Investigating the Generalization Capabilities of Deep FAVIB: A Data-Driven Information Bottleneck-Based Quantization Scheme for Noisy Channels

Matthias Hummert, Shayan Hassanpour, Dirk Wübben, and Armin Dekorsy
Department of Communications Engineering,
University of Bremen,
28359 Bremen, Germany
Email: {hummert, hassanpour, wuebben, dekorsy}@ant.uni-bremen.de

***Abstract*—We consider a generic two-hop transmission setup. Explicitly, a source signal is transmitted over an imperfect channel, yielding a noisy observation. This signal shall then be compressed at a relay node before getting transmitted further over an error-prone and rate-limited channel to the sink, where the source signal is decoded/reconstructed. In [1], [2], we presented a data-driven Information Bottleneck-based quantization scheme called Deep FAVIB. For that, we derived a tractable variational lower-bound of the original objective functional that could be optimized using samples and utilized Deep Neural Networks (DNNs) to realize both the quantizer/encoder at the relay node and the decoder at the sink. Based on this work, we now provide further investigations, showcasing the excellent generalization capabilities of Deep FAVIB by several Symbol-Error-Rate (SER) simulation results. Specifically, we apply the pretrained Deep FAVIB to different environments that have not been present in the training, and show that yet, it yields promising results. This gives clear evidence to the fact that Deep FAVIB can be considered as a practically efficient scheme to be utilized, especially when dealing with the highly dynamic and challenging environments.**

***Index Terms*—6G, deep learning, information bottleneck, joint source-channel coding, NTN, variational auto-encoders**

I. Introduction

Consider a communication system where a User Equipment (UE) transmits its signal over an access channel to a relay node. This relay node compresses the received signal before forwarding it over an *error-prone* and *rate-limited* channel to its destination where the source signal is reconstructed. This setup is of high interest for future communication systems such as Non-Terrestrial Networks (NTNs). NTNs utilize satellites and drones to enable global connectivity, especially, for the low-populated areas. Therein, relaying aspects are key, since information has to be distributed over different satellites to achieve a good coverage. A relevant example of such satellite-aided systems has been shown in Fig. 1.

Next to NTN applications, this generic setup is also found in a variety of terrestrial applications, e.g., Cloud-based Radio Access Networks (Cloud-RANs) [3], [4], distributed inference sensor networks with imperfect channels to the fusion center [5], [6] and Cell-Free massive Multiple-Input Multiple-Output (CF-mMIMO) systems [7]–[10] with non-ideal fronthaul links.

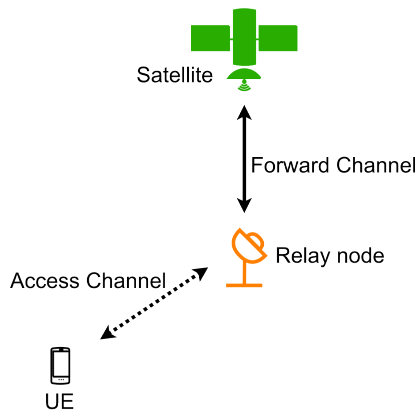


Fig. 1. Example of a two-hop transmission setup: A noisy signal from a UE is received by an on-ground relay node, compressed, and finally forwarded to a satellite transponder via an error-prone forward link.

To design the local compressor, we utilize the Deep FAVIB approach [1], [2]. It is a data-driven solution that is based upon the Forward-Aware Vector Information Bottleneck (FAVIB) algorithm [11]. It makes use of the Information Bottleneck (IB) design method [12]–[14] and leverages Deep Neural Networks (DNNs) to realize the encoding and decoding functionalities.

Deep FAVIB is an extension of some well-known concepts, e.g., the *Variational Auto-Encoders (VAEs)* [15], [16] and *Deep Variational Information Bottleneck (Deep VIB)* [17] to the context of joint source-channel coding. A VAE transforms an input signal into a latent variable (of typically much lower dimension) before reconstructing it via a decoder. An extension of this chain, Deep VIB [17], transforms a *noisy* observation of the input signal into a latent variable. This forms a remote source coding scheme. Extending Deep VIB to a joint-source-channel coding setup, we get the Deep FAVIB approach. It integrates the impacts of an *error-prone* foward channel (FC) into the *joint* training of the encoder and decoder DNNs in the considered two-hop transmission scenario. It has been shown that the performance of Deep FAVIB is on par with the State-of-the-Art (SotA) model-based scheme [1], [2].

979-8-3315-2289-6/25/$31.00 ©2025 IEEE

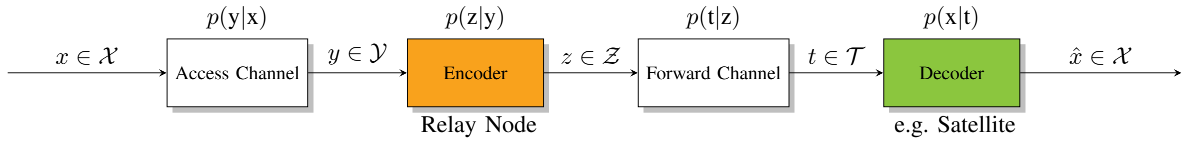


Fig. 2. Two-hop transmission model: A user sends symbol x over an access channel. The observation y is quantized by an encoder to signal z. This signal is then forwarded via an error-prone channel, and the received signal t is fed into a decoder to recover the sent symbol.

In this paper, we intend to provide some further insights into the generalization capabilities of Deep FAVIB scheme through several numerical investigations. Specifically, we train Deep FAVIB once for a specific scenario and then apply it to other setups (that were not available in the training phase) without any retraining. Yet, we show that Deep FAVIB yields a promising performance. By this, we highlight the potential of Deep FAVIB to be applied efficiently in practical systems, especially, when dealing with dynamic environments in which the statistics of access link might change frequently.

Notation: The discrete random variable, a, takes a certain realization, $a \in \mathcal{A}$, according to its probability mass function $p(\mathrm{a})$. Using boldface, the random vector, **a**, is given. $I(\cdot;\cdot)$, $H(\cdot)$, and $D_{\mathrm{KL}}(\cdot||\cdot)$, denote the Mutual Information (MI), the Shannon's entropy and the Kullback-Leibler divergence [18]. $\mathbb{E}_{\bullet}$ denotes the expectation operator.

II. System Overview

A. System Model and Problem Formulation

Fig. 2 depicts the general two-hop transmission model. A source emits discrete-valued modulated symbols $x \in \mathcal{X}$, which are sent over an access channel $p(\mathrm{y}|\mathrm{x})$. The channel distorts the source signal, yielding the received signal $y \in \mathcal{Y}$. The encoder $p(\mathrm{z}|\mathrm{y})$ quantizes the signal y to signal $z \in \mathcal{Z}$ with a certain cardinality $|\mathcal{Z}| = N$. Thereafter, the signal z gets transmitted over a *rate-limited* and *error-prone* forward channel $p(\mathrm{t}|\mathrm{z})$ with capacity R. This yields additional distortions in the forwarded signal $t \in \mathcal{T}$. Finally, at the sink / destination, a decoder $p(\mathrm{x}|\mathrm{t})$ reconstructs the source signal $\hat{x}$. This forms a (remote) joint source-channel coding scheme as the quantizer / encoder is designed in a fashion that it considers the forward channel imperfections when compressing the noisy observation y.

Next, we concisely reiterate the derivations of model-based (FAVIB) compression scheme.

B. Model-Based Design (FAVIB)

To start the technical discussion, we construct the quantizer $p(\mathrm{z}|\mathrm{y})$ using the FAVIB approach [11]. For that, we need to know the full input statistics , i.e., $p(\mathrm{x},\mathrm{y})$. In principle, the goal is to design the quantizer in such a way that the information about the source x is maximized in signal t, while compressing the noisy received signal y. We formulate this as

$$p^{\star}(\mathrm{z}|\mathrm{y}) = \underset{p(\mathrm{z}|\mathrm{y}):\, I(\mathrm{y};\mathrm{z}) \leq R}{\operatorname{argmax}} \; I(\mathrm{x};\mathrm{t}) \;, \tag{1}$$

where we maximize the MI between x and t in order to achieve the best information flow from the source to the sink. The constraint w.r.t. the forward channel capacity R provides an upper-bound to the compression rate (i.e., the MI between y and z). We can rewrite this design problem with the Lagrange Method of multipliers [19] as

$$p^{\star}(\mathrm{z}|\mathrm{y}) = \underset{p(\mathrm{z}|\mathrm{y})}{\operatorname{argmax}} \; \underbrace{I(\mathrm{x};\mathrm{t}) - \lambda I(\mathrm{y};\mathrm{z})}_{\mathcal{L}_{\text{FAVIB}}} \;, \tag{2}$$

where $\lambda \geq 0$ is directly related to the limit R. We can see a fundamental trade-off between the information maximization between x and t (reconstruction), and the compression between y and z. We can further see that we incorporate the forward channel into the optimization problem as we maximize the MI between the source x and the output of the forward channel t. Solving (2) yields a stationary solution, derived in [11] for each pair $(y,z) \in \mathcal{Y} \times \mathcal{Z}$ as

$$p^{\star}(z|y) = \frac{p(z)}{\omega(y,\lambda)} \exp\Big(-\lambda^{-1} \sum_{t \in \mathcal{T}} p(t|z) D_{\mathrm{KL}}\big(p(\mathrm{x}|y)||p(\mathrm{x}|t)\big)\Big), \tag{3}$$

where $\omega(y,\lambda)$ is a normalization function. This forms the core of an iterative algorithm, namely, FAVIB [11], to efficiently address the design problem (1). Typically, for $\lambda > 0$, a *soft* quantizer is achieved.

III. Data-Driven Design (Deep FAVIB)

In this section, we present the Deep FAVIB which is the sample-based counterpart of the FAVIB algorithm. By that, we do not require the full input statistics $p(\mathrm{x},\mathrm{y})$ and only need samples of the inputs. We collect them in a training data set $\{x_m, y_m\}_{m=1}^{M}$ with M being the number of total samples. We solve an approximation of the FAVIB design problem (2).

A. The Variational Lower-Bound

Based on the objective function $\mathcal{L}_{\text{FAVIB}}$ in (2) we introduce a tractable *Variational Lower-Bound (VLB)*. We write

$$\mathcal{L}^{\text{FAVIB}} = I(\mathrm{x};\mathrm{t}) - \lambda I(\mathrm{y};\mathrm{z}) \geq A - \lambda B = \mathcal{L}^{\text{VLB}} \;, \tag{4}$$

in which we introduce variables A for reconstruction and B for compression. B is an upper-bound for compression, i.e., $I(\mathrm{y};\mathrm{z}) \leq B$, and A is a lower-bound for reconstruction, i.e., $I(\mathrm{x};\mathrm{t}) \geq A$. For A we can write

$$I(\mathrm{x};\mathrm{t}) = \underbrace{H(\mathrm{x})}_{\geq 0} - H(\mathrm{x}|\mathrm{t}) \tag{5a}$$

$$\geq \underbrace{\sum_{t \in \mathcal{T}} p(t) D_{\mathrm{KL}}\big(p(\mathrm{x}|t)||q(\mathrm{x}|t)\big)}_{\geq 0} + \sum_{x \in \mathcal{X}, t \in \mathcal{T}} p(x,t) \log q(x|t) \tag{5b}$$

$$\geq E_{\mathrm{x},\mathrm{t}}\{\log q(\mathrm{x}|\mathrm{t})\} = A, \tag{5c}$$

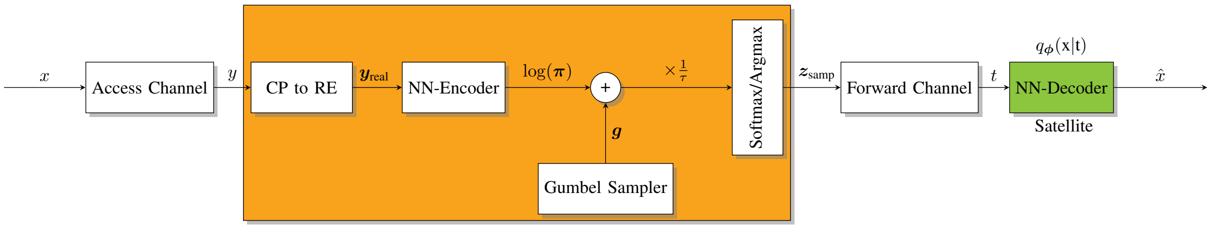


Fig. 3. Detailed Deep FAVIB learning architecture, consisting of two DNNs for encoder and decoder, a softmax/argmax unit and a Gumbel $(0, 1)$ sampler.

where $q(\mathrm{x}|\mathrm{t})$, i.e., the proxy posterior, is introduced to replace the perfect decoder $p(\mathrm{x}|\mathrm{t})$. For the compression, we can write the following

$$I(\mathrm{y};\mathrm{z}) = \sum_{y\in\mathcal{Y},\, z\in\mathcal{Z}} p(y,z)\log\frac{p(z|y)}{r(z)} - \underbrace{D_{\mathrm{KL}}\big(p(\mathrm{z})\|r(\mathrm{z})\big)}_{\geq 0} \tag{6a}$$

$$\leq \mathbb{E}_{\,\mathrm{y,z}}\left\{\log\frac{p(\mathrm{z}|\mathrm{y})}{r(\mathrm{z})}\right\} = B, \tag{6b}$$

wherein $r(\mathrm{z})$ is an arbitrary prior for z. For the VLB, we get

$$\mathcal{L}^{\mathrm{VLB}} = E_{\mathrm{x,t}\sim p(\mathrm{x,t})}\{\log q(\mathrm{x}|\mathrm{t})\} - \lambda E_{\mathrm{y,z}\sim p(\mathrm{y,z})}\left\{\log\frac{p(\mathrm{z}|\mathrm{y})}{r(\mathrm{z})}\right\}. \tag{7}$$

To optimize this VLB, we need to introduce two parameterized distributions (getting realized by DNNs) for the decoder $q(\mathrm{x}|\mathrm{t})$ and the quantizer / encoder $p(\mathrm{z}|\mathrm{y})$. We get

$$\begin{aligned}\mathcal{L}^{\mathrm{DNN}} &= E_{\mathrm{x,t}\sim p(\mathrm{x,t})}\{\log q_{\boldsymbol{\phi}}(\mathrm{x}|\mathrm{t})\} - \lambda E_{\mathrm{y,z}\sim p(\mathrm{y,z})}\left\{\log\frac{p_{\boldsymbol{\theta}}(\mathrm{z}|\mathrm{y})}{r_{\boldsymbol{\psi}}(\mathrm{z})}\right\}\\ &= \underbrace{E_{\mathrm{t}\sim p(\mathrm{t})}\{E_{\mathrm{x}\sim p(\mathrm{x}|\mathrm{t})}\{\log q_{\boldsymbol{\phi}}(\mathrm{x}|\mathrm{t})\}\}}_{\text{reconstruction}}\\ &\qquad - \lambda\underbrace{E_{\mathrm{y}\sim p(\mathrm{y})}\{D_{\mathrm{KL}}(p_{\boldsymbol{\theta}}(\mathrm{z}|\mathrm{y})\|r_{\boldsymbol{\psi}}(\mathrm{z}))\}}_{\text{regularization}},\end{aligned} \tag{8}$$

with the weights $\boldsymbol{\psi}$, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. λ is again a trade-off parameter between reconstruction and compression.

Recall that our goal is to maximize (8). For the first part of the equation (i.e., reconstruction), we maximize the relevant information, corresponding to minimizing the *cross-entropy* loss, averaged over t. This follows the *Maximum-Likelihood* learning rule [20], and is a popular loss for classification. For the second part of the equation, a regularization is present as a Kullback-Leibler Divergence (KLD) term, averaged over y.

B. NN Architecture and Implementation Details

In general, we want to design a *soft / stochastic* encoder $p_{\boldsymbol{\theta}}(\mathrm{z}|\mathrm{y})$ via a DNN, therefore we apply the reparametrization trick [15] to allow sampling and calculate the gradients of $\mathcal{L}^{\mathrm{DNN}}$. This decouples sampling and the gradient calculation. On top, we want to realize a *discrete* latent variable, therefore we use the *Gumbel-Softmax* trick [21], [22]. This yields a soft approximation for our categorical distribution, enabling the gradient calculation. The detailed learning architecture has been illustrated in Fig. 3.

The input to our encoder is the noisy source signal $y \in \mathcal{Y}$, which is (usually) complex-valued. Hence, we need to stack the real and imaginary parts in a 2D vector, namely, $\boldsymbol{y}_{\mathrm{real}}$, since Neural Networks cannot handle the complex numbers straightforwardly. The output of the NN-encoder $\log(\boldsymbol{\pi}) \in \mathbb{R}^N$ directly represents the log-probabilities of the categorical distribution of z. This signal is then combined with N i.i.d. samples drawn from the Gumbel $(0, 1)$ distribution and stacked in the vector $\boldsymbol{g} \in \mathbb{R}^N$. The combined signal $\log(\boldsymbol{\pi}) + \boldsymbol{g}$ then flows into a softmax / argmax unit. During *inference*, argmax is used, yielding *one-hot* outputs / vectors, meaning one entry is set to 1, while all other $N-1$ entries are set to 0. For *training*, softmax is applied as no gradients can be calculated for argmax. For softmax training, another hyperparameter τ is introduced. The combination of softmax with τ approximates the argmax function. This generates the i-th entry of $\boldsymbol{z}_{\mathrm{samp}}$

$$z_{\mathrm{samp},i} = \frac{\exp\Big(\big(\log(\pi_i) + g_i\big)/\tau\Big)}{\sum_{j=1}^{N}\exp\Big(\big(\log(\pi_j) + g_j\big)/\tau\Big)} \in [0, 1], \tag{9}$$

where $\tau > 0$. If τ is small, the softmax approximates argmax more steeply, yielding rapid gradient changes. On the other hand, large τ values yield a smooth softmax and may enable better optimization as gradients change slowly while flowing through. This quantizer yields the compressed signal z, which is then forwarded over an *error-prone* channel, resulting in the signal t. Finally, the NN-decoder $q_{\boldsymbol{\phi}}(\mathrm{x}|\mathrm{t})$ (with weights $\boldsymbol{\phi}$) is applied to recover the source signal. This decoder is a standard feed-forward DNN. This whole system can be interpreted as an extension to the *VAE* structure. That is, the input is a *noisy* observation instead of the source signal itself and the latent variable is further disturbed before getting reconstructed by the decoder.

C. Supervised Learning and DNNs

DNNs are (nonlinear) functions with trainable weights, here our encoder and decoder DNNs with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, which are *jointly* trained w.r.t. a loss function, in this system $-\mathcal{L}^{\mathrm{DNN}}$ (8). These weights are updated by using our data set $\{x_m, y_m\}_{m=1}^{M}$. To update the weights, we can apply *Stochastic Gradient Descent (SGD)*. SGD calculates the derivative of the loss function w.r.t. the weights of a subset of the data set in order to minimize the loss and updates the weights accordingly. The prior $r_{\boldsymbol{\psi}}(z)$ has its own trainable parameters, but not being realized by a DNN.

IV. Numerical Results

Based on the results of our previous works [1], [2] and the fact that we achieved the performance of SotA model-based scheme, we now showcase the generalization capabilities of Deep FAVIB scheme by providing Symbol-Error-Rate (SER) simulations. To that end, we train Deep FAVIB for one specific setup and evaluate it for other setups (that were not present during the training) without retraining. Please recall that Deep FAVIB is the sample-based counterpart of FAVIB [11], hence not requiring the prior knowledge of the joint input statistics. By that, we extend the application range of our approach as samples / data sets ($\{x_m, y_m\}_{m=1}^{M}$) are usually more widely available than the joint (input) statistics.

For our system setup, we use Quadrature Phase Shift Keying (QPSK) as the source alphabet, an Additive White Gaussian Noise (AWGN) access channel with the noise variance σ_{n}^2. For the forward channel, we use an N-ary symmetric model, meaning that a cluster flip occurs with probability $\frac{e}{N-1}$ to one of the $N-1$ remaining clusters, and with probability $1-e$, the cluster is correctly forwarded. The parameters of this system are then τ (the so-called temperature to control the smoothness of softmax (9)), λ (the trade-off in loss (8)), σ_{n}^2 (the noise power of the access channel), and e (the error probability of the forward channel).

Deep FAVIB is only trained once for each parameter setup, where a maximum of 10000 training epochs are used, with a batch size of 10000 and $M = 1e6$ samples. We apply the *Early Stopping* to store the weights with the lowest training loss. As our SGD variant, we use the Adam optimizer [23] with a learning rate of 10^{-5}. The feed-forward configurations of the encoder and decoder DNNs have been given in Table I. These configurations were found experimentally to have low-complex DNNs without sacrificing the performance. We use 3 hidden layers and the Rectified Linear Unit (ReLu) activation functions. The output layer of the NN-encoder applies no activation function to construct the log-probabilities $\log(\boldsymbol{\pi})$. For the NN-decoder the output layer uses a softmax activation function to classify and reconstruct the source symbols in $\mathcal{X}$.

To showcase the generalization capabilities of Deep FAVIB, we apply the pretrained Deep FAVIB to new system setups and check the obtained performance. Therefore, we load the weights of Deep FAVIB and apply a new test data set for each new setup given by $\{x_{m,\,\mathrm{test}}, y_{m,\,\mathrm{test}}\}_{m=1}^{M_{\mathrm{test}}}$. We also use a different performance metric, the SER of the source alphabet, given as

$$\mathrm{SER} = \frac{\text{\# of wrong symbols}}{M_{\mathrm{test}}}, \tag{10}$$

where $M_{\mathrm{test}} = 1e6$ represents the total number of test samples for each tested setup. By that, we get a practical performance measure for the whole system. For the upcoming setups, we choose $\lambda = 0.01$, hence solely focusing on the reconstruction part of the loss. As found in our previous evaluations [1], [2] we choose $\tau = 1$ since this temperature value yields a good performance trade-off.

TABLE I

Configurations for encoder DNN, decoder DNN, and prior

Name	# of Hidden Layers	width of layers	# of weights
$p_{\boldsymbol{\theta}}(\mathrm{z}\vert\mathrm{y})$	3	300, 200, 100	82816
$q_{\boldsymbol{\phi}}(\mathrm{x}\vert\mathrm{t})$	3	300, 200, 100	85602
$r_{\boldsymbol{\psi}}(\mathrm{z})$	0	0	N

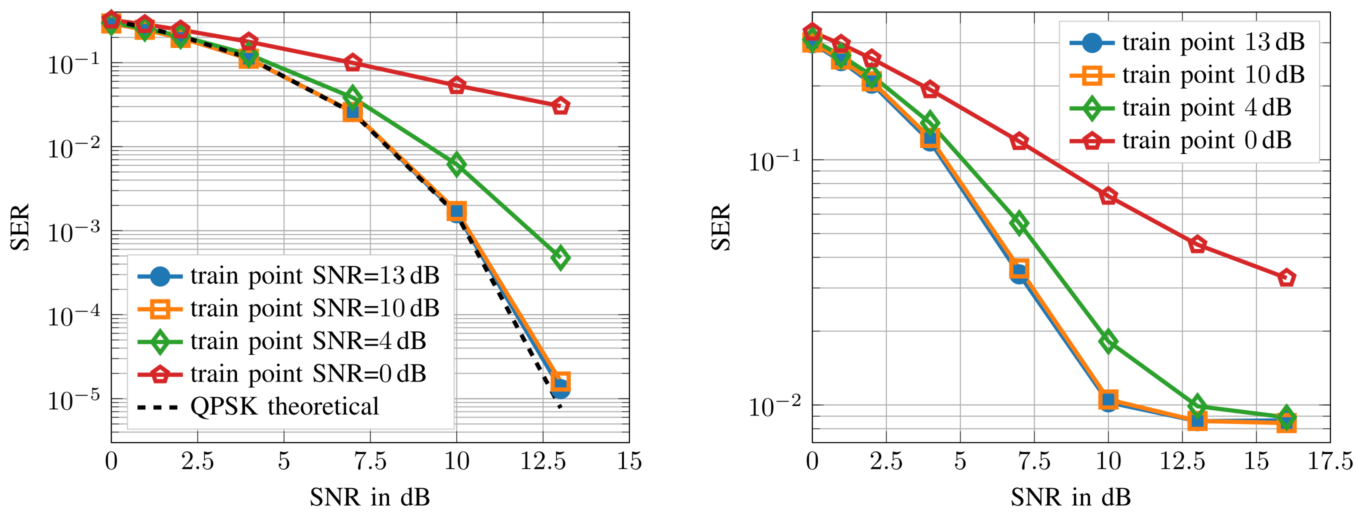
A. 3-bit Quantization (Error-Free & Error-Prone Forwarding)

First, we show the performance results for a relatively small 3-bit quantization, i.e., $N = 8$. We commence our investigation with an error-free forward channel, i.e., $e = 0$. For that, we train Deep FAVIB for 4 different noise powers and evaluate it afterwards on a certain range of Signal-to-Noise-Ratio (SNR) values. We show the plot in Fig. 4a. As a general trend, we observe that all Deep FAVIB setups achieve a lower SER for higher SNR values. Furthermore, as expected, we see that, generally, for each Deep FAVIB training point, the pertinent curve shows the best or on par performance (with other curves) on that specific training points. For example, for a training point of 0 dB, the red curve shows performance on par with all other depicted results at SNR of 0 dB.

On top, we can see that the SER performance worsens, the lower the training SNR becomes. For instance, for a training SNR of 0 dB (i.e., the red curve) we observe the worst overall performance. Furthermore, for this curve, the performance gap widens for large SNR values, the further we move away from the training point. Interestingly, this trend only holds true for rather low training SNR, i.e. 0 dB and 4 dB. Both curves (i.e., red and green) still converge to a lower SER for higher SNR values, but fall short in comparison with the other high SNR training points.

For high SNR training points, i.e., 10 dB and 13 dB, we see the overall best performance and generalization capability, as the performances are very close to each other, even for low SNR values. This means that, the Deep FAVIB setups that have been trained for high SNR values can be used without performance loss for low SNR values as well. Interestingly, for the high training points, we can also observe the performance on par with an optimally demapped QPSK without any quantization. These results are expected, since essentially, for high SNR values, 4 correctly placed quantization regions yield the optimal demapping. These quantization regions are not found for low SNR training points, as the noise during training is more relevant, hence yielding extra quantization regions which limit the performance for high SNR values.

In Fig. 4b, we depict the performance for an error-prone forward channel with $e = 0.01$. Explicitly, now we observe an error floor which is appearing in the SER performances. Nevertheless, the overall behavior of the curves follow the same trend as observed in Fig. 4a. The high SNR training points yield the best performance. Interestingly, the green curve which is trained for lower SNR, shows performance close to the orange and blue curves. For 10 dB and 13 dB, the curves match again. The red curve (i.e., 0 dB), shows the largest performance gap, although still converging to lower



(a) error-free forwarding ($e = 0$)

(b) error-prone forwarding ($e = 0.01$)

Fig. 4. SER versus SNR for different noise training points for a) an error-free forward channel ($e = 0$) and b) an error-prone forward channel ($e = 0.01$), with $N = 8$ clusters and $\lambda = 0.01$.

SER for higher SNR values. All in all, it is observed that, also for the case of error-prone forward channels, Deep FAVIB trained for higher SNR values yields the best performance. The limiting impact of the imperfect forwarding is observable on the obtained performance results.

Summarizing the findings for 3-bit quantization, the Deep FAVIB shows the best generalization capability for high SNR training points. Even for low SNR training points, the performance is as good as the ones obtained by lower SNR training points. Moreover, for perfect forwarding, the performance for the high training points match the performance of an optimal demapper for QPSK as we only need 4 decision regions to do so. This is not the case for low SNR training points as more quantization regions limit that performance. For error-prone forward channels, as expected, an error floor is visible in the obtained performance curves due to the limiting impacts of imperfect forwarding.

B. *5-bit Quantization (Error-Free & Error-Prone Forwarding)*

We now show the performance results for a rather large 5-bit quantization, i.e., $N = 32$ clusters. Here again, we commence with the SER performance curves over the SNR for an error-free forward channel that have been depicted in Fig. 5a.

The general findings of the previous investigations hold true. On top, we can see that the performance for the high SNR training points (i.e., orange and blue curves) remains similar, while the lower training points (i.e., red and green) show better overall convergence and performance behavior. Therefore, it can be deduced that more quantization clusters increase the overall generalization capability of Deep FAVIB. Nevertheless, for low SNR training points, there is still a performance gap to the optimal QPSK demapper, whose performance is achieved by the high SNR training points.

We conclude this section by the SER performance curves for 5-bit quantization over an error-prone forward channel with $e = 0.01$ that have been illustrated in Fig. 5b.

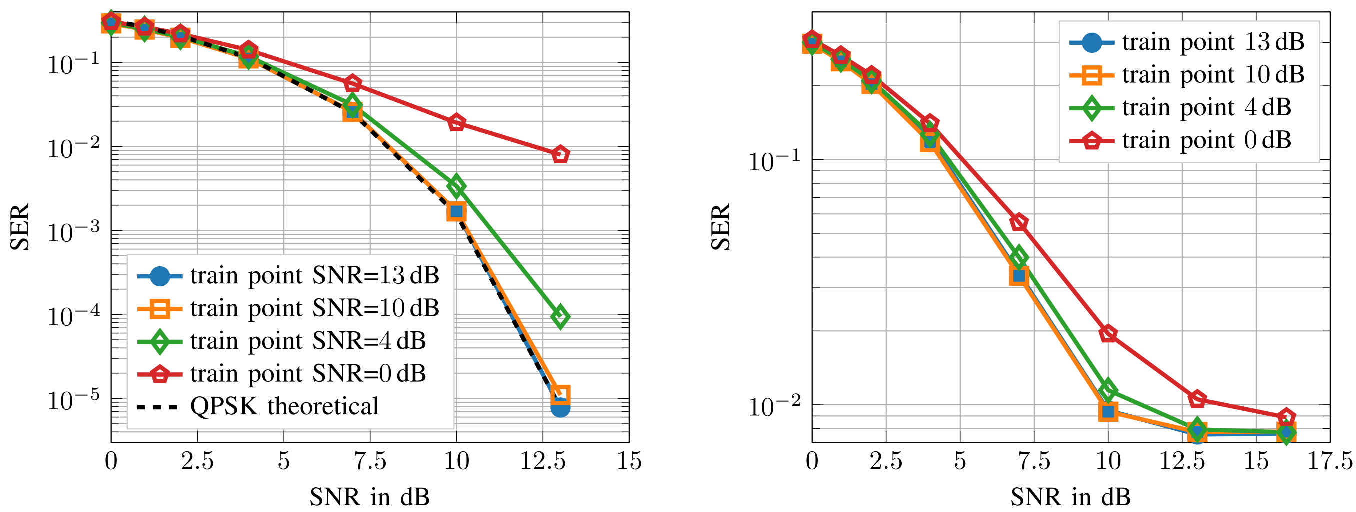
Similar to the case of error-free forwarding, the general findings for 3-bit quantization hold true. Explicitly, it is observed that the high SNR training points yield the best performance. It is further observed that, for 5-bit quantization, the green curve (i.e., 4 dB) now comes very close to the performances of high SNR training points. The lowest SNR training point (i.e., 0 dB) in red shows improved overall performance, while still falling short of the performances of other training points.

On the whole, from the 5-bit quantization results, we can conclude that more quantization clusters help the generalization performance for lower SNR training points, especially, in the case of error-prone forwarding. As expected, for high SNR values, the performance difference is negligible, as we already achieved the best performance (i.e., optimal demapping) with the 3-bit quantization.

As the main takeaway, we clearly showed the (excellent) generalization capabilities of Deep FAVIB, especially, for high SNR training points. This gives a clear evidence to the fact that, Deep FAVIB can be considered as a practically efficient data-driven solution to be exploited in (highly) dynamic and challenging environments.

V. Summary

In this paper, we considered a generic two-hop transmission setup, where a user / source is connected to the sink through a relay node. For efficient transmission, the relay node performs quantization, before forwarding its signal to the sink through an *error-prone* and *rate-limited* channel. The impacts of this *imperfect* forwarding must be integrated in the design of the compression scheme. Following the *Information Bottleneck* design framework, a data-driven solution, the *Deep FAVIB*, was introduced next which eliminates the need for the prior knowledge of the joint input statistics. This widens the extent of target applications as samples are more widely available. As



(a) error-free forwarding ($e = 0$)

(b) error-prone forwarding ($e = 0.01$)

Fig. 5. SER versus SNR for different noise training points for a) an error-free forward channel ($e = 0$) and b) an error-prone forward channel ($e = 0.01$), with $N = 32$ clusters and $\lambda = 0.01$.

the main contribution, here, by performing several numerical investigations, we further showed the excellent generalization capabilities of Deep FAVIB. Explicitly, we showed that training only once on a certain SNR point suffices to get excellent performance on a (wide) range of SNR values, indicating that Deep FAVIB is a practically efficient solution to be applied in challenging and (highly) dynamic environments.

Acknowledgement

This work was partly funded by the German ministry of education and research (BMBF) under grants 16KISK016 (Open6GHub) and 16KISK068 (6G-TakeOff).

References

[1] M. Hummert, S. Hassanpour, D. Wübben, and A. Dekorsy, "Deep FAVIB: Deep Learning-Based Forward-Aware Quantization via Information Bottleneck Method," in *2024 IEEE International Conference on Communications (ICC)*, Denver, CO, USA, June 2024.

[2] ——, "Deep Learning-Based Forward-Aware Quantization for Satellite-Aided Communications via Information Bottleneck Method," in *2024 European Conference on Networks and Communications and 6G Summit (EuCNC and 6G Summit)*, Antwerp, Belgium, June 2024.

[3] D. Wübben, P. Rost, J. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis, "Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible Centralization through Cloud-RAN," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 35–44, November 2014.

[4] S.-H. Park, O. Simeone, O. Sahin, and S. Shamai (Shitz), "Fronthaul Compression for Cloud Radio Access Networks: Signal Processing Advances Inspired by Network Information Theory," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 69–79, November 2014.

[5] B. Chen, L. Tong, and P. K. Varshney, "Channel-Aware Distributed Detection in Wireless Sensor Networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 16–26, July 2006.

[6] S. Movaghati and M. Ardakani, "Distributed Channel-Aware Quantization Based on Maximum Mutual Information," *International Journal on Distributed Sensor Networks*, vol. 12, no. 5, Art. no. 3595389, May 2016.

[7] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-Free Massive MIMO Versus Small Cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, March 2017.

[8] E. Björnson and L. Sanguinetti, "Scalable Cell-Free Massive MIMO Systems," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4247–4261, July 2020.

[9] M. Bashar, P. Xiao, R. Tafazolli, K. Cumanan, A. G. Burr, and E. Björnson, "Limited-Fronthaul Cell-Free Massive MIMO With Local MMSE Receiver Under Rician Fading and Phase Shifts," *IEEE Wireless Communications Letters*, vol. 10, no. 9, pp. 1934–1938, September 2021.

[10] A. Danaee, S. Hassanpour, D. Wübben, and A. Dekorsy, "Relevance-Based Information Processing for Fronthaul Rate Reduction in Cell-Free MIMO Systems," in *2024 International Symposium on Wireless Communication Systems (ISWCS)*, Rio de Janeiro, Brazil, July 2024.

[11] S. Hassanpour, T. Monsees, D. Wübben, and A. Dekorsy, "Forward-Aware Information Bottleneck-Based Vector Quantization for Noisy Channels," *IEEE Transactions on Communications*, vol. 68, no. 12, pp. 7911–7926, December 2020.

[12] N. Tishby, F. C. Pereira, and W. Bialek, "The Information Bottleneck Method," in *37th Annual Allerton Conference on Communication, Control, and Computing,* Monticello, IL, USA, September 1999.

[13] A. Zaidi, I. Estella-Aguerri, and S. Shamai (Shitz), "On the Information Bottleneck Problems: Models, Connections, Applications and Information Theoretic Views," *Entropy*, vol. 22, no. 2, Art. no. 151, January 2020.

[14] S. Hu, Z. Lou, X. Yan, and Y. Ye, "A Survey on Information Bottleneck," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5325–5344, August 2024.

[15] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv 2013*, arXiv:1312.6114.

[16] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Factor Analysis, Probabilistic Principal Component Analysis, Variational Inference, and Variational Autoencoder: Tutorial and Survey," *arXiv 2021*, arXiv:2101.00734.

[17] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep Variational Information Bottleneck," in *International Conference on Learning Representations*, Toulon, France, April 2017.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, 2006.

[19] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

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

[21] E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization with Gumbel-Softmax," *arXiv 2016*, arXiv:1611.01144.

[22] C. J. Maddison, A. Mnih, and Y. W. Teh, "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables," *arXiv 2016*, arXiv:1611.00712.

[23] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv 2017*, arXiv:1412.6980.