Deep Learning-Based Forward-Aware Quantization for Satellite-Aided Communications via Information Bottleneck Method

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 two-hop transmission setup in the context of Non-Terrestrial Networks (NTNs). Explicitly, a noisy source signal should be compressed at an on-ground relay node before getting forwarded over an *error-prone* and *rate-limited* channel to a satellite transponder. The impacts of this *imperfect* forwarding should be integrated into the compressor's design formulation. In full harmony with the *Information Bottleneck (IB)* principle, we choose the *Mutual Information (MI)* as the fidelity criterion and devise a data-driven algorithm, the *Deep Forward-Aware Vector Information Bottleneck (Deep FAVIB)*, to tackle the design problem, when solely a finite sample set is available. To this end, first we derive a tractable objective function and, later on, utilize it to train the encoder and decoder Deep Neural Networks (DNNs) in the introduced learning architecture. Our approach here, that is based on (generative) latent variable models, extends the well-known concepts of *Variational Auto-Encoders (VAEs)* and *Deep Variational Information Bottleneck (Deep VIB)* from remote source coding to joint source-channel coding. To corroborate the effectiveness of our data-driven approach, we also present several numerical results over a typical transmission scenario for NTNs.**

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

I. Introduction

Non-Terrestrial Networks (NTNs) are a promising system architecture for further extension of future communication systems. These networks make use of satellites and drones to enable global connectivity, especially for low-populated areas on earth. One possible use-case is the Internet of Things (IoT) as mass connectivity is a key aspect. Besides, NTNs can help to handle massive data rates in the future.

Herein, as a generic scenario, we consider a satellite-aided communication, illustrated in Fig. 1. A User Equipment (UE) transmits its signal over an access channel to an on-ground relay node. This node then *compresses* its recieved signal and *forwards* it over an *error-prone* and *rate-limited* channel to a satellite transponder for further processing. To compress the data, the Information Bottleneck (IB) framework [1] is used. It builds upon ideas from the seminal work of Shannon [2] on *lossy* source coding. In most practical quantization cases it is a lot easier to define a *relevant / target* variable, whose information shall be kept, than specifying a suitable distortion measure. With this in mind, IB modifies the *single-letter* characterization of the *Rate-Distortion* function by lower-bounding the mutual information between the target and compressed variable instead of upper-bounding the average distortion term. Conceptual ideas, linked to the practical applications of the IB method have been discussed in [3], [4]. The footprints of this *variational principle* can be found in a broad variety of applications, including the Analog-to-Digital converter design [5], Polar code construction [6], [7], discrete channel decoding [8]–[13], and in semantic communications [14]–[17].

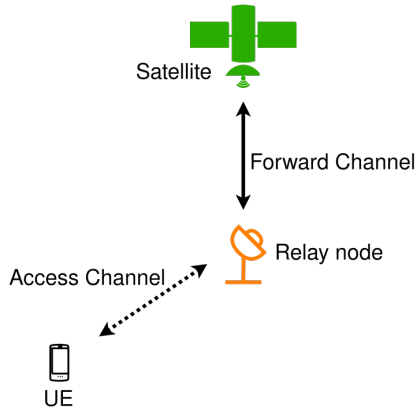

Fig. 1. A satellite-aided communication system with a relaying aspect. A noisy signal from a UE is received by an on-ground relay node, compressed and finally forwarded to a satellite transponder via an error-prone link.

With the massive growth of artificial intelligence and deep learning, Neural Networks (NNs) and data-driven algorithms are very widespread. By taking advantage of the potentials of NNs, sample-based approaches for optimizing the IB method have been derived [18]–[20]. These approaches do not require the prior full statistical knowledge of the system. Instead they operate based on a finite sample set of the input signals. These powerful extensions to the IB method enable its usage in more complex setups when only samples are available. On top, these methods can operate on high-dimensional data efficiently.

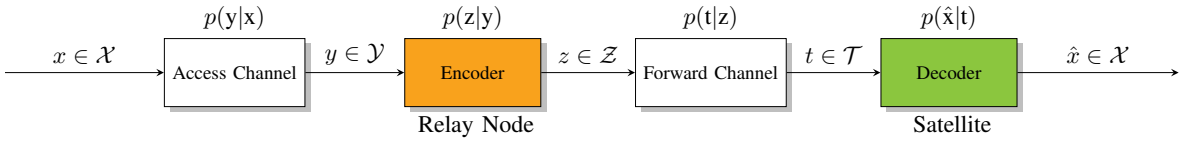

Fig. 2. System model: Source symbols x are sent over an access channel, clustered by an encoder, imperfectly forwarded, and reconstructed by a decoder $\hat{x}$.

Motivated by this, in this paper, we present the Deep FAVIB, a sample-based approach built upon the *Forward-Aware Vector Information Bottleneck (FAVIB)* algorithm [21]. Deep FAVIB directly generalizes the concepts of *Variational Auto-Encoder (VAE)* [22], [23] and *Deep Variational Information Bottleneck (Deep VIB)* [18] to the context of joint source-channel coding. More specifically, VAEs encode a source signal of interest into a latent variable (of typically much lower dimension) before decoding it back to the source signal space. Extending upon this idea, Deep VIB [18] encodes a *noisy* observation of the source signal into a latent variable. This forms a remote source coding scheme. Deep FAVIB further extends these ideas into the context of joint source-channel coding, by integrating the impacts of an *error-prone* foward channel (FC) into the *joint* training of the encoder and decoder DNNs.

We provide several simulation results for a generic satellite-aided communication setup, clearly highlighting the potentials of Deep FAVIB to be deployed in NTNs by demonstrating that it performs on par with the current State-of-the-Art (SotA) scheme, without relying on prior full statistical knowledge of the input signals. Our results indicate that Deep FAVIB is a promising alternative to the conventional algorithm, specially in applications where the joint statistics of input signals are not available (or easy to estimate).

II. System Overview

A. System Model and Problem Formulation

Fig. 2 illustrates the general system. A data source produces discrete-valued modulated symbols $x \in \mathcal{X}$, getting transmitted over an access channel $p(\mathrm{y}|\mathrm{x})$. This channel adds distortion to the source signal, yielding the received signal $y \in \mathcal{Y}$. Clustering is applied via a quantizer $p(\mathrm{z}|\mathrm{y})$ to compress the received signal y to $z \in \mathcal{Z}$ with fixed cardinality $|\mathcal{Z}| = N$. Following that, an *error-prone* and *rate-limited* forward channel $p(\mathrm{t}|\mathrm{z})$ with capacity R adds further distortions to the compressed signal, yielding the signal $t \in \mathcal{T}$. Afterwards, a decoder $p(\hat{\mathrm{x}}|\mathrm{t})$ must reconstruct the source signal $\hat{x}$. This overall system yields a joint source-channel coding scheme, as the encoder $p(\mathrm{z}|\mathrm{y})$ is designed such that, through the compression, it also takes the effects of imperfect forwarding into account. Aside from NTNs, this general setup can be found in numerous terrestrial applications as well, ranging from inference sensor networks with imperfect channels to the fusion center [24], [25] to Cloud-based Radio Access Networks (C-RANs) [26], [27] and Cell-Free massive Multiple Input Multiple Output (CFmMIMO) system [28]–[30] with non-ideal fronthaul links.

B. Conventional IB Method

To start the technical discussion, we construct the quantizer $p(\mathrm{z}|\mathrm{y})$ through the conventional IB method. In order to do so, the underlying distributions of the system need to be known. This assumption is only valid for the conventional approach. Principally, the goal is to preserve in the compressed signal z as much *information* as possible (the quantizer input signal y carries) about the source signal x.

1) Perfect Forwarding: First we consider the simpler case of error-free forwarding. With the IB method, we maximize the *relevant information* $I(\mathrm{x};\mathrm{z})$, with the constraint on the *compression rate* $I(\mathrm{y};\mathrm{z})$ being limited to the capacity R of the FC. Formalising this yields

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

with $0 \leq R \leq \log_2 |\mathcal{Z}|$ bits being the upper-limit to $I(\mathrm{y};\mathrm{z})$. By using the *Lagrange Method of Multipliers* [31] we get

$$p^{\star}(\mathrm{z}|\mathrm{y}) = \underset{p(\mathrm{z}|\mathrm{y})}{\operatorname{argmax}} I(\mathrm{x};\mathrm{z}) - \lambda I(\mathrm{y};\mathrm{z}) \,, \tag{2}$$

where $\lambda \geq 0$ is associated with the rate-limit R of the FC. Eq. (2) establishes a trade-off between reconstruction $I(\mathrm{x};\mathrm{z})$ and compression $I(\mathrm{y};\mathrm{z})$. Minimizing $I(\mathrm{y};\mathrm{z})$ works towards compressing the signal y, while maximizing $I(\mathrm{x};\mathrm{z})$ works towards reconstructing the source x. Basically, we want the compressed variable z to preserve the information about the source x, while being compressive w.r.t the noisy signal y.

The stationary solution of (2) has been derived in [1] as

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

for each pair $(y,z) \in \mathcal{Y}\times\mathcal{Z}$, where $\omega(y,\lambda)$ is a normalization function, to ensure a correct conditional distribution. This solution is the core of an iterative algorithm proposed in [1] to address the (non-convex) design problem (1). In essence, this algorithm performs the *Fixed-Point Iterations* [32] on the *implicit* solution (3). The resulting quantizer is typically *soft*, as long as $\lambda > 0$.

2) Imperfect Forwarding: For the more complicated case of imperfect forwarding the design problem is mathematically formulated as

$$p^{\star}(\mathrm{z}|\mathrm{y}) = \underset{p(\mathrm{z}|\mathrm{y}):\, I(\mathrm{y};\mathrm{z}) \leq R}{\operatorname{argmax}} I(\mathrm{x};\mathrm{t}) \,. \tag{4}$$

Again using the *Lagrange Method of Multipliers* [31] yields

$$p^{\star}(\mathrm{z}|\mathrm{y}) = \underset{p(\mathrm{z}|\mathrm{y})}{\operatorname{argmax}} \underbrace{I(\mathrm{x};\mathrm{t}) - \lambda I(\mathrm{y};\mathrm{z})}_{\mathcal{L}_{\text{FAVIB}}} \tag{5}$$

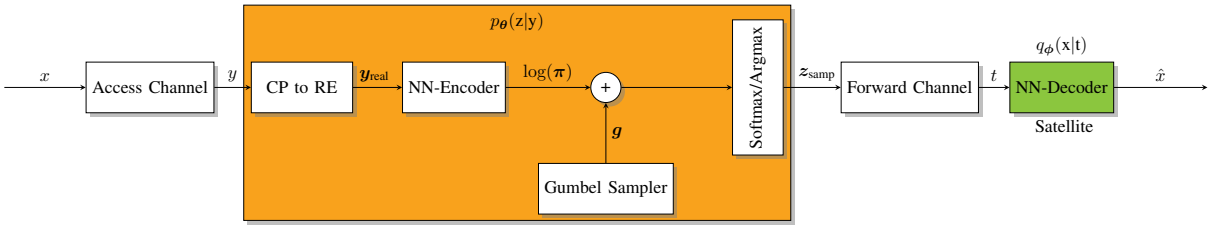

Fig. 3. Detailed Deep FAVIB learning architecture, consisting of two DNNs for encoder and decoder, a Gumbel sampler, and a softmax/argmax unit.

for each $(y, z) \in \mathcal{Y}\times\mathcal{Z}$. The key difference is that the quantizer is redesigned such that the information about the source signal x is mostly preserved at the output t of the error-prone forward channel. Solving (5) yields another stationary solution, derived in [21] as

$$p^{\star}(z|y) = \frac{p(z)}{\omega(y,\lambda)} \exp\Big(-\lambda^{-1}\sum_{t\in\mathcal{T}} p(t|z) D_{\mathrm{KL}}\big(p(\mathrm{x}|y)||p(\mathrm{x}|t)\big)\Big), \tag{6}$$

where $\omega(y,\lambda)$ is again a normalization function. Like before, this appears at the core of an iterative algorithm, namely, the *Forward-Aware (Vector) Information Bottleneck (FAVIB)*, proposed in [21]. Analogous to the previous case, for $\lambda > 0$, typically, a *soft* quantizer is achieved.

III. Deep FAVIB

In this section, we present our data-driven approach, namely, Deep FAVIB. It approximately tackles the design problem (5), when only samples instead of the full statistics are available. This approach generalizes the *Deep Variational Information Bottleneck (Deep VIB)* [18], which is based upon well-known generative latent variable models, specifically, the *Variational Auto-Encoders (VAEs)* [22], [23]

A. Finding the Variational Lower-Bound

We start by introducing a tractable *Variational Lower-Bound (VLB)* on the objective function $\mathcal{L}_{\text{FAVIB}}$ in (5). To that purpose, variables A and B are introduced to represent reconstruction and compression, respectively. By letting B be an upper-bound on the compression rate, i.e. $I(\mathrm{y};\mathrm{z}) \leq B$ and similarly letting A be a lower-bound on the relevant information $I(\mathrm{x};\mathrm{t})$, i.e., $I(\mathrm{x};\mathrm{t}) \geq A$, we can follow

$$\mathcal{L}^{\text{FAVIB}} = I(\mathrm{x};\mathrm{t}) - \lambda I(\mathrm{y};\mathrm{z}) \geq A - \lambda B = \mathcal{L}^{\text{VLB}}\,. \tag{7}$$

For reconstruction, it is true that

$$I(\mathrm{x};\mathrm{t}) = \underbrace{H(\mathrm{x})}_{\geq 0} - H(\mathrm{x}|\mathrm{t}) \tag{8a}$$

$$\geq \underbrace{\sum_{t\in\mathcal{T}} p(t) D_{\mathrm{KL}}\big(p(\mathrm{x}|t)||q(\mathrm{x}|t)\big)}_{\geq 0} + \sum_{x\in\mathcal{X},t\in\mathcal{T}} p(x,t)\log q(x|t) \tag{8b}$$

$$\geq E_{\mathrm{x,t}}\{\log q(\mathrm{x}|\mathrm{t})\} = A, \tag{8c}$$

wherein the proxy posterior $q(\mathrm{x}|\mathrm{t})$ is deployed to replace the perfect decoder $p(\mathrm{x}|\mathrm{t})$. Similarly, for the compression the following holds true

$$I(\mathrm{y};\mathrm{z}) = \sum_{y\in\mathcal{Y},\, z\in\mathcal{Z}} p(y,z)\log\frac{p(z|y)}{r(z)} - \underbrace{D_{\mathrm{KL}}\big(p(\mathrm{z})\|r(\mathrm{z})\big)}_{\geq 0} \tag{9a}$$

$$\leq \mathbb{E}_{\mathrm{y,z}}\Big\{\log\frac{p(\mathrm{z}|\mathrm{y})}{r(\mathrm{z})}\Big\} = B, \tag{9b}$$

where we introduce $r(\mathrm{z})$ as an arbitrary prior for z. Combining these findings, we get the following for the VLB

$$\mathcal{L}^{\text{VLB}} = E_{\mathrm{x,t}\sim p(\mathrm{x,t})}\{\log q(\mathrm{x}|\mathrm{t})\} - \lambda E_{\mathrm{y,z}\sim p(\mathrm{y,z})}\left\{\log\frac{p(\mathrm{z}|\mathrm{y})}{r(\mathrm{z})}\right\}. \tag{10}$$

Finally, in order to optimize the approximate decoder $q(\mathrm{x}|\mathrm{t})$ and the quantizer $p(\mathrm{z}|\mathrm{y})$, we make use of distributions which are parameterized and realised via NNs. By that, we get

$$\begin{aligned}\mathcal{L}^{\text{DNN}} &= E_{\mathrm{x,t}\sim p(\mathrm{x,t})}\{\log q_{\boldsymbol{\phi}}(\mathrm{x}|\mathrm{t})\} - \lambda E_{\mathrm{y,z}\sim p(\mathrm{y,z})}\left\{\log\frac{p_{\boldsymbol{\theta}}(\mathrm{z}|\mathrm{y})}{r_{\boldsymbol{\psi}}(\mathrm{z})}\right\} \\ &= \underbrace{E_{\mathrm{t}\sim p(\mathrm{t})}\big\{E_{\mathrm{x}\sim p(\mathrm{x}|\mathrm{t})}\{\log q_{\boldsymbol{\phi}}(\mathrm{x}|\mathrm{t})\}\big\}}_{\text{reconstruction}} \\ &\qquad - \underbrace{\lambda E_{\mathrm{y}\sim p(\mathrm{y})}\big\{D_{\mathrm{KL}}\big(p_{\boldsymbol{\theta}}(\mathrm{z}|\mathrm{y})||r_{\boldsymbol{\psi}}(\mathrm{z})\big)\big\}}_{\text{regularization}},\end{aligned} \tag{11}$$

with weights $\boldsymbol{\psi}$, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. λ is a trade-off parameter balancing the reconstruction and compression. Remember that we aim at maximizing (11). To maximize the relevant information $I(\mathrm{x};\mathrm{t})$, the *cross-entropy* loss, averaged over t, is minimized. This loss follows the *Maximum-Likelihood* learning rule [33]) and is the popular loss for classification. For the compression, a regularization w.r.t. the prior $r_{\boldsymbol{\psi}}(\mathrm{z})$ is present as the quantizer $p_{\boldsymbol{\theta}}(\mathrm{z}|\mathrm{y})$ is related to it via Kullback-Leibler Divergence (KLD), averaged over y.

B. NN Architecture and Implementation Details

In order to use NNs to design a stochastic encoder $p_{\boldsymbol{\theta}}(\mathrm{z}|\mathrm{y})$, we make use of the reparametrization trick [22] to enable sampling and calculate gradients of $\mathcal{L}^{\text{DNN}}$. This trick makes the sampling approach independent of the gradient flow, as the NNs only influence the statistics of the underlying distribution. Furthermore, the *Gumbel-Softmax* trick [34], [35] is used, as we consider a discrete latent variable. This trick yields a soft approximation to a discrete distribution, enabling the gradients to be calculated. The architecture has been shown in Fig 3.

The signal $y \in \mathcal{Y}$ at the output of the access channel is usually complex. To be used as an input for the NN-Encoder, y is stacked into a 2D vector, named $\boldsymbol{y}_{\text{real}}$, carrying the real and imaginary parts in each entry. This step is done to enable easy usage of NNs, as they cannot handle complex numbers straightforwardly since the gradient calculations can become tricky. The output of the NN-Encoder $\boldsymbol{\pi} \in (0,1)^N$, directly yields the categorical distribution of z. Sampling from the *concrete* variable is then done by generating samples from the Gumbel distribution $\boldsymbol{g} \in \mathbb{R}^N$. The combined signal $\log(\boldsymbol{\pi})+\boldsymbol{g}$ flows into a softmax/argmax unit. If argmax is applied, the *one-hot* encoding is performed, i.e. one entry is set to 1, while all other $N-1$ entries are set to 0. To enable training, argmax cannot be used, as gradients cannot be calculated, hence the softmax with a hyperparameter τ is used to approximate the argmax. This yields for the ith entry of $\boldsymbol{z}_{\text{samp}}$

$$z_{\text{samp},i} = \frac{\exp\Big(\big(\log(\pi_i)+g_i\big)/\tau\Big)}{\sum_{j=1}^{N}\exp\Big(\big(\log(\pi_j)+g_j\big)/\tau\Big)} \in [0,1], \qquad (12)$$

where $\tau > 0$. If τ is close to 0, the softmax becomes close to the argmax, and gradients change rapidly. If τ is large, on the other hand, the softmax gets smooth and may enable better optimization as gradients change slowly while flowing through. During the inference the argmax can be used to get a scalar z. The so created quantized z is then fed into the FC, introducing further signal distortions and the resulting t is handed over to the NN decoder $q_{\boldsymbol{\phi}}(\mathrm{x}|\mathrm{t})$ (with weights $\boldsymbol{\phi}$) for the source signal recovery. The decoder is an of-the-shelf Feed-Forward NN. This whole transmission chain can be seen as an extended VAE strucutre, where the input of the VAE structure is a *noisy* observation instead of the source signal itself. Furthermore, the latent variable is further disturbed by a relaying FC, before being fed to the decoder.

C. Supervised Learning and NNs

NNs are (nonlinear) functions with trainable parameters, in this context our encoder and decoder NNs with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, which are jointly trained w.r.t. a loss function, here $-\mathcal{L}^{\text{DNN}}$ (11). The parameters are updated by using a given data set $\{x_m, y_m\}_{m=1}^{M}$ and backpropagation utilizing a version of Stochastic Gradient Descent. For training the decoder, the output of the forward channel t_m is required. Samples t_m are indirectly created through the Markov chain from y_m to t_m. In this chain, the learned quantizer determines how z_m and afterwards t_m are generated. The prior $r_{\boldsymbol{\psi}}(z)$ has its own trainable parameters, although not being a NN.

IV. Numerical Results

For comparison, we use the SotA FAVIB as the baseline. As Deep FAVIB is closely related to FAVIB, its performance should, in ideal case, be on par with it. FAVIB needs the joint statistics of input signals $p(\mathrm{x},\mathrm{y})$, while Deep FAVIB runs on a sample set $\{x_m, y_m\}_{m=1}^{M}$, hence not requiring the knowledge of input signal statistics. FAVIB is executed 100 times and the best outcome is stored and presented. To enable easy comparison, we use an Additive White Gaussian Noise (AWGN) access channel with the noise variance σ_{n}^2. For the source signal, we apply the Quadrature Phase Shift Keying (QPSK) modulation. For the link from the on-ground relay node towards satellite (the FC), we use a discrete symmetric channel model with the error probability e. Specifically, each input symbol is received correctly with probability $1-e$, and erroneously (to any other symbols) with probability $\frac{e}{N-1}$, with N denoting the number of clusters. The variable parameters are N (number of output clusters of the encoder/quantizer), λ (trade-off in loss (11)), σ_{n}^2 (noise variance of access channel), τ (temperature to control the smoothness of the softmax (12)) and e (FC error probability), respectively. Deep FAVIB is only trained once for each parameter set, where a maximum of 50000 training epochs are conducted, using a batch size of 10000 and $M = 1\text{e}6$ samples. We make use of *Early Stopping* to save the weights with the lowest training loss. We use the Adam optimizer [36] with a learning rate of 10^{-5}. The number of neurons and layers have been provided in Table I. We use NNs with of-the-shelf Feed-Forward structures with 3 layers and the Rectified Linear Unit (ReLu) activation functions. The output layer of the NN-Encoder uses a linear activation function to construct the log-probabilities $\log(\boldsymbol{\pi})$. For the NN-Decoder the output layer uses a softmax activation function to distinguish the transmit symbols in $\mathcal{X}$. Training is conducted for each parameter set once. For testing, the weights are loaded and used for inference without retraining. Please note that we need to perform a training for each parameter set we want to test. and hence computational complexity of DEEP FAVIB will not be an issue for deployment.

TABLE I
SETUPS FOR PRIOR, DECODER NN AND ENCODER NN

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

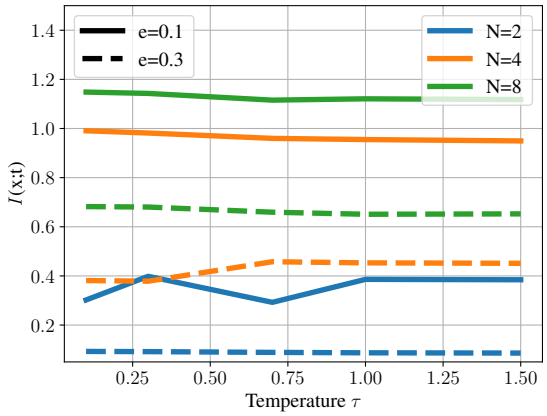

Fig. 4. Relevant information $I(\mathrm{x};\mathrm{t})$ versus temperature τ for different number of output clusters N and error probabilities e, for $\sigma_{\text{n}}^2 = 0.4$ and $\lambda = 0.01$.

A. Temperature τ

Since the temperature τ is an important hyper-parameter for Deep FAVIB, first we investigate its effect on the obtained performance for different number of output clusters N and forward error probabilities e. In Fig. 4 we show the relevant information $I(\mathrm{x};\mathrm{t})$ versus τ, when varying both N and e, for a fixed access noise variance, namely, $\sigma_{\mathrm{n}}^2 = 0.4$. Further, we set $\lambda = 0.01$, i.e., we fully focus on the information preservation.

From Fig. 4, it is observed that, for $e = 0.1$ and $N = 2$, the obtained relevant information fluctuates over τ. By increasing N, it is seen that the overall performance behaves in a more stable fashion w.r.t. τ. Specifically, it is observed that for $N = 4$ and $N = 8$, the obtained relevant information shows a decreasing trend, when increasing the temperature τ. Focusing on the depicted results for a higher forward error probability, namely, $e = 0.3$, it is observed that, for $N = 2$, contrary to the previous case, the obtained performance remains steady throughout the whole range of τ. For $N = 4$, it is seen that the relevant information shows an increasing trend up to $\tau = 0.75$ and flattens afterwards. Contrarily, for $N = 8$, a decreasing trend is observed in the obtained performance by increasing the temperature τ. As the main takeaway, one shall realize that the temperature τ has to be chosen quite carefully depending on the scenario, since the overall performance of Deep FAVIB heavily depends on it. So, for the other performance plots, we always select the best τ (which, most often, is $\tau = 0.1$).

B. End-to-End Transmission Rate vs. Number of Clusters

As the second numerical investigation, in Fig. 5, we present the relevant information / end-to-end transmission rate $I(\mathrm{x};\mathrm{t})$ when varying the (allowed) number of output clusters N, for various forward error probabilities e. For the benchmark, we also plot the SotA FAVIB results. Here again, we fix the access noise variance to $\sigma_{\mathrm{n}}^2 = 0.4$ and by choosing $\lambda = 0.01$, we fully focus on the information preservation.

From Fig. 5, it is observed that, expectedly, the performance improves through loosening the compression bottleneck or, in other words, by increasing the number of output clusters N. Furthermore, it is directly observed that by degrading the FC quality or, in other words, by increasing the forward error probability e, the end-to-end transmission rate decreases. This is again expected, as the end-to-end transmission rate is upper-bounded by the capacity of forward channel. Important to note is the fact that, while Deep FAVIB runs only on samples of input signals instead of requiring the full statistical knowledge, its performance comes on par with FAVIB, regardless of the chosen parameter set.

C. End-to-End Transmission Rate vs. Access Noise Variance

As the final numerical investigation, in Fig. 6, we present the relevant information / end-to-end transmission rate $I(\mathrm{x};\mathrm{t})$ when varying the access channel noise variances σ_{n}^2. Since the transmit power is fixed, this directly translates into varying the Signal-to-Noise Ratio (SNR). Here as well, we set the trade-off parameter to $\lambda = 0.01$ (fully concentrating on information preservation). Moreover, we fix the number of output clusters

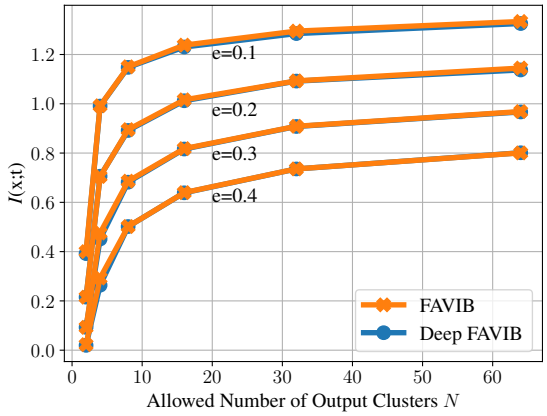

Fig. 5. Relevant information $I(\mathrm{x};\mathrm{t})$ versus allowed number of output clusters N for different error probabilities e of FC, with $\sigma_{\mathrm{n}}^2 = 0.4$ and $\lambda = 0.01$.

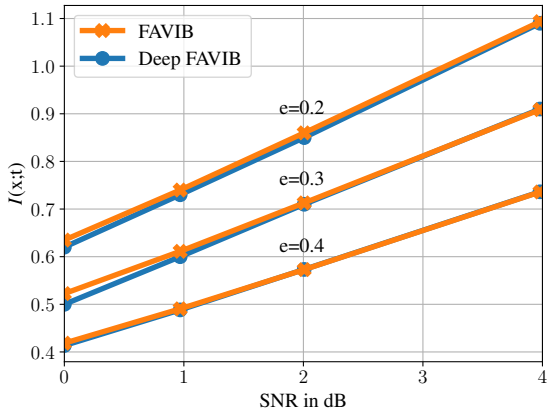

Fig. 6. Relevant information $I(\mathrm{x};\mathrm{t})$ versus SNR in dB of AWGN access channel for different error probabilities e, with $\lambda = 0.01$ and $N = 32$ clusters.

to $N = 32$ and consider three different forward error probabilities, namely, $e = 0.2, 0.3, 0.4$.

From Fig. 6, it is observed that, the lower the SNR gets, the lower the end-to-end transmission rate becomes. This again is an expected behavior, because the worse the access channel conditions become, the less information can be transported through the system, noting that the end-to-end transmission rate is also upper-bounded by the capacity of access channel. Totally aligned with the previous results, the same holds true for the forward channel. More specifically, the larger the forward error probability, the lower the obtainable end-to-end transmission rate becomes. Here again, it is clearly observed that Deep FAVIB performs (almost) on par with FAVIB over the entire range of model parameters. This vividly shows the promising performance of Deep FAVIB. Explicitly, a single training round for Deep FAVIB (per parameter set) is sufficient to yield the best result out of 100 reruns of the SotA FAVIB.

As the main takeaway, from the presented results in this section, it is immediately inferred that the Deep FAVIB can be deployed as a promising alternative to conventional (iterative) algorithms, specially, in scenarios where the joint statistics of input signals are not available (or easy to estimate).

V. Summary

In this paper, we focused on a two-hop NTN transmission scenario in which a UE is connected to a satellite transponder through an on-ground relay node. For efficient transmission, this node should perform data compression, before forwarding its signal to the satellite over an *error-prone* and *rate-limited* channel. The impacts of this *imperfect* forwarding should be integrated in the design of the compression scheme. To address the introduced design problem based on a finite sample set of input variables, we presented a deep learning-based approach, the Deep FAVIB, as the counterpart of SotA FAVIB algorithm [21]. Deep FAVIB, as a latent variable model, extends the well-known concepts of *Variation Auto-Encoders (VAEs)* [22] and *Deep Variational Information Bottleneck* [18] from (indirect) source coding to the context of joint source-channel coding. By several numerical results, we corroborated the effectiveness of this sample-based algorithm and highlighted its potential as an alternative to the conventional approaches. Promising for future work are applications where the joint statistics of input signals are not available or easy to estimate.

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

[2] C. E. Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion," *IRE Int. Convention Record,* part 4, vol. 7, pp. 142–163, March 1959.

[3] 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.

[4] Z. Goldfeld and Y. Polyanskiy, "The Information Bottleneck Problem and Its Applications in Machine Learning," *IEEE Jrnl. on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 19–38, May 2020.

[5] G. Zeitler, A. C. Singer, and G. Kramer, "Low-Precision A/D Conversion for Maximum Information Rate in Channels with Memory," *IEEE Trnsact. on Comms.*, vol. 60, no. 9, pp. 2511–2521, September 2012.

[6] I. Tal and A. Vardy, "How to Construct Polar Codes," *IEEE Trnsact. on Information Theory*, vol. 59, no. 10, pp. 6562–6582, October 2013.

[7] M. Stark, A. Shah, and G. Bauch, "Polar Code Construction Using the Information Bottleneck Method," in *IEEE Wireless Comms. and Networking Conf. Workshops*, Barcelona, Spain, April 2018.

[8] F. J. C. Romero and B. M. Kurkoski, "LDPC Decoding Mappings That Maximize Mutual Information," *IEEE Jrnl. on Selected Areas in Comms.*, vol. 34, no. 9, pp. 2391–2401, September 2016.

[9] J. Lewandowsky and G. Bauch, "Information-Optimum LDPC Decoders Based on the Information Bottleneck Method," *IEEE Access*, vol. 6, pp. 4054–4071, 2018.

[10] M. Stark, J. Lewandowsky, and G. Bauch, "Information-Optimum LDPC Decoders with Message Alignment for Irregular Codes," in *IEEE Global Comms. Conf.*, Abu Dhabi, UAE, December 2018.

[11] M. Stark, L. Wang, G. Bauch, and R. D. Wesel, "Decoding Rate-Compatible 5G-LDPC Codes With Coarse Quantization Using the Information Bottleneck Method," *IEEE Open Jrnl. of the Comms. Society*, vol. 1, pp. 646–660, May 2020.

[12] T. Monsees, O. Griebel, M. Herrmann, D. Wübben, A. Dekorsy, and N. Wehn, "Minimum-Integer Computation Finite Alphabet Message Passing Decoder: From Theory to Decoder Implementations towards 1 Tb/s," *Entropy*, vol. 24, no. 10, Art. no. 19, October 2022.

[13] T. Monsees, D. Wübben, A. Dekorsy, O. Griebel, M. Herrmann, and N. Wehn, "Finite-Alphabet Message Passing Using Only Integer Operations for Highly Parallel LDPC Decoders," in *IEEE 23rd Int. Workshop on Signal Proc. Advances in Wireless Comms.*, July 2022.

[14] J. Shao, Y. Mao, and J. Zhang, "Learning Task-Oriented Comm. for Edge Inference: An Information Bottleneck Approach," *IEEE Jrnl. on Selected Areas in Comms.*, vol. 40, no. 1, pp. 197–211, January 2022.

[15] F. Pezone, S. Barbarossa, and P. Di Lorenzo, "Goal-Oriented Communication for Edge Learning Based on the Information Bottleneck," in *IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Singapore, Singapore, May 2022.

[16] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond Transmitting Bits: Context, Semantics, and Task-Oriented Communnications," *IEEE Jrnl. on Selected Areas in Comms.*, vol. 41, no. 1, pp. 5–41, January 2023.

[17] E. Beck, C. Bockelmann, and A. Dekorsy, "Semantic Information Recovery in Wireless Networks," *Sensors*, vol. 23, no. 14, Art. no. 6347, July 2023.

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

[19] A. Zaidi and I. Estella-Aguerri, "Distributed Deep Variational Information Bottleneck," in *IEEE Int. Workshop on Signal Proc. Advances in Wireless Comms.*, Atlanta, GA, USA, May 2020.

[20] Q. Wang, C. Boudreau, Q. Luo, P.-N. Tan, and J. Zhou, "Deep Multi-View Information Bottleneck," in *SIAM Int. Conf. on Data Mining*, Calgary, Alberta, Canada, May 2019.

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

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

[23] 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.

[24] 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, 2006.

[25] S. Movaghati and M. Ardakani, "Distributed Channel-Aware Quantization Based on Maximum Mutual Information," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 5, Art. no. 3595389, May 2016.

[26] 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.

[27] 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.

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

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

[30] 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 Comms. Letters*, vol. 10, no. 9, pp. 1934–1938, September 2021.

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

[32] J. H. Mathews and K. D. Fink, *Numerical Methods Using MATLAB*, 4th ed. Pearson Prentice Hall, 2004.

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

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

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

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