional linear minimum mean squared error estimator. The variational autoencoder models the underlying unknown data distribution as conditionally Gaussian, yielding the conditional first and second moments of the estimand, given a noisy observation. The derived estimator is shown to approximate the minimum mean squared error estimator by utilizing the variational autoencoder as a generative prior for the estimation problem. We propose three estimator variants that differ in their access to ground-truth data during the training and estimation phases. The proposed estimator variant trained solely on noisy observations is particularly noteworthy as it does not require access to ground-truth data during training or estimation. We conduct a rigorous analysis by bounding the difference between the proposed and the minimum mean squared error estimator, connecting the training objective and the resulting estimation performance. Furthermore, the resulting bound reveals that the proposed estimator entails a bias-variance tradeoff, which is well-known in the estimation literature. As an example application, we portray channel estimation, allowing for a structured covariance matrix parameterization and low-complexity implementation. Nevertheless, the proposed framework is not limited to channel estimation but can be applied to a broad class of estimation problems. Extensive numerical simulations first validate the theoretical analysis of the proposed variational autoencoder-based estimators and then demonstrate excellent estimation performance compared to related classical and machine learning-based state-of-the-art estimators.**

***Index Terms*—Parameter estimation, variational autoencoder, conditional mean estimator, generative model, inverse problem.**

I. Introduction

A prominent class of machine learning (ML) techniques designed to learn data distributions based on samples are generative models (GMs) [1]. A successfully trained GM allows for likelihood evaluation and the creation of entirely new samples that follow the same distribution. Instances of GMs are the Gaussian mixture model (GMM) [2, Ch. 9], variational autoencoder (VAE) [3], [4], generative adversarial network (GAN) [5], and score-based model [6]. GMs provide a generative prior characteristic for the data distribution, which can be leveraged to solve sophisticated tasks such as inverse problems [7]. Exemplarily, the work in [8] utilizes a GAN to reconstruct images in a compressed sensing (CS)-fashioned way and [9] extends the approach to MRI images. Other image processing-related examples involve phase retrieval [10] and blind image deconvolution [11]. In the context of wireless communications, generative priors find application in channel estimation (CE) [12]–[16], which is also an inverse problem.

For the solution of an estimation task, a frequentist framework assumes the data to be deterministic and commonly constrains the estimator class to be unbiased in search for a minimum variance unbiased estimator [17]. In opposition, if a Bayesian approach is considered, it is well-known in estimation theory that the conditional mean estimator (CME) delivers minimum mean squared error (MMSE) estimates [18, Ch. 10]. Therefore, a (parameterized) Bayesian estimator's goal should be approximating the CME. Moreover, a fundamental aspect of the Bayesian framework is modeling the data as a random variable (RV), enabling the incorporation of a prior distribution into the estimation process. The result is an excellent estimation performance if the prior distribution accurately models the data, e.g., in the form of a generative prior. Therefore, GMs and Bayesian inference can be ideally combined to perform estimation tasks due to the distribution modeling abilities of the GMs.

A well-known GM that can be used for directly approximating the CME is the GMM [12]. However, connections between the CME and other GMs are yet to be discovered in the literature. Exemplarily, the GAN-based estimator from [13]–[15] is used in a CS-fashioned way and the score-based approach from [16] requires an iterative posterior sampling process, causing a massive computational complexity. A closely related GM to the GMM is the VAE. Both GMs maximize a lower bound to the data log-likelihood and introduce an artificial latent space. Nevertheless, the GMM utilizes a discrete latent space, which limits its expressiveness. On the contrary, a VAE uses a continuous latent space, resulting in a better representation ability and a more flexible architectural design. The VAE intrinsically makes no assumption about the data distribution and was shown to work well in domains where it is traditionally challenging to derive statistical data models, e.g., in image processing [19].

In this work, we propose a VAE-parameterized estimator, combining a GM and classical estimation theory, with the following *contributions*:

- We model the analytically intractable data distribution as conditionally Gaussian (CG) with the help of the VAE, yielding conditional first and second moments to parameterize conditional linear minimum mean squared error (LMMSE) estimators given the latent representation and noisy observations. The conditional LMMSE estimators are mean squared error (MSE)-optimal and analytically tractable in closed-form due to the CG likelihood model.

This work is funded by the Bavarian Ministry of Economic Affairs, Regional Development, and Energy within the project 6G Future Lab Bavaria. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany, project ID: 16KISK002.

- Since the VAE inherently makes no assumptions about the data distribution, the proposed estimation framework works independently of the adopted data distribution.
- We introduce a low-complexity estimator version based on a maximum a posteriori (MAP) estimate requiring only one neural network (NN) forward pass (MAP-VAE estimator). In contrast to many existing GM-based estimation frameworks, e.g. [13]–[16], this procedure allows for a computationally efficient approximation of the CME, which is high-performing and robust, as a consequence of the VAE serving as generative prior.
- Three estimator variants are proposed, differing in the availability of ground-truth data during their training and estimation phases. The *VAE-real* variant is particularly noteworthy as it requires no access to samples of ground-truth data in either the training or estimation phases.
- We rigorously derive a bound on the performance gap between the MAP-VAE estimator and the CME, allowing for an interpretable estimation procedure. The bound connects the training objective of the VAE with the resulting estimation performance and reveals that the proposed estimator entails a bias-variance tradeoff that is well-known in the estimation literature.
- As an application example, we consider CE, offering a low-complexity implementation due to the structural properties of the CE problem. Our extensive numerical simulations first validate the theoretical analysis and then underline the superiority of the proposed VAE-based estimator variants compared to the baseline methods under various system configurations.

Moreover, we provide the following extensions in this manuscript compared to the preliminary results in [20]. The analyses in Sections III-A and III-D enhance the theoretical foundation of the VAE-based estimator's MSE-optimality. We provide a more general treatment by providing a scheme for linear inverse problems of which multiple-input multiple-output (MIMO)-CE is a special instance. We make the training of the VAE signal-to-noise ratio (SNR)-independent, meaning that we use a single trained VAE for every SNR value, in opposition to [20], where an individual VAE is trained for every SNR value. Finally, the numerical simulations in this manuscript are more comprehensive.

The structure of this manuscript is as follows. Section II discusses the signal model and the general problem formulation and provides background information about the VAE. In Section III, we introduce the VAE-based estimator and its three variants, followed by the derivation and interpretation of the error bound between the proposed estimator and the CME. We discuss CE as an application example in Section IV. Numerical simulation results are presented in Section V, and we conclude this manuscript in Section VI.

Notation: We denote vectors and matrices as lower-case and upper-case bold-faced symbols, respectively. Element-wise multiplication is denoted as $\odot$, the all-zeros vector as $\mathbf{0}$, and the all-ones vector as $\mathbf{1}$. The vectorization operation $\mathrm{vec}(\boldsymbol{G}) \in \mathbb{C}^{g_1 g_2}$ stacks the columns of $\boldsymbol{G} \in \mathbb{C}^{g_1 \times g_2}$ into a vector. The Kronecker product of two matrices $\boldsymbol{B} \in \mathbb{C}^{b_1 \times b_2}$ and $\boldsymbol{D} \in \mathbb{C}^{d_1 \times d_2}$ is $(\boldsymbol{B} \otimes \boldsymbol{D}) \in \mathbb{C}^{b_1 d_1 \times b_2 d_2}$.

II. Preliminaries

A. Signal Model and Problem Formulation

We consider the generic linear inverse problem

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{h} + \boldsymbol{n} \tag{1}$$

with the observation matrix $\boldsymbol{A} \in \mathbb{C}^{M \times N}$ and additive noise $\boldsymbol{n} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \boldsymbol{\Sigma})$. It is assumed that the matrix $\boldsymbol{A}$ and the noise covariance $\boldsymbol{\Sigma}$ are given. The task is to recover $\boldsymbol{h}$ based on $\boldsymbol{y}$. The design of $\boldsymbol{A}$ is characteristic of the problem to be solved, e.g., in CE, $\boldsymbol{A}$ represents the pilot allocation [12]–[15]. For further examples, we refer to [21].

For the solution of (1), we aim to estimate $\boldsymbol{h}$ based on the noisy observation $\boldsymbol{y}$. In the Bayesian framework, $\boldsymbol{h}$ is a RV with an unknown prior $p(\boldsymbol{h})$. The goal is to minimize the MSE

$$\mathrm{E}\left[\|\boldsymbol{h} - \hat{\boldsymbol{h}}\|^2\right] = \int \left[\int \|\boldsymbol{h} - \hat{\boldsymbol{h}}\|^2 p(\boldsymbol{h} \,|\, \boldsymbol{y}) \,\mathrm{d}\boldsymbol{h}\right] p(\boldsymbol{y}) \,\mathrm{d}\boldsymbol{y} \tag{2}$$

with the estimate $\hat{\boldsymbol{h}} \in \mathbb{C}^N$. For minimizing the MSE, minimizing the inner integral is sufficient due to $p(\boldsymbol{y}) \geq 0$. The minimizer is the well-known CME

$$\mathrm{E}[\boldsymbol{h} \,|\, \boldsymbol{y}] = \arg\min_{\hat{\boldsymbol{h}}} \mathrm{E}\left[\|\boldsymbol{h} - \hat{\boldsymbol{h}}\|^2\right] \tag{3}$$

resulting in MSE-optimal estimates, cf. [18, Ch. 10] for details. More generally, the CME is the optimal predictor for all Bregman loss functions, of which the MSE is a special case [22]. Application of Bayes' rule to $p(\boldsymbol{h} \,|\, \boldsymbol{y})$ yields

$$\mathrm{E}[\boldsymbol{h} \,|\, \boldsymbol{y}] = \int \boldsymbol{h} \frac{p_{\boldsymbol{n}}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h})\, p(\boldsymbol{h})}{p(\boldsymbol{y})} \mathrm{d}\boldsymbol{h}. \tag{4}$$

Note that $p_{\boldsymbol{n}}$ represents the noise probability density function (PDF). By inspection of (4), it becomes clear why the CME is difficult to compute. First, it requires access to the unknown and difficult-to-determine prior $p(\boldsymbol{h})$, necessitating an estimate of $p(\boldsymbol{h})$. Second, an approximation of the integral in (4) is required since, in general, there exists no closed-form solution.

B. VAE Fundamentals

In a parametric approach, the parameterized likelihood model $p_{\boldsymbol{\theta}}(\boldsymbol{h})$ approximates the unknown prior $p(\boldsymbol{h})$. One of the simplest parametric models is assuming a Gaussian prior, parameterized with the sample mean and covariance. The resulting parameterized CME approximation is the well-known LMMSE estimator [18, Ch. 10]. However, assuming a Gaussian prior is restrictive, causing the estimator to perform weakly if the true prior strongly deviates from a Gaussian distribution, which is the case in real-world systems. A way to significantly improve the expressiveness of the likelihood model while preserving the favorable properties of a Gaussian distribution is to let it hold only conditionally so the data is modeled as CG. The CG likelihood model has the form

$$\boldsymbol{h} \mid \boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{h} \,|\, \boldsymbol{z}) = \mathcal{N}_{\mathbb{C}}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}), \boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})) \tag{5}$$

with the so-called latent vector $\boldsymbol{z} \in \mathbb{R}^{N_{\mathrm{L}}}$ such that

$$p_{\boldsymbol{\theta}}(\boldsymbol{h}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{h} \,|\, \boldsymbol{z}) p(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \tag{6}$$

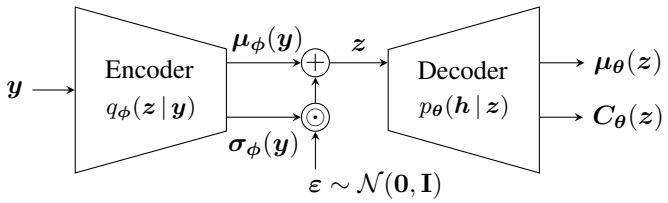

Fig. 1. Structure of a VAE with CG distributions for $q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})$. The encoder and decoder each represent a NN.

with a fixed $p(\boldsymbol{z})$. Besides its great properties in terms of expressiveness, the CG model in (5) will be a key aspect for deriving the VAE-parameterized estimator in Section III-A. Since $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})$ is defined according to (5), $\boldsymbol{\theta}$ also implicitly parameterizes the intractable posterior

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{h}) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})p(\boldsymbol{z})}{\int p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})p(\boldsymbol{z})\mathrm{d}\boldsymbol{z}}. \tag{7}$$

A tractable method to obtain $\boldsymbol{\theta}$ for (5) is the VAE [3], [4], for which an illustration is visible in Fig. 1. For this purpose, $p_{\boldsymbol{\theta}}(\boldsymbol{h})$ is typically decomposed as [23]

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{h}) = \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{h}) + \mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y}) \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{h})) \tag{8}$$

with the evidence lower bound (ELBO)

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{h}) = \mathrm{E}_{q_{\boldsymbol{\phi}}}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})\right] - \mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y}) \,\|\, p(\boldsymbol{z})). \tag{9}$$

and the non-negative Kullback-Leibler (KL) divergence

$$\mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y}) \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{h})) = \mathrm{E}_{q_{\boldsymbol{\phi}}}\left[\log\left(\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{h})}\right)\right]. \tag{10}$$

Note that $\mathrm{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{y})}[\cdot] = \mathrm{E}_{q_{\boldsymbol{\phi}}}[\cdot]$. The variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y})$ is introduced aiming to approximate the intractable $p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{h})$ as can be seen in (10). Consequently, a maximization of the ELBO is independent of (7), maximizes $\log p_{\boldsymbol{\theta}}(\boldsymbol{h})$, as well as minimizes (10). In summary, a sufficiently trained VAE yields $\boldsymbol{\theta}$ for the CG model in (5), as well as an approximation of the intractable posterior in (7) via $q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y})$.

The remaining distributions in (9) are defined as:

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y}), \mathrm{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{y}))), \tag{11}$$

$$p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{12}$$

Moreover, the VAE implements $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y})$ as NNs. With these considerations, let us revisit Fig. 1. The encoder takes an observation $\boldsymbol{y}$ and maps it to $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y})$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{y})$ to obtain a reparameterized sample $\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{y}) \odot \boldsymbol{\varepsilon}$. The sample $\boldsymbol{z}$ is fed into the decoder to obtain $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})$ and $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$ representing the first and second moments of $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})$.

Due to the CG distributions, the terms in the ELBO can be calculated analytically, which is beneficial for the training of the VAE. The expectation term in (9) can be approximated with a single sample $\tilde{\boldsymbol{z}} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y})$, i.e., $(-\mathrm{E}_{q_{\boldsymbol{\phi}}}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})\right])$ is replaced by the estimate

$$\log\det(\pi\,\boldsymbol{C}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}})) + (\boldsymbol{h} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}))^{\mathrm{H}}\boldsymbol{C}^{-1}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}})(\boldsymbol{h} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}})). \tag{13}$$

The KL divergence $\mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y}) \,\|\, p(\boldsymbol{z}))$ in (9) results in

$$\frac{1}{2}\left(\mathbf{1}^{\mathrm{T}}\left(-\log\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{y}) + \boldsymbol{\mu}^2_{\boldsymbol{\phi}}(\boldsymbol{y}) + \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{y})\right) - N_{\mathrm{L}}\right). \tag{14}$$

By utilizing an expressive decoder NN and $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$ parameterization, we assume that a properly trained VAE where (5) holds delivers a $p_{\boldsymbol{\theta}}(\boldsymbol{h})$ that well approximates $p(\boldsymbol{h})$. We will explicitly discuss conditional covariance matrix parameterizations for $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$ in Sections III-B and IV.

III. VAE-Parameterized Estimator

A. MMSE Estimation with the VAE

After its successful training, the VAE yields $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})$ as CG according to (5). The law of total expectation enables reformulating the CME from (4) as [24, Sec. 4.3]:

$$\mathrm{E}[\boldsymbol{h}\,|\,\boldsymbol{y}] = \mathrm{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{y})}[\mathrm{E}_{\boldsymbol{\theta}}[\boldsymbol{h}\,|\,\boldsymbol{z},\boldsymbol{y}]\,|\,\boldsymbol{y}], \tag{15}$$

where the inner expectation is with respect to $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z},\boldsymbol{y})$. We neglect a possible approximation error between $p_{\boldsymbol{\theta}}(\boldsymbol{h})$ and $p(\boldsymbol{h})$ in (15) as it is irrelevant for the estimator derivation. Since $p(\boldsymbol{h})$ is anyway inaccessible, an analysis of such an error is only possible empirically in terms of an MSE investigation, which will be done in Section V. Similar to $p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{h})$, $p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{y})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z},\boldsymbol{y})$ are also implicitly parameterized by $\boldsymbol{\theta}$ due to (7), the fixed prior $p(\boldsymbol{z})$ in (12) and the model in (1). Indeed, $p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{y})$ is generally inaccessible for the same reason as $p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{h})$, cf. (7). Since the encoder receives $\boldsymbol{y}$ as input and $\boldsymbol{h}$ is conditioned on $\boldsymbol{z}$ according to (5), the training objective in (9) enforces $\boldsymbol{n}$ and $\boldsymbol{z}$ to be independent. Then, given (5), we obtain a closed-form expression for the inner expectation in (15) due to the CG property causing $\boldsymbol{y}$ and $\boldsymbol{h}$ to be jointly Gaussian given $\boldsymbol{z}$. Therefore, $\mathrm{E}_{\boldsymbol{\theta}}[\boldsymbol{h}\,|\,\boldsymbol{z},\boldsymbol{y}]$ results in [25]:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}) + \boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})\boldsymbol{A}^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{\Sigma})^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})), \tag{16}$$

where the matrix $\boldsymbol{A}$ and vector $\boldsymbol{y}$ belong to (1), and $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})$, $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$, and $\boldsymbol{z}$ to (5). See Appendix A for a step-by-step derivation of (16).

It remains to solve the intractable outer expectation in (15). To this end, the approximation of $p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{h})$ via $q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y})$ in (10) comes into play. Although (10) shows the approximation of $p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{h})$ instead of $p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{y})$, the parameter combination that maximizes $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})$ in (9) also maximizes $p_{\boldsymbol{\theta}}(\boldsymbol{y}\,|\,\boldsymbol{z})$ since the noise distribution is considered to be known and not subject to optimization, thus permitting the substitution. Consequently, by replacing $p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{y})$ with $q_{\phi}(\boldsymbol{z}\,|\,\boldsymbol{y})$ in (15),

$$\mathrm{E}[\boldsymbol{h}\,|\,\boldsymbol{y}] \approx \mathrm{E}_{q_{\boldsymbol{\phi}}}\left[t_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{y})\right], \quad t_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{y}) = \mathrm{E}_{\boldsymbol{\theta}}[\boldsymbol{h}\,|\,\boldsymbol{z},\boldsymbol{y}]. \tag{17}$$

As we can easily obtain samples of $q_{\phi}(\boldsymbol{z}\,|\,\boldsymbol{y})$ with the help of the encoder, we can approximate $\mathrm{E}[\boldsymbol{h}\,|\,\boldsymbol{y}]$ using samples of the form $\boldsymbol{z}^{(k)} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y}) + \boldsymbol{\varepsilon}^{(k)} \odot \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{y})$ where every $\boldsymbol{\varepsilon}^{(k)}$ is a sample from $\mathcal{N}(\mathbf{0},\mathbf{I}), k = 1,\ldots,K$. Based on the samples $\boldsymbol{z}^{(k)}$ we can approximate the MMSE estimator as a consequence of the law of large numbers [26]:

$$\hat{\boldsymbol{h}}^{(K)}_{\mathrm{VAE}}(\boldsymbol{y}) = \frac{1}{K}\sum_{k=1}^{K} t_{\boldsymbol{\theta}}(\boldsymbol{z}^{(k)},\boldsymbol{y}), \quad \boldsymbol{z}^{(k)} \sim q_{\phi}(\boldsymbol{z}\,|\,\boldsymbol{y}), \tag{18}$$

where $t_{\boldsymbol{\theta}}(\boldsymbol{z}^{(k)},\boldsymbol{y})$ is evaluated with (16).

The estimator $\hat{\boldsymbol{h}}^{(K)}_{\mathrm{VAE}}(\boldsymbol{y})$ generally becomes better for a large number of samples K. However, a large K is unwanted in a real-time system. It is desirable to reduce the complexity of

the estimator as much as possible, which means that K should be low. To this end, we first obtain a MAP estimate for $\boldsymbol{z}$, which is $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y})$ at the encoder output due to the Gaussianity of $q_{\boldsymbol{\phi}}(\boldsymbol{z} \,|\, \boldsymbol{y})$, cf. (11). The MAP estimate is subsequently passed in a single step through the decoder to evaluate $t_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{y})$. Consequently, we define the MAP-VAE estimator

$$\hat{\boldsymbol{h}}_{\text{VAE}}(\boldsymbol{y}) = \hat{\boldsymbol{h}}^{(1)}_{\text{VAE}}(\boldsymbol{y}) = t_{\boldsymbol{\theta}}(\boldsymbol{z}^{(1)} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y}), \boldsymbol{y}) \tag{19}$$

based on the MAP estimate $\boldsymbol{z}^{(1)} = \boldsymbol{\mu}_{\phi}(\boldsymbol{y})$ from $q_{\boldsymbol{\phi}}(\boldsymbol{z} \,|\, \boldsymbol{y})$. In Section III-D, we rigorously analyze the performance gap between the MAP-VAE estimator and the CME. Furthermore, in Section V-A, we compare $\hat{\boldsymbol{h}}^{(K)}_{\text{VAE}}(\boldsymbol{y})$ and $\hat{\boldsymbol{h}}_{\text{VAE}}(\boldsymbol{y})$ for different K, demonstrating that their estimation quality is nearly identical. Unless otherwise stated, we use the MAP-VAE estimator in (19) for the numerical simulations.

B. Covariance Matrix Parameterization

According to (5), the VAE aims to learn a full covariance matrix for $\boldsymbol{h} \,|\, \boldsymbol{z}$. However, learning such a full matrix requires learning a large number of parameters, resulting in huge NNs. It is also known that covariances usually exhibit problem-specific structures, which can be exploited to drastically reduce the number of parameters to learn.

In this work, we consider equidistantly sampled wide-sense stationary (WSS) random processes covering a broad class of applications such as array signal processing [27], speech and audio processing [28], or CE [12], [29]. As a result, the covariance matrix is Toeplitz structured. The parameterization of a Toeplitz matrix is possible with an oversampled discrete fourier transform (DFT) matrix as demonstrated in [30]–[32]. However, if the covariance matrix dimensions are large, a circulant matrix can asymptotically approximate the Toeplitz covariance matrix [33]. By reasonably assuming that the VAE finds latent conditions that preserve the structural properties of the second moments, we can choose,

$$\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z}) = \boldsymbol{F}_N^{\mathrm{H}} \operatorname{diag}(\boldsymbol{c}_{\boldsymbol{\theta}}(\boldsymbol{z})) \boldsymbol{F}_N, \qquad \boldsymbol{c}_{\boldsymbol{\theta}}(\boldsymbol{z}) \in \mathbb{R}_+^N, \tag{20}$$

parameterizing a circulant matrix, where $\boldsymbol{F}_N \in \mathbb{C}^{N \times N}$ is a DFT matrix. Circulant matrices have the advantage that they allow for a low-complexity and memory-efficient implementation and have already been used in previous work, cf. [29]. This can be seen in (20) since a positive and real-valued vector $\boldsymbol{c}_{\boldsymbol{\theta}}(\boldsymbol{z})$ suffices to parameterize a full covariance matrix. Due to the DFT matrix, (20) is furthermore straightforwardly invertible in $\mathcal{O}(N \log N)$ time (by using FFTs), motivating its usage in the proposed VAE-based estimation framework.

C. Variants of VAE-based Channel Estimators

We present three possible estimator variants that leverage the VAE. All three estimators have in common that the VAEs can be trained offline before application. The estimators differ in their ground-truth data knowledge during the training and evaluation phase. A comprehensive overview of all variants with their losses will be shown in Section IV in Table I.

1) VAE-genie: To determine the full potential of our method, we assume $\boldsymbol{n} = \boldsymbol{0}$ in (1) for the encoder input while (16) is still evaluated with a non-zero noise realization. VAE-genie is supposed to exhibit the best estimation results among all variants because the $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})$ and $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$ at the decoder are inferred with the ground-truth data at the encoder and its latent representation. Although VAE-genie even has the potential to outperform the CME, as the ground-truth data acts as side information, this estimator is not applicable in practice, since it requires ground-truth knowledge during the evaluation phase. Instead, it can be a suitable benchmark result in a scenario where the optimal estimator is unknown and inaccessible. VAE-genie requires ground-truth data knowledge during the training *and* evaluation phase.

2) VAE-noisy: This estimator version directly relates to Fig. 1. The encoder receives the noisy observation $\boldsymbol{y}$ as input with $\boldsymbol{n} \neq \boldsymbol{0}$. VAE-noisy only requires ground-truth data access during the training phase to compute (13) for its loss. During the evaluation phase, the mean $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y})$ is obtained based on the noisy observation $\boldsymbol{y}$ to compute (16), which is the reason for the name of this estimator. We expect that VAE-noisy delivers worse estimation quality than VAE-genie as VAE-genie has ground-truth knowledge in the evaluation phase. VAE-noisy is, in contrast, applicable in practice.

3) VAE-real: Similar to VAE-noisy, this estimator variant also receives $\boldsymbol{y}$ as encoder input. The change compared to VAE-noisy happens at the decoder in Fig. 1 where VAE-real learns first and second moments for $p_{\boldsymbol{\theta}}(\boldsymbol{y} \,|\, \boldsymbol{z})$ instead of $p_{\boldsymbol{\theta}}(\boldsymbol{h} \,|\, \boldsymbol{z})$. However, to efficiently compute $\mathrm{E}_{\boldsymbol{\theta}}[\boldsymbol{h} \,|\, \boldsymbol{z}, \boldsymbol{y}]$ we require a CG model for $\boldsymbol{h}$ and not $\boldsymbol{y}$. As long as $\mathrm{E}[\boldsymbol{n}] = \boldsymbol{0}$, which is the case in (1), the mean of $\boldsymbol{y} \,|\, \boldsymbol{z}$ is $\boldsymbol{A}\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})$. A simple workaround can determine the conditional covariance of $\boldsymbol{y} \,|\, \boldsymbol{z}$. While the VAE decoder continues to output $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$, e.g., according to (20), the matrix $\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{\Sigma}$ is used as covariance for $p_{\boldsymbol{\theta}}(\boldsymbol{y} \,|\, \boldsymbol{z})$. Consequently, in (13), VAE-real replaces $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})$ with $\boldsymbol{A}\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})$ and $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$ with $\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{\Sigma}$ during the training. This way, the decoder forces to substitute only the desired part, the conditional covariance $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$, which is used for the computation of (16). It should be noted that no ground-truth data is needed by VAE-real, neither during training nor during evaluation. VAE-real is the most realistic estimator variant since noisy observations can be utilized to train the VAE. In contrast, access to ground-truth data during the training phase is usually related to a considerable additional effort and may sometimes be impractical.

D. MSE-Optimality and Conditional Bias-Variance Tradeoff

In this section, we provide a theoretical analysis of the introduced MAP-VAE estimator. Before establishing a bound on the difference between the MAP-VAE estimator and the CME, let us denote the decoder NN functions as

$$f_{\boldsymbol{\theta},1} : \mathbb{R}^{N_{\mathrm{L}}} \to \mathbb{C}^N, \boldsymbol{z} \mapsto \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}), \tag{21}$$

$$f_{\boldsymbol{\theta},2} : \mathbb{R}^{N_{\mathrm{L}}} \to \mathcal{C}_+^N, \boldsymbol{z} \mapsto \boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z}), \tag{22}$$

where $\mathcal{C}_+^N$ is the set of all $N \times N$ positive semi-definite (PSD) matrices (we consider the case of a full covariance matrix as this trivially includes all parameterized covariances discussed in Section III-B). In this section, we assume $\boldsymbol{y} = \boldsymbol{h} + \boldsymbol{n}$ to analyze the theoretical properties independent of $\boldsymbol{A}$.

Theorem 1. *Consider a decorrelated observation $\boldsymbol{y} = \boldsymbol{h} + \boldsymbol{n}$ with $\boldsymbol{n} \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{0}, \varsigma^2 \mathbf{I})$ and let* (5) *and* (15) *hold. Further, assume the decoder neural network functions are Lipschitz continuous, i.e., for $i = \{1, 2\}$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^{N_L}$,*

$$\|f_{\boldsymbol{\theta},i}(\boldsymbol{a}) - f_{\boldsymbol{\theta},i}(\boldsymbol{b})\|_2 \leq L_i \|\boldsymbol{a} - \boldsymbol{b}\|_2. \tag{23}$$

Then, the expected Euclidean distance between the CME (15) *and the MAP-VAE estimator* (19) *is upper bounded as*

$$\begin{aligned} \mathrm{E}\Big[\big\|\mathrm{E}[\boldsymbol{h} \,|\, \boldsymbol{y}] - \hat{\boldsymbol{h}}_{\mathrm{VAE}}(\boldsymbol{y})\big\|_2\Big] &\leq (C_1 L_1 + C_2 L_2) \\ &\cdot \left(\sqrt{\mathrm{tr}(\boldsymbol{C}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{y})})} + \sqrt{\mathrm{E}\Big[\big\|\boldsymbol{\mu}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{y})} - \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y})\big\|_2^2\Big]} \right) \end{aligned} \tag{24}$$

with the SNR-dependent factors

$$C_1 = \sqrt{\mathrm{E}\left[\frac{\varsigma^4}{(\xi_{\min}(\boldsymbol{y}) + \varsigma^2)^2}\right]}, \quad C_2 = \sqrt{\frac{N}{\varsigma^2}}. \tag{25}$$

where $\xi_{\min}(\boldsymbol{y})$ is the smallest eigenvalue of $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y}))$.

Proof: See Appendix B.

1) MSE-Optimality: Theorem 1 shows that the distance of the MAP-VAE estimator to the CME only depends on the first two moments of the posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z} \,|\, \boldsymbol{y})$, which is approximated by $q_{\boldsymbol{\phi}}(\boldsymbol{z} \,|\, \boldsymbol{y})$. In particular, the bound is smaller if the first moments of these posterior distributions match, which can reasonably assumed to be the case after successfully training the VAE, being a mild assumption as no restrictions on their higher moments apply.

Moreover, the smaller the variances of $p_{\boldsymbol{\theta}}(\boldsymbol{z} \,|\, \boldsymbol{y})$, the better the CME approximation. Intuitively, this means that the less stochastic a mapping from the observation to the latent space is, the better the MAP-VAE estimator performs. Let us consider the following setup to motivate the encoder variances to become small after training. Assume the input data is compressible onto a lower-dimensional manifold, i.e., a lossless compression mapping exists from $\mathbb{C}^N$ to $\mathbb{R}^{N_L}$. In particular, this is known to be fulfilled for *natural signals*, e.g., images or audio signals, wireless channels (especially in mmWave systems), or, in general, signals that exhibit a sparse representation through a dictionary. Then, a deterministic mapping exists into the latent space that can be learned by the VAE. In other words, there is no necessity for a stochastic mapping, and the variances in (24) can be chosen as zero without performance loss. This holds without restriction for the VAE-genie variant, where the encoder input is noiseless. For the VAE-noisy and VAE-real variants, although the latent encoding is trained to be stochastically independent of the noise, finding a deterministic mapping may be more intricate, especially in the low SNR regime, yielding a possibly higher encoder variance. We elaborate on this hypothesis in more detail for the example of channel estimation in MIMO systems in Section IV and show through simulations in Section V-A that the VAE's encoder variances are indeed converging towards zero during the training process.

Concluding the above discussion, the bound in Theorem 1 establishes a connection between the training of the VAE, purely relying on likelihood maximization, and the resulting MSE performance. Moreover, the impact of the latent dimension on the estimation performance is better interpretable. Thus, by a successful training of a well-designed VAE, the resulting parameterized estimator converges to the CME, thereby achieving a low MSE. We validate this proposition also through numerical results in Section V-A.

2) Conditional Bias-Variance Tradeoff: In addition to the above insights about the connection of the VAE's training and the resulting estimation performance, the constants C_1 and C_2 in (25) have a reciprocal behavior over the SNR and, in particular, are vanishing in the high and low SNR, respectively. That is,

$$\lim_{\varsigma^2 \to 0} C_1 = 0, \qquad \lim_{\varsigma^2 \to \infty} C_2 = 0. \tag{26}$$

Interestingly, this can be interpreted as a *conditional bias-variance tradeoff* since $C_2 L_2$ in (24), addressing the contribution of the conditional covariances, vanishes in low SNR; moreover, $C_1 L_1$, attributed to the conditional means, vanish in high SNR, cf. (54). Thus, the parameterized conditional covariance quality is less critical in the low SNR regime, as the parameterized LMMSE estimator relies more on the conditional first moment and vice versa in the high SNR regime. Consequently, the respective error terms have less impact on the bound to the CME. The entailment of such a *conditional bias-variance tradeoff* is a highly desirable property of the proposed estimator as it serves as a regularization for the estimation performance and allows for great interpretability. Moreover, the analysis holds without restriction for all discussed estimator variants in Section III-B and all parameterized conditional covariance matrices since we made no assumptions about their structural properties.

IV. Example Application: Channel Estimation

In this work, we consider MIMO CE as an application example. In a MIMO communications system, the transmitter with N_{tx} antennas sends N_{p} pilots to the receiver with N_{rx} antennas for estimating the channel matrix $\boldsymbol{H} \in \mathbb{C}^{N_{\mathrm{rx}} \times N_{\mathrm{tx}}}$. More precisely, the noisy observations

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{N} \in \mathbb{C}^{N_{\mathrm{rx}} \times N_{\mathrm{p}}} \tag{27}$$

are obtained at the receiver with the pilot matrix $\boldsymbol{X} \in \mathbb{C}^{N_{\mathrm{tx}} \times N_{\mathrm{p}}}$ and noise matrix $\boldsymbol{N}$. After vectorizing (27), the relation to (1) becomes apparent. Consequently, $\boldsymbol{y} = \mathrm{vec}(\boldsymbol{Y})$, $\boldsymbol{h} = \mathrm{vec}(\boldsymbol{H})$, $\boldsymbol{A} = (\boldsymbol{X}^{\mathrm{T}} \otimes \mathbf{I})$, and $\boldsymbol{n} = \mathrm{vec}(\boldsymbol{N})$. Further, $M = N_{\mathrm{rx}} N_{\mathrm{p}}$ and $N = N_{\mathrm{rx}} N_{\mathrm{tx}}$. We investigate the uplink of a communications system where the mobile terminal (MT) transmits to the base station (BS) with $N_{\mathrm{tx}} < N_{\mathrm{rx}}$. However, the proposed framework can also be applied to the downlink since $\boldsymbol{A}$ has a comparable structure.

We assume that the BS and MT are both equipped with a uniform linear array (ULA) with half-wavelength spacing. Note that a different array structure or antenna spacing can be straightforwardly reflected by the VAE's parameterized covariance at the decoder output. Furthermore, we consider the fully determined case of (27), i.e., $N_{\mathrm{p}} = N_{\mathrm{tx}}$. We utilize DFT pilots, resulting in a unitary $\boldsymbol{X}$, which results in a unitary $\boldsymbol{A}$. Moreover, we assume $\boldsymbol{\Sigma} = \varsigma^2 \mathbf{I}$ with given ς^2. Therefore, we

perform a least squares (LS) estimate of (1) to interpret it as a denoising task relating directly to the theoretical analysis in the previous section. The underdetermined case involving a wide $\boldsymbol{A}$ is investigated in [31] and the uniform rectangular array (URA) case at the BS in [32] covering more advanced setups. The works [31], [32] demonstrate a superior performance of the VAE-based estimators, highlighting the framework's versatile applicability under various system configurations.

Due to the uniform antenna spacing at a ULA, the covariance matrix at the BS or MT side becomes Toeplitz structured, which are approximated by circulant matrices as explained in Section III-B. Under the uncorrelated scattering assumption, we can model the channel covariance matrix (CCM) as the Kronecker product of two circulant matrices, representing the BS and MT side covariances. More precisely, let

$$\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z}) = \boldsymbol{Q}^{\mathrm{H}} \operatorname{diag}(\boldsymbol{c}_{\boldsymbol{\theta}}(\boldsymbol{z}))\boldsymbol{Q}, \quad \boldsymbol{c}_{\boldsymbol{\theta}}(\boldsymbol{z}) \in \mathbb{R}_{+}^{N} \tag{28}$$

be the CCM parameterization, where $\boldsymbol{Q} = (\boldsymbol{F}_{N_{\mathrm{tx}}} \otimes \boldsymbol{F}_{N_{\mathrm{rx}}})$. In (28), $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$ is a block-circulant matrix, possessing the same favorable attributes regarding memory efficiency and low-complexity as an ordinary circulant matrix. For a single-input multiple-output (SIMO) system, which implies $N_{\mathrm{tx}} = 1$, (28) simplifies to (20).

A. Training Loss and Network Architecture

In principle, we train a VAE with the loss in (9), and, after the training, perform CE as described in Section III-A. Indeed, we can simplify (9) as a result of the circulant parameterization. Let $\boldsymbol{h}_Q = \boldsymbol{Q}\boldsymbol{h}$, then the negative decoder likelihood in (13) can be expressed as

$$N \log \pi + \mathbf{1}^{\mathrm{T}} \left(\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) \odot |\boldsymbol{h}_Q - \boldsymbol{Q}\boldsymbol{\mu}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}})|^2 - \log \boldsymbol{\lambda}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) \right) \tag{29}$$

with $\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\boldsymbol{z}) = \boldsymbol{c}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})$ and the element-wise absolute value $|\cdot|$. Eq. (13) reduces the numerical complexity during the training process because it avoids the inversion of a full covariance matrix compared to (29). What is more, we utilize the LS estimate of (1) (or $\boldsymbol{h}$ for VAE-genie) multiplied with $\boldsymbol{Q}$ as encoder input. Thus, the encoder input is effectively transformed into the angular or beamspace domain [34, Sec. 7.3], which is known to be sparse or highly compressible in massive MIMO systems, especially in mmWave systems. This validates the hypothesis of having a deterministic compression mapping that can be learned through the encoder in Section III-D. Therefore, a performance of the MAP-VAE estimator close to the CME can be expected, which is also seen later in Section V.

Combining every aspect from this section, the reformulated training objective that VAE-noisy is supposed to minimize is:

$$\begin{aligned} \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}} = \mathbf{1}^{\mathrm{T}} \big[& \boldsymbol{\lambda}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) \odot |\boldsymbol{h}_Q - \boldsymbol{Q}\boldsymbol{\mu}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}})|^2 - \log \boldsymbol{\lambda}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) \\ & - \log \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{y}) + 0.5(\boldsymbol{\mu}_{\boldsymbol{\phi}}^2(\boldsymbol{y}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\boldsymbol{y})) \big]. \end{aligned} \tag{30}$$

The argument and constants are omitted for brevity and $\tilde{\boldsymbol{z}}$ is a sample from $q_{\boldsymbol{\phi}}(\boldsymbol{z} \,|\, \boldsymbol{y})$. Since VAE-genie has the ground-truth channel as encoder input, the training loss for this variant replaces $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{y})$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{y})$ with $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{h})$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{h})$, respectively. Apart from that, the training loss is identical to (30). For the training of VAE-real, in (30), $\boldsymbol{h}_Q$ is replaced with $\boldsymbol{y}_Q = \boldsymbol{Q}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{y}$, and $\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\boldsymbol{z})$ with $(\boldsymbol{c}_{\boldsymbol{\theta}}(\boldsymbol{z}) + \varsigma^2\mathbf{1})^{-1}$. Table I summarizes the proposed estimator variants with an overview regarding the respective encoder input and training loss. In each case, the training loss refers to a single batch element.

We briefly describe our VAE implementation at this point. The simulation code with the corresponding architectures is also publicly available.[1] Fig. 2 illustrates the VAE implementation. The colorings of the arrows in Fig. 2 symbolize different standard layers. On the left, it is visible that the real and imaginary parts of the encoder input are stacked as convolutional channels (CCs). As a first block, the purple arrow ⇨ represents a 1x1 convolutional layer (CL) that maps to a higher number of CCs, which is different for every system configuration. Subsequently, three orange arrows ⇨ follow, representing a block of a CL, a batch normalization (BN) layer, and a ReLU activation function. In each CL, the CC amount at the output is multiplied by a factor of 1.75. After a reshaping layer (RL) and linear layer (LL), symbolized by the gray arrow ⇨ and green arrow ⇨, respectively, we arrive at the latent space. The reparameterized sample $\boldsymbol{z}$ is fed into the decoder, which is a symmetrically flipped version of the encoder. The red arrows ⇨ symbolize blocks of a transposed CL, a BN layer, and a ReLU activation function. At the output, we have a sample with three CCs that is fed into an RL and LL to produce the decoder outputs. We use exponential functions to enforce strictly positive values as it is required for $\boldsymbol{\sigma}_{\boldsymbol{\phi}}$ and $\boldsymbol{c}_{\boldsymbol{\theta}}$.

The number of CCs, kernel size, and latent dimension are different for every system configuration and are found by a random search over the hyperparameter space by searching for the combination that yields the highest value for (29) [35]. We perform the random search with the help of the *Tune* package [36]. We use 2D CLs in the MIMO case and 1D CLs in the SIMO case. A batch size of 128, a learning rate of $7 \cdot 10^{-4}$ in combination with Adam [23], and a stride of two in the CLs are used. We implement the NNs with PyTorch and refer the reader to the simulation code for further details. Additionally, we experimented with BN and its variants to determine how we can achieve the best performance [37]–[39]. We achieve the best performance with BN as is proposed in [37]. The only important point is to consider a large enough batch size to limit the variance of the stochastic gradient. We additionally use the method of free bits during the training as described in [23].

B. Computational Complexity

In this section, we discuss the computational complexity of the proposed estimator. The procedure to determine $\hat{\boldsymbol{h}}_{\mathrm{VAE}}(\boldsymbol{y})$ can be split into two parts. The first step is a forward pass through the VAE to acquire $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})$ and $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$. The second step is the evaluation of $t_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{y})$ in (16) with given $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})$ and $\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})$. The computational complexity of the first step is tied to the VAE architecture in Fig. 2. Since all layers exhibit a different complexity, we need a complexity bound for which two aspects are relevant. First, a CL requires $\mathcal{O}(RN)$ time, with R being the product of the number of parameters divided by the stride in the CL. Second, the final LL requires $\mathcal{O}(N^2)$ time. The remaining layers exhibit less complexity than the

[1] https://github.com/tum-msv/vae-estimator.

TABLE I
OVERVIEW OF THE PROPOSED VAE-BASED ESTIMATOR VARIANTS.

variant	VAE encoder input	training loss $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}$ (relates to one batch element)
VAE-genie	ground-truth channel	$\mathbf{1}^{\mathrm{T}}\left[\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) \odot \lvert\boldsymbol{h}_Q - \boldsymbol{Q}\boldsymbol{\mu}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}})\rvert^2 - \log \boldsymbol{\lambda}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) - \log \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{h}) + 0.5\,(\boldsymbol{\mu}_{\boldsymbol{\phi}}^2(\boldsymbol{h}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\boldsymbol{h}))\right]$
VAE-noisy	noisy observation	$\mathbf{1}^{\mathrm{T}}\left[\boldsymbol{\lambda}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) \odot \lvert\boldsymbol{h}_Q - \boldsymbol{Q}\boldsymbol{\mu}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}})\rvert^2 - \log \boldsymbol{\lambda}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) - \log \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{y}) + 0.5\,(\boldsymbol{\mu}_{\boldsymbol{\phi}}^2(\boldsymbol{y}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\boldsymbol{y}))\right]$
VAE-real	noisy observation	$\mathbf{1}^{\mathrm{T}}\left[(\boldsymbol{c}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) + \varsigma^2\mathbf{1})^{-1} \odot \lvert\boldsymbol{y}_Q - \boldsymbol{Q}\boldsymbol{\mu}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}})\rvert^2 + \log(\boldsymbol{c}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}) + \varsigma^2\mathbf{1}) - \log \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{y}) + 0.5\,(\boldsymbol{\mu}_{\boldsymbol{\phi}}^2(\boldsymbol{y}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\boldsymbol{y}))\right]$

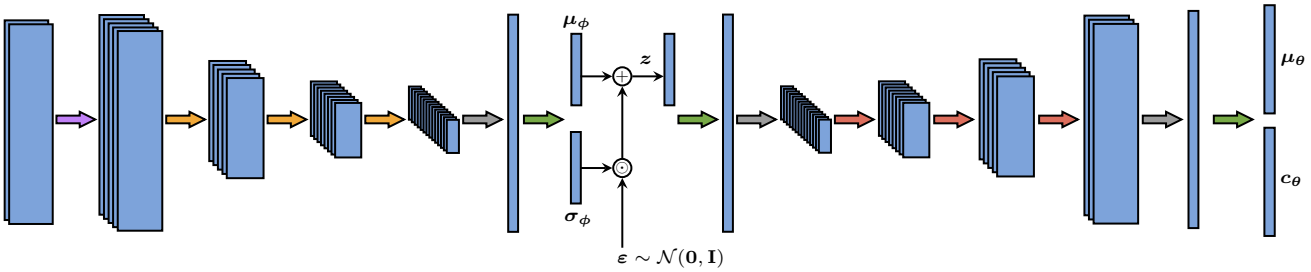

Fig. 2. Detailed illustration of the different layers constituting our VAE implementation. The real and imaginary parts of the input are stacked as CCs and processed. Each colored arrow represents a different standard layer. Purple ⇨ stands for a 1x1 CL, orange ⇨ for a block of a CL, BN layer, and ReLU activation function, gray ⇨ for a RL, green ⇨ for a LL, and red ⇨ for a block of a transposed CL, BN layer, and ReLU activation function.

CLs and final LL. Although R should be increased if N grows, R arguably does not show more than linear growth in N. In conclusion, utilizing $\mathcal{O}(N^2)$ as complexity bound per layer is reasonable. For the D layers of the VAE forward pass, this makes an overall complexity of $\mathcal{O}(DN^2)$.

We come to the second step of obtaining $\hat{\boldsymbol{h}}_{\mathrm{VAE}}(\boldsymbol{y})$, which is the evaluation of $t_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{y})$. In principle, the inversion of $\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})\boldsymbol{A}^{\mathrm{H}} + \varsigma^2\mathbf{I}$ dominates the complexity. Let us inspect $\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})\boldsymbol{A}^{\mathrm{H}} = \boldsymbol{A}\,\boldsymbol{Q}^{\mathrm{H}}\operatorname{diag}(\boldsymbol{c}_{\boldsymbol{\theta}}(\boldsymbol{z}))\boldsymbol{Q}\,\boldsymbol{A}^{\mathrm{H}}$ in more detail. If we assume to have unitary pilots and set $\tilde{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{Q}^{\mathrm{H}}$ we can show that $\tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^{\mathrm{H}} = \mathbf{I}$ and $\tilde{\boldsymbol{A}}^{\mathrm{H}}\tilde{\boldsymbol{A}} = \mathbf{I}$ holds, so $\tilde{\boldsymbol{A}}$ is unitary. Hence, the inverse of $\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})\boldsymbol{A}^{\mathrm{H}}$ is $\tilde{\boldsymbol{A}}\operatorname{diag}(\boldsymbol{c}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))\tilde{\boldsymbol{A}}^{\mathrm{H}}$. We can therefore simplify the estimate $t_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{y})$ as in (16) to

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}) + \boldsymbol{Q}^{\mathrm{H}}\operatorname{diag}(\mathbf{1} + \boldsymbol{c}_{\boldsymbol{\theta}}(\boldsymbol{z}) \odot \varsigma^{-2}\mathbf{1})\boldsymbol{Q}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z})) \quad (31)$$

whose complexity is $\mathcal{O}(N \log N)$ due to multiplying with $\boldsymbol{Q}$. As can be seen from our elaborations above, the evaluation of the VAE requires $\mathcal{O}(DN^2)$ time, which outweighs the evaluation time of $\mathcal{O}(N \log N)$ for (31). Additionally, many potentials exist to reduce the VAE complexity, e.g., with pruning [40]. Moreover, the computations in the VAE are highly parallelizable due to the CLs, which mitigates the $\mathcal{O}(DN^2)$ complexity.

C. Channel Models

We consider different channel models in this work to validate the proposed methods. The 3rd Generation Partnership Project (3GPP) defines an urban macrocell spatial channel model which computes the transmit-side CCM as [41]:

$$\boldsymbol{C}_{\boldsymbol{\delta},\mathrm{tx}} = \int_{-\pi}^{\pi} g_{\mathrm{tx}}(\vartheta;\boldsymbol{\delta})\boldsymbol{a}_{\mathrm{tx}}(\vartheta)\boldsymbol{a}_{\mathrm{tx}}(\vartheta)^{\mathrm{H}}\mathrm{d}\vartheta. \quad (32)$$

The vector $\boldsymbol{a}_{\mathrm{tx}}(\vartheta)$ denotes the transmitter array steering vector, which is $[1, \exp(\mathrm{j}\pi\sin(\vartheta)), \ldots, \exp(\mathrm{j}\pi(N_{\mathrm{tx}}-1)\sin(\vartheta))]^{\mathrm{H}}$ in the case of a ULA. Analogously, the receive-side CCM $\boldsymbol{C}_{\boldsymbol{\delta},\mathrm{rx}}$ is obtained. The function $g_{\mathrm{tx}}(\cdot;\boldsymbol{\delta})$ describes an angular power spectrum and is parameterized by the vector $\boldsymbol{\delta}$, which follows a prior distribution $p(\boldsymbol{\delta})$ that accounts for the involved path gains and angles of the propagation cluster. More precisely, $g_{\mathrm{tx}}(\cdot;\boldsymbol{\delta})$ is a mixture of Laplace densities whose standard deviations represent the angular spreads, cf. [29] for more details. The CCM for $\boldsymbol{h}$ in (1) is determined as $\boldsymbol{C}_{\boldsymbol{\delta}} = (\boldsymbol{C}_{\boldsymbol{\delta},\mathrm{tx}} \otimes \boldsymbol{C}_{\boldsymbol{\delta},\mathrm{rx}})$, under the assumption of uncorrelated scattering [42]. Once $\boldsymbol{C}_{\boldsymbol{\delta}}$ is constructed, a channel realization can be obtained according to $\boldsymbol{h} \mid \boldsymbol{\delta} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \boldsymbol{C}_{\boldsymbol{\delta}})$ [29]. This way, a correlated Rayleigh fading model is enforced that only holds conditionally, meaning that every channel is individually associated with a different set of path gains and angles contained in $\boldsymbol{\delta}$ representing different propagation clusters. Note that $\boldsymbol{C}_{\boldsymbol{\delta}}$ is different for every channel realization, causing $p(\boldsymbol{h})$ to be non-Gaussian.

The QuaDRiGa channel simulator allows for the simulation of realistic channels with spatial consistency and time evolution [43], [44]. MIMO channel matrices are modeled as a superposition of in total L propagation paths such that $\boldsymbol{H} = \sum_{\ell=1}^{L} \boldsymbol{G}_\ell \exp(-2\pi\mathrm{j} f_c \tau_\ell)$ where the carrier frequency is denoted as f_c and the delay of the ℓ-th path as τ_ℓ. The entries of the matrix $\boldsymbol{G}_\ell$ represent the complex-valued gain between every antenna pair caused by the path loss, antenna radiation pattern, and polarization. We use version 2.6.1 of QuaDRiGa to simulate channels at a frequency of 6 GHz in an urban macrocell scenario. We place the BS at a height of 25 m, and it covers a sector of 120 °. Twenty percent of the MTs are outdoors 1.5 m above the ground at a distance between 35 and 500 m from the BS. The remaining eighty percent are situated indoors at different floor levels. We consider a line of sight (LOS) propagation environment, where $L = 37$. We equip the BS with "3GPP-3D" antennas and the MTs with omnidirectional antennas. After generation, the channels are post-processed to normalize the path gain. Compared to the 3GPP channel model, which is fully stochastic, the QuaDRiGa simulator is of a geometric nature. QuaDRiGa determines channel realizations by a geometric simulation in a randomized

and approximately realistic BS environment. The QuaDRiGa model enables us to highlight that the proposed framework works independently of the adopted channel model.

D. Related Channel Estimators

This section presents related channel estimators as baselines for the numerical simulations in Section V-B. In the case of the 3GPP channel model from Section IV-C, we have access to the true CCM $\boldsymbol{C}_{\boldsymbol{\delta}}$. This allows us to evaluate a genie covariance-based estimator (genie-cov) [29], which is given by the LMMSE formula

$$\hat{\boldsymbol{h}}_{\text{genie-cov}}(\boldsymbol{y}) = \boldsymbol{C}_{\boldsymbol{\delta}}\boldsymbol{A}^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{\delta}}\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{\Sigma})^{-1}\boldsymbol{y}. \tag{33}$$

This estimator uses utopian genie knowledge to acquire $\boldsymbol{C}_{\boldsymbol{\delta}}$.

A practical estimator can be based on the sample covariance matrix $\hat{\boldsymbol{C}} = \frac{1}{T_{\mathrm{r}}}\sum_{i=1}^{T_{\mathrm{r}}} \boldsymbol{h}_i\boldsymbol{h}_i^{\mathrm{H}}$ for T_{r} samples in the training dataset. The corresponding estimator reads as

$$\hat{\boldsymbol{h}}_{\text{global-cov}}(\boldsymbol{y}) = \hat{\boldsymbol{C}}\boldsymbol{A}^{\mathrm{H}}(\boldsymbol{A}\hat{\boldsymbol{C}}\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{\Sigma})^{-1}\boldsymbol{y}. \tag{34}$$

LS estimation is another comparison method we investigate in our simulations. An LS estimate can be obtained as $\hat{\boldsymbol{h}}_{\mathrm{LS}}(\boldsymbol{y}) = \boldsymbol{A}^{\mathrm{H}}\boldsymbol{y}$.

CS-based CE techniques are another prominent topic in the literature. Especially regarding millimeter waves, CS algorithms are potentially interesting candidates [45]. This work compares the proposed estimators with the approximate message passing (AMP) algorithm [46], [47]. As a dictionary for AMP, we use a two times oversampled DFT matrix.

We also want to compare the proposed estimators with current ML-based channel estimators. A recently proposed method exploits structural information of the MMSE estimator to design a neural network-based estimator for the SIMO signal model [29]. The derivation leads to a convolutional neural network with ReLU activation function, so we call this estimator CNN. The extension of [29] to the MIMO case is proposed in [48], to which we also refer in our simulations.

The last comparison method in this section, also recently proposed, is based on a GMM [12]. The idea is to fit a GMM to the underlying channel distribution and parameterize a channel estimator with the help of the GMM, representing an estimator based on a generative prior. We fit a GMM with 128 mixture components for all simulations and a restriction on the fitted covariances such that they are block-circulant.

V. Simulation Results

This section presents the CE results based on numerical simulations. We create 200,000 channel realizations for every system configuration in the upcoming section representing a randomly sampled realistic BS environment. The channels are divided into $T_{\mathrm{r}} = 180{,}000$ training, $T_{\mathrm{v}} = 10{,}000$ validation, and $T_{\mathrm{e}} = 10{,}000$ test samples. The channels are normalized such that $\mathrm{E}[\|\boldsymbol{h}\|^2] = N$. In our experiments, we calculate the normalized mean squared error (NMSE) as $\frac{1}{T_{\mathrm{e}}N}\sum_{i=1}^{T_{\mathrm{e}}} \|\boldsymbol{h}_i - \hat{\boldsymbol{h}}_i\|^2$ for the test dataset, where we denote the i-th channel realization and corresponding estimate as $\boldsymbol{h}_i$ and $\hat{\boldsymbol{h}}_i$, respectively. Accordingly, we define the SNR as $N_{\mathrm{tx}}/\varsigma^2$. We train the VAEs for a range of SNR values between

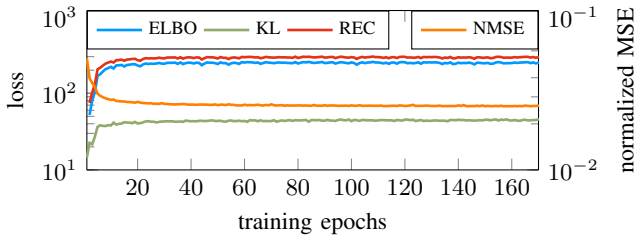

Fig. 3. Training of the VAE-noisy variant for the 3GPP channel model (SIMO case) with three propagation clusters at an SNR of 10 dB. ELBO refers to the complete training loss in (30), REC to the negative of (29), and KL to (14). REC is plotted including the in (29) omitted constants.

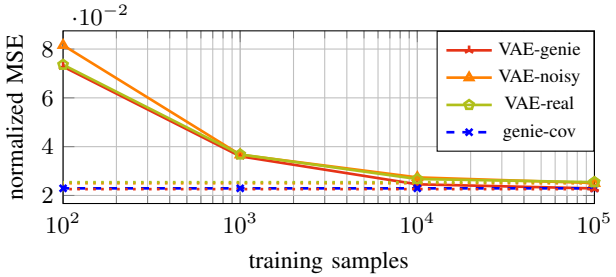

Fig. 4. Normalized MSE for different numbers of training samples at an SNR of 10 dB for the 3GPP channel model (SIMO case) with three propagation clusters and 128 antennas at the receiver. The dotted lines display the achieved result with the complete training dataset of 180,000 samples.

-19 and 39 dB. The proposed estimators are, therefore, SNR-independent. During the training of VAE-noisy and VAE-real, we sample new realizations $\boldsymbol{n}$ after every epoch. We train the VAEs until (29) does not improve for 100 consecutive epochs on the validation dataset. If not stated otherwise, $N_{\mathrm{L}} = 16$ for one propagation cluster and $N_{\mathrm{L}} = 32$ in all other cases.

A. Numerical Convergence Analysis

At first, we illustrate the training progress of the VAE-noisy variant for the 3GPP SIMO signal model with three propagation clusters at an SNR of 10 dB in Fig. 3. The VAE-genie and VAE-real variants exhibit a similar behavior, so we only display VAE-noisy here. It can be observed that most of the training progress happens in the first 20 epochs. Interestingly, an increase of the REC term from (29) coincides with a decrease of the NMSE, which indicates that the VAE learns to properly model the data. Moreover, this validates the theoretical analysis in Theorem 1 that showed a smaller gap to the CME and, thus, a lower NMSE for a VAE that better matches the first moments of the posterior distributions, which is achieved during a successful training.

Further, we investigate two critical quantities of the model selection process: the training dataset's size and the latent space's dimensionality. Regarding the size of the training dataset, a larger size is likely to lead to better estimation results. Fig. 4 provides insights into this matter. We display the estimation results of the test dataset for the three proposed variants of VAE-based estimators depending on the size of the training dataset. The 3GPP channel model (SIMO case) with three propagation clusters and 128 antennas at the receiver is used in Fig. 4. As dotted lines, we display the attained estimation result for the complete training dataset of 180,000

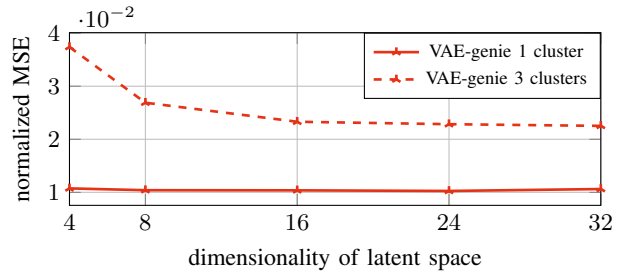

Fig. 5. Normalized MSE for different sizes of the latent space at an SNR of 10 dB for the 3GPP channel model (SIMO case) with one or three propagation clusters and 128 antennas at the receiver.

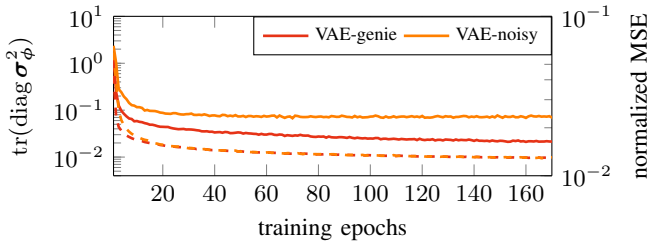

Fig. 6. Trace of encoder variances and NMSE over training epochs on the 3GPP channel model (SIMO case) with 128 antennas, one propagation cluster, 10 dB SNR, and $N_{\mathrm{L}} = 4$. The NMSE curves are displayed as dashed.

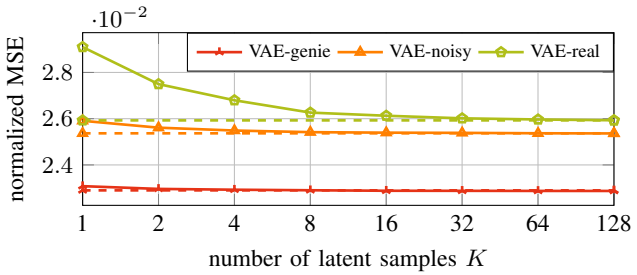

Fig. 7. Normalized MSE for different numbers of samples K drawn in the latent space for the evaluation of $\hat{\boldsymbol{h}}_{\mathrm{VAE}}^{(K)}(\boldsymbol{y})$ at an SNR of 10 dB for the 3GPP channel model (SIMO case) with three propagation clusters and 128 antennas at the receiver. The dashed lines represent the estimate $\hat{\boldsymbol{h}}_{\mathrm{VAE}}(\boldsymbol{y})$ in (19).

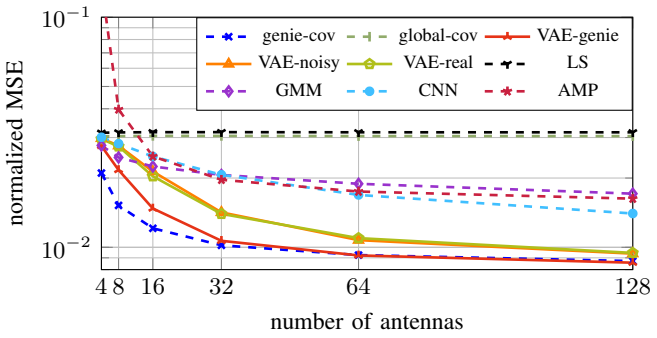

Fig. 8. Normalized MSE for the 3GPP channel model (SIMO case) with three propagation clusters for different numbers of antennas at the receiver at an SNR of 15 dB. The proposed methods are displayed with solid linestyles.

samples. We also show the estimation performance of the genie-cov estimator in blue. It is visible that the most progress is reached from 10^2 to 10^4 training samples. More than 10^4 training samples only lead to minor NMSE improvements for all three types of VAE-based estimators.

The influence of the dimensionality of the latent space on the estimation result is less apparent than the size of the training dataset. We illustrate the NMSE for dimensionalities in the range $[4, 32]$ for the 3GPP SIMO channel model with 128 antennas at the receiver in Fig. 5 by considering one and three propagation clusters. The NMSE is nearly constant for the case with one propagation cluster. In contrast, the NMSE decreases from dimensionality 4 to 16 for the three propagation clusters case and saturates for larger dimensional latent spaces. In practice, the operator must select an ample enough latent space to obtain a desirable performance.

Theorem 1 in Section III-D described that the convergence of the MAP-VAE estimator to the CME depends on the vanishing of $\mathrm{tr}(\boldsymbol{C}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{y})})$. To this end, we analyze the trace of the encoder variance, i.e., the summed variance $\boldsymbol{\sigma}_{\boldsymbol{\phi}}^2$ of $q_{\boldsymbol{\phi}}$, which optimally is a good approximation of $\mathrm{tr}(\boldsymbol{C}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}\,|\,\boldsymbol{y})})$, over the training epochs on the validation dataset in Fig. 6. We evaluate VAE-genie and VAE-noisy with $N_{\mathrm{L}} = 4$ on the 3GPP channel model (SIMO case) with 128 antennas and one propagation cluster at an SNR of 10 dB. It is visible that both encoder variances are decreasing in a comparable way as the NMSE, indicating that lower traces improve the NMSE. The noise variance detrimentally influences VAE-noisy's encoder variance trace since it is always higher than that of VAE-genie. In conclusion, since the wireless channel data is expected to be sparse or compressible in the angular domain, cf. Section IV, the VAE indeed aims to find a less stochastic encoder mapping during training. This is in agreement with the argumentation in Section III-D and the observation of a decreasing NMSE of the parameterized estimator during training in Fig. 3.

As pointed out in Section III-A, we approximate the CME with the MAP-VAE estimator by only forwarding the latent mean vector $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{z})$ to approximate the outer expectation in (15). It is interesting to see the NMSE performance for different numbers of K samples from $q_{\boldsymbol{\phi}}(\boldsymbol{z}\,|\,\boldsymbol{y})$ to compute $\hat{\boldsymbol{h}}_{\mathrm{VAE}}^{(K)}(\boldsymbol{y})$ from (18). Fig. 7 provides such an analysis by showing the NMSE for different numbers of latent samples. As dashed lines, we display the MAP-VAE estimator, which only uses the single sample $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{z})$ for the input of the VAE's decoder. We observe that VAE-real benefits the most from more samples. For VAE-genie and VAE-noisy, there are only slight improvements present. Interestingly, only taking the mean value, representing the MAP-VAE estimator, delivers an estimation performance of about $K = 64$ samples for VAE-real. The excellent performance of the MAP-VAE estimator is a supporting argument for the theoretical analysis of the estimator in Theorem 1 that predicts a small distance from the CME if the VAE is well-trained.

B. Normalized MSE Results

We begin with an NMSE investigation for the 3GPP channel model (SIMO case) with three propagation clusters and varying numbers of antennas at the receiver at an SNR of 15 dB in Fig. 8. The illustration shows that the proposed VAE-based methods need a sufficiently large amount of antennas to

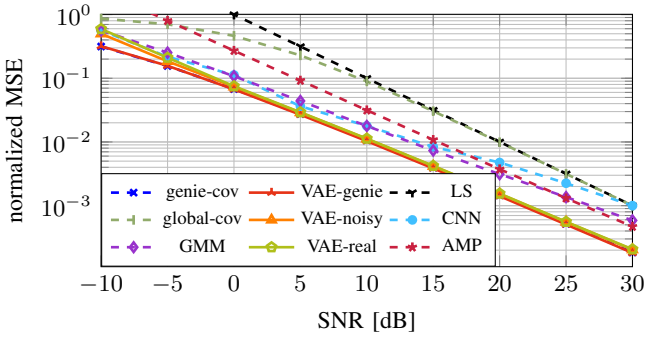

Fig. 9. Normalized MSE for the 3GPP channel model (SIMO case) with one propagation cluster and 128 antennas at the receiver. The proposed methods are displayed with solid linestyles.

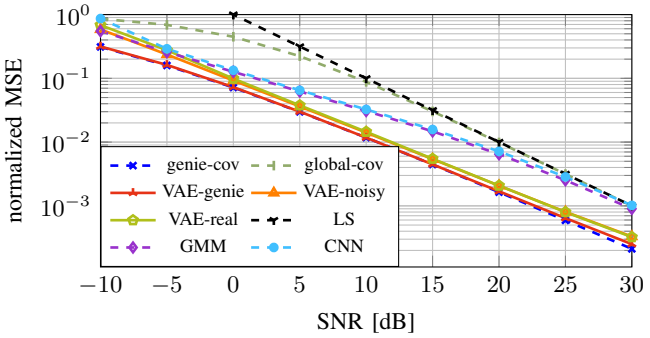

Fig. 11. Normalized MSE for the 3GPP channel model (MIMO case) with one propagation cluster, 32 antennas at the receiver, and 4 antennas at the transmitter. The proposed methods are displayed with solid linestyles.

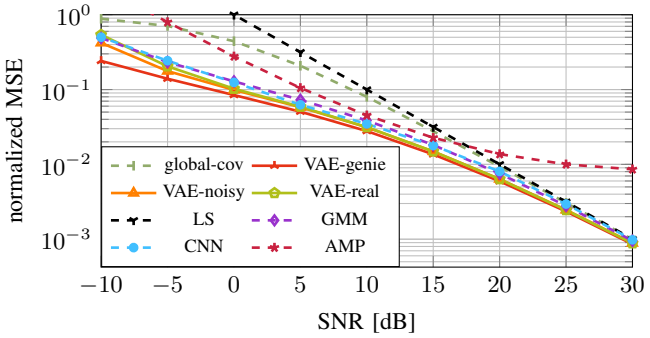

Fig. 10. Normalized MSE for the QuaDRiGa channel model (SIMO case) with LOS channels and 128 antennas at the receiver. The proposed methods are displayed with solid linestyles.

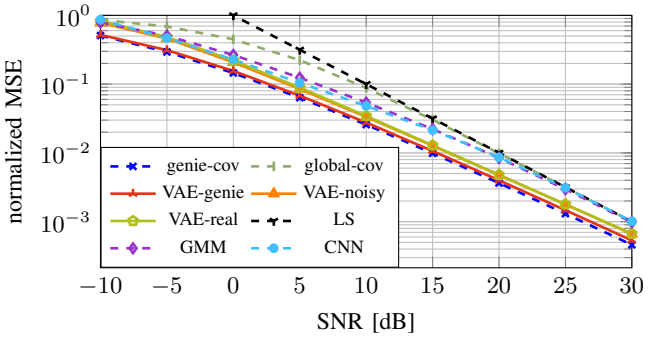

Fig. 12. Normalized MSE for the 3GPP channel model (MIMO case) with three propagation clusters, 32 antennas at the receiver, and 4 antennas at the transmitter. The proposed methods are displayed with solid linestyles.

develop their full potential. From 16 antennas on, the VAE-based methods outperform the baselines and exhibit increasing performance gains if more antennas are considered. All other baselines perform significantly worse than the proposed methods in the large antenna regime. It is also visible that VAE-genie converges to the genie-cov curve. Surprisingly, the VAE-real variant is almost on par with the VAE-noisy variant, although no ground-truth data is available for its training, underlining the strong performance of the VAE as generative prior even in cases with imperfect training data.

A massive amount of antennas is significant for prospective communications systems. Hence, we investigate the large antenna regime in more detail in the following. We inspect the NMSE performance for the 3GPP channel model with one propagation cluster and 128 antennas at the receiver in the SIMO case over the SNR in Fig. 9. The proposed estimators outperform the baseline methods and achieve about 10 dB advantage compared to LS over the whole SNR range.

To highlight the independence of the adopted channel model, we show simulation results for the QuaDRiGa channel model (SIMO case) in Fig. 10. This time, we cannot display the genie-cov curve as the true CCM is unavailable. As can be seen in the plot, all VAE-based estimators show superior NMSE results. The advantages are not as pronounced as in the previous figures but still noticeable.

Fig. 11 shows the NMSE performance for the 3GPP channel model (MIMO case) with one propagation cluster, 32 antennas at the receiver, and 4 antennas at the transmitter. The qualitative behavior of the curves is similar to Fig. 9, where also one propagation cluster is considered. CNN and GMM show the worst NMSE among the ML-based methods for SNRs larger than -5 dB. In this case as well, VAE-noisy and VAE-real show comparable performance. Compared to LS, the VAE-based methods attain a performance gain between 6 and 13 dB. In Fig. 12, we illustrate estimation results for the 3GPP channel model (MIMO case) with three propagation clusters, 32 antennas at the receiver, and 4 antennas at the transmitter. As in the previous figure, the proposed estimators clearly outperform the baselines. However, the performance gaps in Fig. 11 are more noticeable than in Fig. 12. The performance gain compared to LS shrinks to a range from 2 to 11 dB.

In summary, the VAE-based methods exhibit immense performance gains for large antenna arrays, i.e., larger equal 16 antennas and all considered numbers of propagation clusters, significantly outperforming the baseline methods. The strong performance for arrays with many antennas is likely due to the circulant approximation to the Toeplitz CCM, which becomes better for large arrays. VAE-genie lies almost on the genie-cov estimator, and the performance of VAE-noisy and VAE-real is almost identical for all considered scenarios. Moreover, all ML-based methods use genie knowledge during the training phase in the form of ground-truth channel training data, except

for VAE-real, which is trained and evaluated solely based on noisy pilot observations. From this point of view, the strong estimation results of VAE-real are even more meaningful.

VI. Conclusion

This manuscript presents a novel estimation technique based on the VAE. The idea is to tractably model the underlying data distribution as CG via a VAE, yielding a powerful generative prior. The CG modeling allows us to parameterize the MSE-optimal CME under the VAE framework. We propose three estimator variants, of which we find the VAE-real variant particularly appealing as it does not require access to ground-truth data during training or evaluation. We provide theoretical analysis that quantifies the error gap between the proposed MAP-VAE estimator and the CME and relates the training process of the VAE to the NMSE, supporting the strong estimation capabilities of the proposed VAE-based estimators. Our extensive CE simulations highlight that the proposed methods attain excellent performance for various system configurations. In future work, we want to investigate the effects of regularization terms in the training objective and analyze other (especially wide) observation matrices.

Appendix

A. Derivation of (16)

For the parameterized joint PDF of $\boldsymbol{y}$ and $\boldsymbol{h}$ given $\boldsymbol{z}$,

$$p_{\boldsymbol{\theta}}(\boldsymbol{y},\boldsymbol{h}\,|\,\boldsymbol{z}) = p(\boldsymbol{y}\,|\,\boldsymbol{h},\boldsymbol{z})\,p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z}) = p(\boldsymbol{y}\,|\,\boldsymbol{h})\,p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z}) \quad (35)$$

with $p(\boldsymbol{y}\,|\,\boldsymbol{h}) = \mathcal{N}_{\mathbb{C}}(\boldsymbol{A}\boldsymbol{h},\boldsymbol{\Sigma})$, $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z})$ as in (5), and assuming independence of $\boldsymbol{z}$ and $\boldsymbol{n}$. Since the multiplication of two Gaussian distributions is again Gaussian, $p_{\boldsymbol{\theta}}(\boldsymbol{y},\boldsymbol{h}\,|\,\boldsymbol{z}) =$

$$\mathcal{N}_{\mathbb{C}}\left(\begin{bmatrix}\boldsymbol{A}\boldsymbol{\mu}_{\boldsymbol{\theta}}\\ \boldsymbol{\mu}_{\boldsymbol{\theta}}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}^{-1} & -\boldsymbol{\Sigma}^{-1}\boldsymbol{A}\\ -\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Sigma}^{-1} & \boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{A}+\boldsymbol{C}_{\boldsymbol{\theta}}^{-1}\end{bmatrix}^{-1}\right) \quad (36)$$

after some algebraic reformulations, which shows that $\boldsymbol{y}$ and $\boldsymbol{h}$ are jointly Gaussian given $\boldsymbol{z}$. We omit the $\boldsymbol{z}$-argument here for notational brevity. For the derivation of $\mathrm{E}_{\boldsymbol{\theta}}[\boldsymbol{h}\,|\,\boldsymbol{z},\boldsymbol{y}]$, we are interested in the conditional $p_{\boldsymbol{\theta}}(\boldsymbol{h}\,|\,\boldsymbol{z},\boldsymbol{y})$. Using standard results for jointly Gaussian distributions, the conditional is again Gaussian with the mean vector $\mathrm{E}_{\boldsymbol{\theta}}[\boldsymbol{h}\,|\,\boldsymbol{z},\boldsymbol{y}] =$

$$\boldsymbol{\mu}_{\boldsymbol{\theta}} + \left(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{A}+\boldsymbol{C}_{\boldsymbol{\theta}}^{-1}\right)^{-1}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{\mu}_{\boldsymbol{\theta}}) \quad (37)$$

and covariance matrix $(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{A}+\boldsymbol{C}_{\boldsymbol{\theta}}^{-1})^{-1}$. Application of the matrix inversion lemma to (37) and further algebraic reformulations yield (16), concluding the derivation.

B. Proof of Theorem 1

Proof. Let us define the variables

$$\boldsymbol{\Psi} = f_{\boldsymbol{\theta},2}(\boldsymbol{z}) - f_{\boldsymbol{\theta},2}(\boldsymbol{\mu}_{\boldsymbol{\phi}}), \quad \boldsymbol{\psi} = f_{\boldsymbol{\theta},1}(\boldsymbol{z}) - f_{\boldsymbol{\theta},1}(\boldsymbol{\mu}_{\boldsymbol{\phi}}), \quad (38)$$

$$\boldsymbol{\Gamma} = f_{\boldsymbol{\theta},2}(\boldsymbol{\mu}_{\boldsymbol{\phi}}) + \varsigma^2\,\mathbf{I}, \quad (39)$$

and denote $\mathrm{E}_{\boldsymbol{z}}[\cdot] := \mathrm{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{y})}[\cdot]$ for notational convenience. First, let us rewrite $\boldsymbol{T}(\boldsymbol{z}) = \boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})(\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})+\varsigma^2\,\mathbf{I})^{-1}$ as

$$\begin{aligned}\boldsymbol{T}(\boldsymbol{z}) &= (\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z})+\varsigma^2\,\mathbf{I})^{-1}\boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{z}) && (40)\\ &= (\boldsymbol{\Gamma}+\boldsymbol{\Psi})^{-1} f_{\boldsymbol{\theta},2}(\boldsymbol{z}) && (41)\\ &= \left(\boldsymbol{\Gamma}^{-1}-\boldsymbol{\Gamma}^{-1}\boldsymbol{\Psi}(\boldsymbol{\Gamma}+\boldsymbol{\Psi})^{-1}\right) f_{\boldsymbol{\theta},2}(\boldsymbol{z}) && (42)\\ &= \boldsymbol{\Gamma}^{-1} f_{\boldsymbol{\theta},2}(\boldsymbol{\mu}_{\boldsymbol{\phi}}) + \boldsymbol{\Gamma}^{-1}\boldsymbol{\Psi}\left(\mathbf{I}-(\boldsymbol{\Gamma}+\boldsymbol{\Psi})^{-1} f_{\boldsymbol{\theta},2}(\boldsymbol{z})\right) && (43)\end{aligned}$$

by using the push-through identity in (40) and the matrix inversion lemma in (42). Using this result, we rewrite the CME $\mathrm{E}[\boldsymbol{h}\,|\,\boldsymbol{y}]$ from (15) in terms of the MAP-VAE estimator $\hat{\boldsymbol{h}}_{\mathrm{VAE}}(\boldsymbol{y})$ from (19) and an additive error term as

$$\mathrm{E}[\boldsymbol{h}\,|\,\boldsymbol{y}] = \mathrm{E}_{\boldsymbol{z}}[f_{\boldsymbol{\theta},1}(\boldsymbol{z}) + \boldsymbol{T}(\boldsymbol{z})(\boldsymbol{y}-f_{\boldsymbol{\theta},1}(\boldsymbol{z}))] \quad (44)$$

$$\begin{aligned}&= \underbrace{f_{\boldsymbol{\theta},1}(\boldsymbol{\mu}_{\boldsymbol{\phi}}) + \boldsymbol{\Gamma}^{-1} f_{\boldsymbol{\theta},2}(\boldsymbol{\mu}_{\boldsymbol{\phi}})(\boldsymbol{y}-f_{\boldsymbol{\theta},1}(\boldsymbol{\mu}_{\boldsymbol{\phi}}))}_{=\hat{\boldsymbol{h}}_{\mathrm{VAE}}(\boldsymbol{y})}\\ &\quad + \mathrm{E}_{\boldsymbol{z}}[(\mathbf{I}-\boldsymbol{\Gamma}^{-1} f_{\boldsymbol{\theta},2}(\boldsymbol{\mu}_{\boldsymbol{\phi}}))\boldsymbol{\psi}]\\ &\quad + \mathrm{E}_{\boldsymbol{z}}[\boldsymbol{\Gamma}^{-1}\boldsymbol{\Psi}(\mathbf{I}-(\boldsymbol{\Gamma}+\boldsymbol{\Psi})^{-1} f_{\boldsymbol{\theta},2}(\boldsymbol{z}))(\boldsymbol{y}-f_{\boldsymbol{\theta},1}(\boldsymbol{z}))].\end{aligned} \quad (45)$$

We further note that we can simplify the term

$$\begin{aligned}\mathbf{I}-(\boldsymbol{\Gamma}+\boldsymbol{\Psi})^{-1} f_{\boldsymbol{\theta},2}(\boldsymbol{z}))(\boldsymbol{y}-f_{\boldsymbol{\theta},1}(\boldsymbol{z})) &= \boldsymbol{y}-\mathrm{E}[\boldsymbol{h}|\boldsymbol{y},\boldsymbol{z}] && (46)\\ &= \mathrm{E}[\boldsymbol{n}|\boldsymbol{y},\boldsymbol{z}]. && (47)\end{aligned}$$

Thus, we get an upper bound on the expected Euclidean distance between the MAP-VAE estimator and the CME $\varepsilon = \mathrm{E}[\|\,\mathrm{E}[\boldsymbol{h}\,|\,\boldsymbol{y}] - \hat{\boldsymbol{h}}_{\mathrm{VAE}}(\boldsymbol{y})\|_2]$ as

$$\begin{aligned}\varepsilon &\le \mathrm{E}\left[\|\,\mathbf{I}-\boldsymbol{\Gamma}^{-1} f_{\boldsymbol{\theta},2}(\boldsymbol{\mu}_{\boldsymbol{\phi}})\|_2\,\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{\psi}\|_2]\right]\\ &\quad + \mathrm{E}\left[\|\boldsymbol{\Gamma}^{-1}\|_2\cdot\|\,\mathrm{E}_{\boldsymbol{z}}[\boldsymbol{\Psi}\,\mathrm{E}[\boldsymbol{n}|\boldsymbol{y},\boldsymbol{z}]]\|_2\right] && (48)\\ &\le \mathrm{E}\left[\tfrac{\varsigma^2}{\xi_{\min}+\varsigma^2}\,\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{\psi}\|_2]\right] + \tfrac{1}{\varsigma^2}\,\mathrm{E}\left[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{\Psi}\,\mathrm{E}[\boldsymbol{n}|\boldsymbol{y},\boldsymbol{z}]\|_2]\right] && (49)\end{aligned}$$

where we used the reformulation from (47) and the bounds on the spectral norms

$$\|\,\mathbf{I}-\boldsymbol{\Gamma}^{-1} f_{\boldsymbol{\theta},2}(\boldsymbol{\mu}_{\boldsymbol{\phi}})\|_2 = \frac{\varsigma^2}{\xi_{\min}+\varsigma^2}, \quad \|\boldsymbol{\Gamma}^{-1}\|_2 \le \frac{1}{\varsigma^2}, \quad (50)$$

together with the triangle and Cauchy-Schwarz inequalities. Note that $\xi_{\min}$ is a function of $\boldsymbol{y}$. Thus, we employ Hölder's inequality for both summands in (49) to get

$$\begin{aligned}\varepsilon &\le C_1\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{\psi}\|_2^2]}\\ &\quad + \frac{1}{\varsigma^2}\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\,\mathrm{E}[\boldsymbol{n}|\boldsymbol{y},\boldsymbol{z}]\|_2^2]}\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{\Psi}\|_2^2]}} && (51)\\ &\le C_1\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{\psi}\|_2^2]} + \frac{1}{\varsigma^2}\sqrt{\mathrm{E}[\|\boldsymbol{n}\|_2^2]}\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{\Psi}\|_2^2]} && (52)\\ &= C_1\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{\psi}\|_2^2]} + C_2\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{\Psi}\|_2^2]} && (53)\end{aligned}$$

where we used Jensen's inequality in combination with the law of total expectation in (52), and with C_1, C_2 in (25). By resubstituting the variables in (38), we get

$$\begin{aligned}\varepsilon &\le C_1\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|f_{\boldsymbol{\theta},1}(\boldsymbol{z})-f_{\boldsymbol{\theta},1}(\boldsymbol{\mu}_{\boldsymbol{\phi}})\|_2]}\\ &\quad + C_2\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|f_{\boldsymbol{\theta},2}(\boldsymbol{z})-f_{\boldsymbol{\theta},2}(\boldsymbol{\mu}_{\boldsymbol{\phi}})\|_2^2]}} && (54)\\ &\le (C_1L_1+C_2L_2)\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{z}-\boldsymbol{\mu}_{\boldsymbol{\phi}}\|_2^2]}} && (55)\end{aligned}$$

by using the Lipschitz continuity (23). By defining the first and second moments of the posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z} \,|\, \boldsymbol{y})$ as $\bar{\boldsymbol{\mu}} = \mathrm{E}_{\boldsymbol{z}}[\boldsymbol{z}]$ and $\bar{\boldsymbol{C}} = \mathrm{E}_{\boldsymbol{z}}[(\boldsymbol{z} - \bar{\boldsymbol{\mu}})(\boldsymbol{z} - \bar{\boldsymbol{\mu}})^{\mathrm{H}}]$, we write

$$\sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{z} - \boldsymbol{\mu}_{\boldsymbol{\phi}}\|_2^2]]} = \sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[\|\boldsymbol{z} - \bar{\boldsymbol{\mu}} + \bar{\boldsymbol{\mu}} - \boldsymbol{\mu}_{\boldsymbol{\phi}}\|_2^2]]} \quad (56)$$

$$\leq \sqrt{\mathrm{E}[\mathrm{E}_{\boldsymbol{z}}[(\|\boldsymbol{z} - \bar{\boldsymbol{\mu}}\|_2 + \|\bar{\boldsymbol{\mu}} - \boldsymbol{\mu}_{\boldsymbol{\phi}}\|_2)^2]]} \quad (57)$$

$$\leq \sqrt{\mathrm{tr}(\bar{\boldsymbol{C}})} + \sqrt{\mathrm{E}[\|\bar{\boldsymbol{\mu}} - \boldsymbol{\mu}_{\boldsymbol{\phi}}\|_2^2]} \quad (58)$$

since only one summand in (57) depends on $\boldsymbol{y}$ or $\boldsymbol{z}$, respectively. Plugging (58) in (55) yields (24), completing the proof. □

REFERENCES

[1] L. Ruthotto and E. Haber, "An introduction to deep generative modeling," *GAMM-Mitteilungen*, vol. 44, no. 2, p. e202100008, 2021.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014.

[4] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent.*, 2014.

[5] I. Goodfellow *et al.*, "Generative Adversarial Networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[6] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[7] G. Ongie *et al.*, "Deep Learning Techniques for Inverse Problems in Imaging," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 39–56, 2020.

[8] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed Sensing using Generative Models," *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, pp. 537–546, 2017.

[9] A. Jalal *et al.*, "Robust Compressed Sensing MRI with Deep Generative Priors," in *Adv. Neural Inf. Process. Syst.*, 2021.

[10] P. Hand, O. Leong, and V. Voroninski, "Phase Retrieval Under a Generative Prior," in *Adv. Neural Inf. Process. Syst.*, 2018.

[11] M. Asim, F. Shamshad, and A. Ahmed, "Blind Image Deconvolution Using Deep Generative Priors," *IEEE Trans. Comput. Imaging*, vol. 6, pp. 1493–1506, 2020.

[12] M. Koller, B. Fesl, N. Turan, and W. Utschick, "An Asymptotically MSE-Optimal Estimator Based on Gaussian Mixture Models," *IEEE Trans. Signal Process.*, vol. 70, pp. 4109–4123, 2022.

[13] E. Balevi and J. G. Andrews, "Wideband Channel Estimation With a Generative Adversarial Network," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 5, pp. 3049–3060, 2021.

[14] E. Balevi, A. Doshi, A. Jalal, A. Dimakis, and J. G. Andrews, "High Dimensional Channel Estimation Using Deep Generative Networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 18–30, 2021.

[15] A. S. Doshi, M. Gupta, and J. G. Andrews, "Over-the-Air Design of GAN Training for mmWave MIMO Channel Estimation," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 3, pp. 557–573, 2022.

[16] M. Arvinte and J. I. Tamir, "MIMO Channel Estimation Using Score-Based Generative Models," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 6, pp. 3698–3713, 2023.

[17] T. Diskin, Y. C. Eldar, and A. Wiesel, "Learning to Estimate Without Bias," *IEEE Trans. Signal Process.*, vol. 71, pp. 2162–2171, 2023.

[18] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1993.

[19] S. Zhao, J. Song, and S. Ermon, "Towards a Deeper Understanding of Variational Autoencoding Models," *arXiv preprint arXiv:1702.08658*, 2017.

[20] M. Baur, B. Fesl, M. Koller, and W. Utschick, "Variational Autoencoder Leveraged MMSE Channel Estimation," in *56th Asilomar Conf. Signals, Syst., Comput.*, 2022, pp. 527–532.

[21] M. Rani, S. B. Dhok, and R. B. Deshmukh, "A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications," *IEEE Access*, vol. 6, pp. 4875–4894, 2018.

[22] A. Banerjee, X. Guo, and H. Wang, "On the Optimality of Conditional Expectation as a Bregman Predictor," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2664–2669, 2005.

[23] D. P. Kingma and M. Welling, "An Introduction to Variational Autoencoders," *Found. Trends® Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[24] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*, 2nd ed. Nashua, NH: Athena Scientific, 2008.

[25] J. Yang *et al.*, "Compressive Sensing by Learning a Gaussian Mixture Model From Measurements," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 106–119, 2015.

[26] M. Loève, *Probability Theory I*, 4th ed. Springer New York, NY, 1977.

[27] D. Fuhrmann, "Application of Toeplitz covariance estimation to adaptive beamforming and detection," *IEEE Trans. Signal Process.*, vol. 39, no. 10, pp. 2194–2198, 1991.

[28] Y. Ephraim, D. Malah, and B.-H. Juang, "On the application of hidden Markov models for enhancing noisy speech," *IEEE Trans. Acoust.*, vol. 37, no. 12, pp. 1846–1856, 1989.

[29] D. Neumann, T. Wiese, and W. Utschick, "Learning The MMSE Channel Estimator," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2905–2917, 2018.

[30] B. Fesl *et al.*, "Channel Estimation based on Gaussian Mixture Models with Structured Covariances," in *2022 56th Asilomar Conf. Signals, Syst. Comput.*, 2022, pp. 533–537.

[31] M. Baur, N. Turan, B. Fesl, and W. Utschick, "Channel Estimation in Underdetermined Systems Utilizing Variational Autoencoders," in *2024 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024, pp. 9031–9035.

[32] M. Baur, B. Böck, N. Turan, and W. Utschick, "Variational Autoencoder for Channel Estimation: Real-World Measurement Insights," *arXiv preprint arXiv:2312.03450*, 2023.

[33] R. M. Gray, "Toeplitz and Circulant Matrices: A Review," *Found. and Trends® in Commun. and Inf. Theory*, no. 3, pp. 155–239, 2006.

[34] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY: Cambridge University Press, 2005.

[35] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *J. Mach. Learn. Res.*, vol. 13, no. 10, pp. 281–305, 2012.

[36] R. Liaw *et al.*, "Tune: A Research Platform for Distributed Model Selection and Training," *arXiv preprint arXiv:1807.05118*, 2018.

[37] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[38] T. Salimans and D. P. Kingma, "Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2016.

[39] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," in *Adv. Neural Inf. Process. Syst. - Deep Learn. Symp.*, 2016.

[40] S. Anwar, K. Hwang, and W. Sung, "Structured Pruning of Deep Convolutional Neural Networks," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 3, pp. 1–18, 2017.

[41] 3GPP, "Spatial channel model for Multiple Input Multiple Output (MIMO) simulations (Release 16)," 3rd Generation Partnership Project (3GPP), Tech. Rep. 25.996 V16.0.0, 2020.

[42] J. Kermoal, L. Schumacher, K. Pedersen, P. Mogensen, and F. Frederiksen, "A Stochastic MIMO Radio Channel Model With Experimental Validation," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 6, pp. 1211–1226, 2002.

[43] S. Jaeckel, L. Raschkowski, K. Borner, and L. Thiele, "QuaDRiGa: A 3-D Multi-Cell Channel Model With Time Evolution for Enabling Virtual Field Trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, 2014.

[44] S. Jaeckel *et al.*, "QuaDRiGa - Quasi Deterministic Radio Channel Generator, User Manual and Documentation," Fraunhofer Heinrich Hertz Institute, Tech. Rep. v2.6.1, 2021.

[45] S. A. Busari, K. M. S. Huq, S. Mumtaz, L. Dai, and J. Rodriguez, "Millimeter-Wave Massive MIMO Communication for Future Wireless Systems: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 2, pp. 836–869, 2018.

[46] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *2010 IEEE Inf. Theory Work. Inf. Theory*. IEEE, 2010, pp. 1–5.

[47] A. Maleki, L. Anitori, Z. Yang, and R. G. Baraniuk, "Asymptotic Analysis of Complex LASSO via Complex Approximate Message Passing (CAMP)," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4290–4308, 2013.

[48] B. Fesl, N. Turan, M. Koller, and W. Utschick, "A Low-Complexity MIMO Channel Estimator with Implicit Structure of a Convolutional Neural Network," in *22nd Int. Work. Signal Process. Adv. Wirel. Commun.*, 2021, pp. 11–15.