Deep Learning-Based Distributed Remote Source Coding via Information Bottleneck Method: The Parallel Processing Scheme

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

***Abstract*—We focus on a generic multiterminal remote source coding scenario, appearing in a variety of real-world applications. Specifically, several *noisy* observations from a remote user/source signal should be quantized at some intermediate nodes prior to a forward transmission via multiple *error-free* and *rate-limited* links to a processing unit. To design the local quantizers, we follow the *Information Bottleneck* method, and devise a purely data-driven solution, which can be categorized as a *Latent Variable Model* in the context of generative AI. To that end, we derive a tractable *variational lower-bound* of the original objective functional, and present the pertinent *learning architecture*, over which, the design problem can be addressed by the joint training of the encoder DNNs and the decoder DNN, e.g., by some form of the *Stochastic Gradient Descent*. By several numerical investigations, we further show that this data-driven compression scheme performs (almost) on par with the SotA model-based approach, without requiring the prior knowledge of the (joint) statistics of input signals. This becomes quite important, especially, in those applications where the joint input statistics are either unavailable or hard to estimate.**

***Index Terms*—6G, deep learning, distributed data compression, generative AI, information bottleneck, remote source coding.**

I. Introduction

The *Information Bottleneck (IB)* method, that first introduced in [1], was developed as an information-theoretic approach for data compression. The main idea behind was to retain as much information as possible about a *target / relevant* variable, when compressing a (correlated) data set. The design problem was formulated by modifying the *single-letter* characterization of the *Rate-Distortion (RD)* function in Shannon's seminal work on the *lossy* source coding [2]. Later, it was shown that (see, e.g., [3]) this modified formulation characterizes the boundary of achievable RD region for a *remote source coding* problem with the *Logarithmic Loss* distortion. We refer the interested readers to [4] for a more detailed discussion on various aspects of this *variational framework* from the standpoints of both the Information and Learning theory.

From a practical perspective, it is also noteworthy that the IB method has already been exploited in the design of various parts of modern communication systems. Those applications include (but are not limited to) the Analog-to-Digital (A / D) converters [5], the (efficient) construction of *Polar Codes* [6], the discrete channel decoding schemes [7], and recently in the Semantic / Task-Oriented Communications [8].

In this work, we focus on a multiterminal extension of the original IB setup, wherein by following a *joint* design, several *noisy* observations from a source signal should be compressed at some intermediate nodes, ahead of a forward transmission via multiple *error-free* and *rate-limited* channels to a (remote) processing unit. This (very generic) setup appears in a broad variety of real-world applications, from distributed inference sensor networks with *rate-limited* links to the fusion center, to the Cloud-based Radio Access Networks (Cloud-RANs) as well as (User Centric) massive Multiple-Input Multiple-Output (UC-mMIMO) systems with *rate-limited* fronthaul links, and in relaying schemes with the *compress-and-forward* strategy.

As the main contribution, by leveraging the framework of *Latent Variable Models (LVMs)* in the context of *Generative Artificial Intelligence (GenAI)*, we present here a purely data-driven distributed IB-based compression scheme that extends the well-known concepts of *Variational Auto-Encoders (VAEs)* [9], [10] and the *Deep Variational Information Bottleneck* [11], while performing on par with the SotA model-based scheme. The practical importance of such a sample-based compression approach reveals itself in applications where the joint statistics of input variables are either unavailable or hard to estimate.

Outline: We start our discussion by a brief presentation of the point-to-point IB-based remote source coding in Section II as a prelude towards the distributed extension. In Section III, the system model and the corresponding design problem are presented for the distributed IB-based compression approach. This will be followed by a concise presentation of the SotA model-based solution in Section IV. Thereupon, in Section V, we present our data-driven approach to address the challenging design problem purely based on a finite sample set. To further substantiate the effectiveness of this novel data-driven method, we present some numerical investigations in Section VI, prior to a short wrap-up in Section VII, containing the salient points.

Notations: For the discrete random variable, a, each sample, $a \in \mathcal{A}$, happens according to the probability distribution, $p(\mathsf{a})$. The same applies to the random vector, $\mathbf{a}_{1:J} = \{\mathsf{a}_1, \cdots, \mathsf{a}_J\}$, with the boldface counterparts. Moreover, $\mathbf{a}_{1:J}^{\bar{j}} = \mathbf{a}_{1:J} \setminus \{\mathsf{a}_j\}$, and, $\mathbb{E}_{\bullet}\{\cdot\}$, represents the expectation operator. $H(\cdot)$, $D_{\mathrm{KL}}(\cdot\|\cdot)$, and, $I(\cdot;\cdot)$, denote Shannon's entropy, *Kullback-Leibler (KL)* divergence, and Mutual Information [12], respectively. Finally, $\{\cdot\}_{j=1}^{J}$ represents a set of J elements.

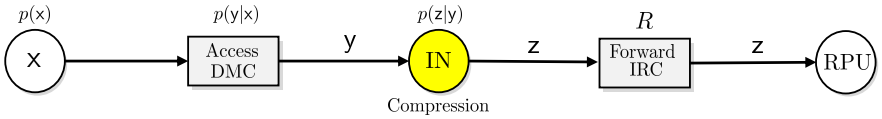


Fig. 1. The considered system model for point-to-point remote source coding. DMC, IN, IRC, and RPU stand for Discrete Memoryless Channel, Intermediate Node, Ideal Rate-limited Channel, and Remote Processing Unit, respectively. The Markov chain, $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$, applies.

II. Point-to-Point IB-Based Remote Source Coding

Consider the presented system model in Fig. 1. Explicitly, a *noisy* observation, y, from the user / source signal, x, should be compressed at an Intermediate Node (IN) to the signal, z, before getting forwarded via an Ideal Rate-limited Channel (IRC) to a Remote Processing Unit (RPU). The interrelation between the source signal, x, and the *noisy* observation, y, is modeled through a Discrete Memoryless Channel (DMC). It is further presumed that the source statistics, $p(\mathsf{x})$, as well as the transition probabilities, $p(\mathsf{y}|\mathsf{x})$, are given, and the Markov chain, $\mathsf{x} \leftrightarrow \mathsf{y} \leftrightarrow \mathsf{z}$, applies.

The goal is to design the compressor, $p(\mathsf{z}|\mathsf{y})$, in a fashion that, while satisfying a rate constraint, the compressed signal, z, retains as much information as possible about the source signal, x. Mathematically, the design problem is formulated as a constrained optimization in which, the *relevant information*, $I(\mathsf{x};\mathsf{z})$, is maximized such that the *compression rate*, $I(\mathsf{y};\mathsf{z})$, does not exceed the capacity / rate-limit, R, of the forward link. Explicitly, the following holds

$$p^*(\mathsf{z}|\mathsf{y}) = \underset{p(\mathsf{z}|\mathsf{y})\colon I(\mathsf{y};\mathsf{z}) \le R}{\operatorname{argmax}} \; I(\mathsf{x};\mathsf{z}), \tag{1}$$

in which, $0 \le R \le \log_2 |\mathcal{Z}|$ bits, sets an upper-bound on the compression rate, $I(\mathsf{y};\mathsf{z})$. Exploiting the method of *Lagrange multipliers* [13], the design problem (1) can then be recast into the following unconstrained optimization (up to the validity of the conditional compressor mapping)

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

where $\lambda \ge 0$ is associated with the forward channel capacity, R. The form of stationary solution of the design problem (2) has been derived in [1] (for each pair $(y,z) \in \mathcal{Y} \times \mathcal{Z}$) as

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

where $\beta = \frac{1}{\lambda}$, and $\omega(y,\beta)$ is a normalization function to ensure the validity of the pertinent conditional quantizer mapping. An iterative algorithm has also been presented in [1] to address the design problem (2), by performing the *Fixed-Point Iterations* [14] on the derived *implicit* solution (3).

III. Distributed Extension: System Model & Problem Formulation

Consider the illustrated system model in Fig. 2. A source signal, x, is observed imperfectly at J INs. These nodes should then (locally) compress their *noisy* observations, y_j for $j = 1$ to J, into the signals, z_j for $j = 1$ to J, before forwarding them via J *error-free* and *rate-limited* channels with the capacities, R_j for $j = 1$ to J, to an RPU. The interrelation between the source signal, x, and the j-th noisy observation, y_j, is modeled by a DMC whose transition probabilities, $p(\mathsf{y}_j|\mathsf{x})$, as well as its input statistics, $p(\mathsf{x})$, are presumed to be known. Later on, when we present our data-driven approach, we will relax this presumption by requiring solely a (finite) sample set from the joint input statistics, $p(\mathsf{x}, \mathbf{y}_{1:J})$, instead. Also, we assume that, given the source signal, x, the counterpart signals of different INs are statistically independent. By following the *Information Bottleneck (IB)* principle, we formulate the design problem of the set of (local) compressors, $P = \{p(\mathsf{z}_1|\mathsf{y}_1), \cdots, p(\mathsf{z}_J|\mathsf{y}_J)\}$, as a constrained optimization wherein the goal is to maximize the *relevant information*, i.e., $I(\mathsf{x};\mathbf{z}_{1:J})$, without exceeding the capacities, R_j for $j = 1$ to J, of the individual forward links. Mathematically, it holds

$$P^* = \underset{P\colon \forall j \; I(\mathsf{y}_j;\mathsf{z}_j) \le R_j}{\operatorname{argmax}} \; I(\mathsf{x};\mathbf{z}_{1:J}), \tag{4}$$

with $0 \le R_j \le \log_2 |\mathcal{Z}_j|$ bits, upper-bounding the *compression rate*, $I(\mathsf{y}_j;\mathsf{z}_j)$, of the j-th IN. It must be mentioned that, this design formulation corresponds to a purely *parallel processing* at RPU to retrieve the source signal, x. Interested readers are referred to [15] for a successive processing flow / strategy at RPU to retrieve the source signal.

IV. Model-Based Approach in a Nutshell

By application of the method of *Lagrange multipliers* [13], we can recast the design problem (4) into an unconstrained maximization (up to a validity condition regarding the involved conditional distributions), namely,

$$P^* = \underset{P}{\operatorname{argmax}} \; I(\mathsf{x};\mathbf{z}_{1:J}) - \sum_{j=1}^{J} \lambda_j I(\mathsf{y}_j;\mathsf{z}_j), \tag{5}$$

where $\lambda_j \ge 0$ is associated with the capacity, R_j, in the original formulation. For a given R_j, the respective λ_j can be found, e.g., via an iterative bi-section search. The stationary solution of the design problem (5) for the j-th compressor mapping, $p(\mathsf{z}_j|\mathsf{y}_j)$, has been derived in [16] (for each $(y_j, z_j) \in \mathcal{Y}_j \times \mathcal{Z}_j$) as

$$p(z_j|y_j) = \frac{p(z_j)}{\omega_{\mathsf{z}_j}(y_j,\beta_j)} \exp\Big(-\beta_j\, d(y_j,z_j)\Big), \tag{6}$$

where $\beta_j = \frac{1}{\lambda_j}$, and $\omega_{\mathsf{z}_j}(y_j,\beta_j)$ is a partition function to ensure the validity of the pertinent conditional distribution. Moreover, the multivariate relevant distortion, $d(y_j,z_j)$, is calculated as

$$d(y_j,z_j) = \sum_{\boldsymbol{z}^{-j}_{1:J}} p(\boldsymbol{z}^{-j}_{1:J}|y_j) D_{\mathrm{KL}}\big(p(\mathsf{x}|y_j,\boldsymbol{z}^{-j}_{1:J}) \| p(\mathsf{x}|z_j,\boldsymbol{z}^{-j}_{1:J})\big)\,. \tag{7}$$

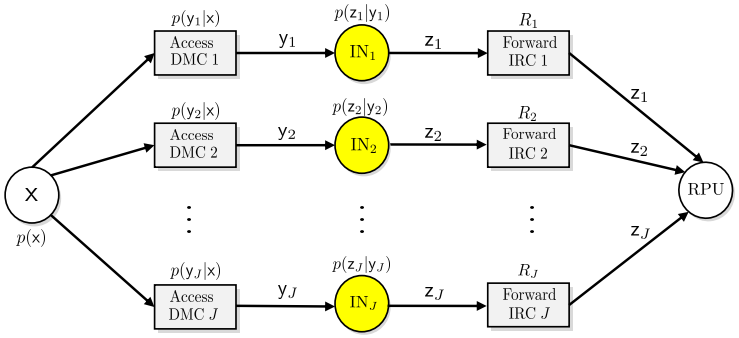


Fig. 2. The considered system model for distributed remote source coding. Given source, x, the counterpart signals of different branches are independent.

Taking a closer look at the derived relevant distortion in (7), it is realized that it quantifies the degree of proximity between the conditional distributions in which $y_j \in \mathcal{Y}_j$ is involved with those wherein y_j is replaced by its compressed representative, $z_j \in \mathcal{Z}_j$. In other word, the probability of allocating $y_j \in \mathcal{Y}_j$ to any bin / cluster, $z_j \in \mathcal{Z}_j$, is commensurate with how well that z_j represents y_j. An iterative algorithm has also been presented in [16] to address the design problem (5) based on the derived stationary solutions in (6). This algorithm, termed *Multivariate iterative IB (MultiIB)*, in core, runs the so-called *Multivariate Fixed-Point Iterations* [14] on the *implicit* solutions (6).

For fixed $p(\mathsf{z}_m|\mathsf{y}_m)$ and finite λ_m ($m=1$ to J and $m \neq j$), by letting $\lambda_j \to 0$, a *deterministic* quantizer mapping, $p(\mathsf{z}_j|\mathsf{y}_j)$, is obtained, corresponding to the state of *full concentration*. For finite λ_j, usually a *stochastic* quantizer mapping, $p(\mathsf{z}_j|\mathsf{y}_j)$, is generated, while in the case of letting $\lambda_j \to \infty$, the state of *full diffusion* is achieved wherein each $y_j \in \mathcal{Y}_j$ is allocated to all output clusters, $z_j \in \mathcal{Z}_j$, equiprobably.

V. Novel Deep Learning-Based Approach

In this section, we present our data-driven approach, termed *Deep MultiIB*, to address the design problem (5) based upon a (finite) sample set from the joint statistics, $p(\mathsf{x}, \mathbf{y}_{1:J})$, of the source and the *noisy* observations. This novel approach, which can be categorized under the umbrella of the (generative) latent variable models, directly extends some well-known concepts, including the *Deep Variational Information Bottleneck* [11] and the *Variational Auto-Encoders* [9], [10].

A. Variational Lower-Bound

The first step towards developing a sample-based distributed data compression scheme is to introduce a tractable *Variational Lower-Bound (VLB)* on the objective Lagrangian in (5), i.e.,

$$\mathcal{L} = I(\mathsf{x}; \mathbf{z}_{1:J}) - \sum_{j=1}^{J} \lambda_j I(\mathsf{y}_j; \mathsf{z}_j) \; . \tag{8}$$

To that end, first we find a global lower-bound on the relevant information, $I(\mathsf{x}; \mathbf{z}_{1:J})$, by defining an *auxiliary* distribution, $q(\mathsf{x}|\mathbf{z}_{1:J})$, for the joint decoder. Explicitly, it holds

$$I(\mathsf{x}; \mathbf{z}_{1:J}) = \underbrace{H(\mathsf{x})}_{\geq 0} - H(\mathsf{x}|\mathbf{z}_{1:J}) \tag{9a}$$

$$\geq \underbrace{\sum_{\boldsymbol{z}_{1:J}} p(\boldsymbol{z}_{1:J}) D_{\mathrm{KL}}\big(p(\mathsf{x}|\boldsymbol{z}_{1:J}) \| q(\mathsf{x}|\boldsymbol{z}_{1:J})\big)}_{\geq 0} + \sum_{x,\, \boldsymbol{z}_{1:J}} p(x, \boldsymbol{z}_{1:J}) \log q(x|\boldsymbol{z}_{1:J}) \tag{9b}$$

$$\geq \mathbb{E}_{\mathsf{x}, \mathbf{z}_{1:J}} \{ \log q(\mathsf{x}|\mathbf{z}_{1:J}) \} \; . \tag{9c}$$

From (9a) to (9b), the non-negativity of entropy / uncertainty (for the discrete source signal, x), has been applied. Further, from (9b) to (9c), the non-negativity of KL divergence (a.k.a. the *information inequality*) [12] has been applied.

Next, we find a global upper-bound on the j-th compression rate, $I(\mathsf{y}_j; \mathsf{z}_j)$, by defining *auxiliary* distributions, $\{r(\mathsf{z}_j)\}_{j=1}^{J}$, for the latent priors. Explicitly, it holds

$$I(\mathsf{y}_j; \mathsf{z}_j) = \sum_{y_j \in \mathcal{Y}_j,\, z_j \in \mathcal{Z}_j} p(y_j, z_j) \log \frac{p(z_j|y_j)}{r(z_j)} - \underbrace{D_{\mathrm{KL}}\big(p(\mathsf{z}_j) \| r(\mathsf{z}_j)\big)}_{\geq 0} \tag{10a}$$

$$\leq \mathbb{E}_{\mathsf{y}_j, \mathsf{z}_j} \big\{ \log \frac{p(\mathsf{z}_j|\mathsf{y}_j)}{r(\mathsf{z}_j)} \big\} = \sum_{y_j \in \mathcal{Y}_j} p(y_j) D_{\mathrm{KL}}\big(p(\mathsf{z}_j|y_j) \| r(\mathsf{z}_j)\big), \tag{10b}$$

wherein, from (10a) to (10b), the information inequality has been applied. Now, we are in the position to finally define the VLB for the design Lagrangian, $\mathcal{L}$, in (8) as

$$\mathcal{L}^{\mathrm{VLB}} = \mathbb{E}_{\mathsf{x}, \mathbf{z}_{1:J}} \{ \log q(\mathsf{x}|\mathbf{z}_{1:J}) \} - \sum_{j=1}^{J} \lambda_j \mathbb{E}_{\mathsf{y}_j, \mathsf{z}_j} \big\{ \log \frac{p(\mathsf{z}_j|\mathsf{y}_j)}{r(\mathsf{z}_j)} \big\}, \tag{11}$$

since from (9) and (10), it is directly deduced that

$$\mathcal{L}\big(\{p(\mathsf{z}_j|\mathsf{y}_j)\}_{j=1}^{J}\big) \geq \max_{q, \{r\}} \mathcal{L}^{\mathrm{VLB}}\big(\{p(\mathsf{z}_j|\mathsf{y}_j)\}_{j=1}^{J}, q(\mathsf{x}|\mathbf{z}_{1:J}), \{r(\mathsf{z}_j)\}_{j=1}^{J}\big), \tag{12}$$

and, consequently,

$$\max_{P=\{p\}} \mathcal{L}\big(\{p(\mathsf{z}_j|\mathsf{y}_j)\}_{j=1}^{J}\big) \geq \max_{P=\{p\}} \max_{q, \{r\}} \mathcal{L}^{\mathrm{VLB}}\big(\{p(\mathsf{z}_j|\mathsf{y}_j)\}_{j=1}^{J}, q(\mathsf{x}|\mathbf{z}_{1:J}), \{r(\mathsf{z}_j)\}_{j=1}^{J}\big) \; . \tag{13}$$

Next, we consider a parameterized family for all input arguments of the derived lower-bound, $\mathcal{L}^{\mathrm{VLB}}$. Explicitly, we denote by $\{\boldsymbol{\theta}_j\}_{j=1}^{J}$, $\boldsymbol{\phi}$, and $\{\boldsymbol{\psi}_j\}_{j=1}^{J}$, the parameter sets for families of distributions regarding the encoders, $\{p(\mathsf{z}_j|\mathsf{y}_j)\}_{j=1}^{J}$,

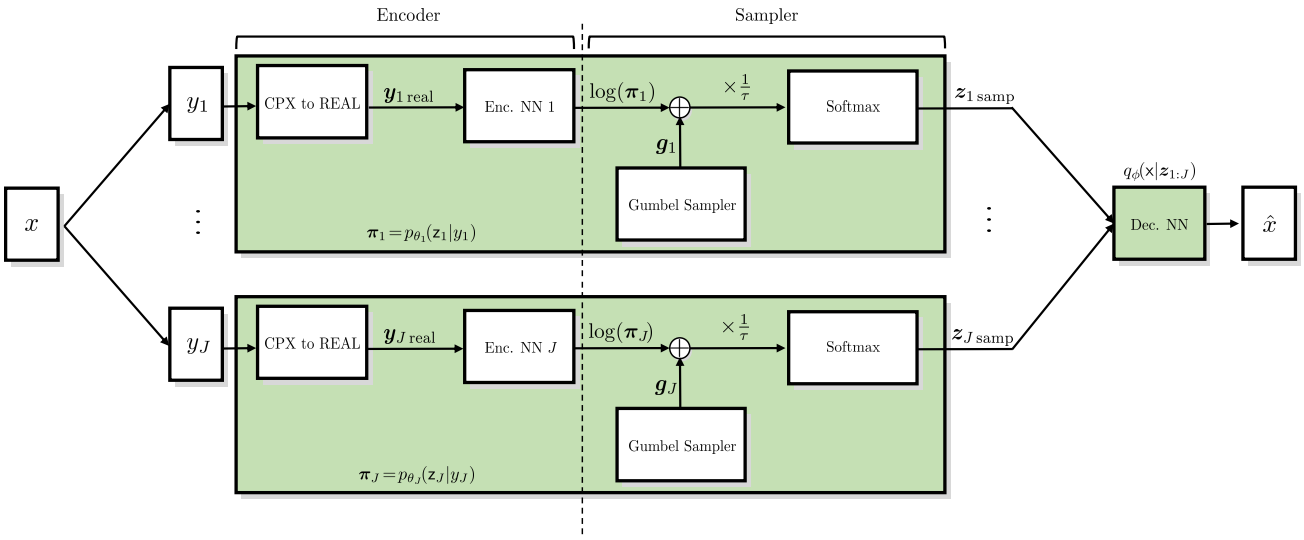


Fig. 3. The introduced learning architecture for distributed IB-based remote source coding scheme, featuring J local encoder DNNs and a joint decoder DNN.

the joint decoder, $q(\mathsf{x}|\mathbf{z}_{1:J})$, and the latent priors, $\{r(\mathsf{z}_j)\}_{j=1}^J$, and by $\mathcal{L}^{\text{DNN}}$, the introduced VLB with the parameterized input distributions, namely,

$$\mathcal{L}^{\text{DNN}} = \mathcal{L}^{\text{VLB}}\big(\{p_{\boldsymbol{\theta}_j}(\mathsf{z}_j|\mathsf{y}_j)\}_{j=1}^J, q_{\boldsymbol{\phi}}(\mathsf{x}|\mathbf{z}_{1:J}), \{r_{\boldsymbol{\psi}_j}(\mathsf{z}_j)\}_{j=1}^J\big)\,. \tag{14}$$

Then, the following inequality holds

$$\max_{\{p\}}\max_{q,\{r\}} \mathcal{L}^{\text{VLB}}\big(\{p(\mathsf{z}_j|\mathsf{y}_j)\}_{j=1}^J, q(\mathsf{x}|\mathbf{z}_{1:J}), \{r(\mathsf{z}_j)\}_{j=1}^J\big) \geq \max_{\{\boldsymbol{\theta}_j\},\boldsymbol{\phi},\{\boldsymbol{\psi}_j\}} \mathcal{L}^{\text{DNN}}\big(\{\boldsymbol{\theta}_j\}_{j=1}^J, \boldsymbol{\phi}, \{\boldsymbol{\psi}_j\}_{j=1}^J\big), \tag{15}$$

as the search space over valid distributions, $\{p\}$, q, $\{r\}$, will be restricted to the hypothesis space of respective parameterized families. A closer look at the derived VLB reveals that $\mathcal{L}^{\text{DNN}}$ consists of two separate terms, one for the *reconstruction*, and another for the *regularization*. Explicitly, it applies

$$\begin{aligned}\mathcal{L}^{\text{DNN}} &= E_{\mathsf{x},\mathbf{z}_{1:J}\sim p(\mathsf{x},\mathbf{z}_{1:J})}\{\log q_{\boldsymbol{\phi}}(\mathsf{x}|\mathbf{z}_{1:J})\} \\ &\quad - \sum_{j=1}^J \lambda_j E_{\mathsf{y}_j,\mathsf{z}_j\sim p(\mathsf{y}_j,\mathsf{z}_j)}\left\{\log\frac{p_{\boldsymbol{\theta}_j}(\mathsf{z}_j|\mathsf{y}_j)}{r_{\boldsymbol{\psi}_j}(\mathsf{z}_j)}\right\} \\ &= \underbrace{E_{\mathbf{z}_{1:J}\sim p(\mathbf{z}_{1:J})}\{E_{\mathsf{x}\sim p(\mathsf{x}|\mathbf{z}_{1:J})}\{\log q_{\boldsymbol{\phi}}(\mathsf{x}|\mathbf{z}_{1:J})\}\}}_{\text{reconstruction}} \\ &\quad - \sum_{j=1}^J \lambda_j \underbrace{E_{\mathsf{y}_j\sim p(\mathsf{y}_j)}\{D_{\text{KL}}(p_{\boldsymbol{\theta}_j}(\mathsf{z}_j|\mathsf{y}_j)||r_{\boldsymbol{\psi}_j}(\mathsf{z}_j))\}}_{\text{regularization}}\,.\end{aligned} \tag{16}$$

It is straightly seen that maximizing the relevant information, $I(\mathsf{x};\mathbf{z}_{1:J})$, corresponds to minimizing the *cross-entropy* loss (that is, principally, the reconstruction loss for classification, when following the *Maximum-Likelihood* learning rule [17]), averaged over $\mathbf{z}_{1:J}$. On the other hand, the counterpart term for the compression rates, $I(\mathsf{y}_j;\mathsf{z}_j)$ for $j=1$ to J, acts as a regularizer since every local compressor, $p_{\boldsymbol{\theta}_j}(\mathsf{z}_j|\mathsf{y}_j)$ for $j=1$ to J, should match the corresponding latent prior, $r_{\boldsymbol{\psi}_j}(\mathsf{z}_j)$, via a KL divergence term, averaged over y_j.

Finally, it must also be mentioned that the derived objective function in (16) extends the *Evidence Lower-BOund (ELBO)* that is used to train the *Variational Auto-Encoders (VAEs)* [9], [10] when the Evidence itself (that is the *Maximum Likelihood* objective function) becomes intractable.

B. Learning Architecture & Implementation Details

Generally, we intend to design the *stochastic* local encoders, $p_{\boldsymbol{\theta}_j}(\mathsf{z}_j|\mathsf{y}_j)$ for $j=1$ to J, and the joint decoder, $q_{\boldsymbol{\phi}}(\mathsf{x}|\mathbf{z}_{1:J})$, via Deep Neural Networks (DNNs). To estimate the gradients of the derived objective function, $\mathcal{L}^{\text{DNN}}$, we can simply resort to the conventional approach of exploiting the *reparameterization trick*. Subsequently, by performing the *Monte-Carlo* sampling, we can then replace the expectation terms with their empirical estimates. Since the focus here is on *discrete* latent spaces, we can use the so-called *Gumbel-Softmax / Concrete Distribution* [18], [19] to do the trick for us, that is, to reparameterize the underlying categorical distributions. Having all these points in mind, we present the learning architecture of our data-driven distributed IB-based remote source coding approach, the *Deep MultiIB*, in Fig. 3.

Generally, the *noisy* observation, $y_j \in \mathcal{Y}_j$ for $j=1$ to J, is complex-valued. So, to be fed into the j-th local encoder DNN, the signal, $y_j \in \mathcal{Y}_j$, is stacked into a two-dimensional vector, $\boldsymbol{y}_{j\,\text{real}} \in \mathbb{R}^2$, that contains the inphase and quadrature parts, separately. We do this since DNNs cannot handle the complex numbers straightforwardly in the training phase (as complex derivatives are not always straightforward to calculate). The j-th local encoder that is fed by $\boldsymbol{y}_{j\,\text{real}}$ outputs a categorical distribution, $\boldsymbol{\pi}_j \in (0,1)^N$, wherein N denotes the number of categories / bins for the discrete latent variable, z_j. To generate samples from the pertinent *concrete variable*, first we draw N independent and identically distributed (i.i.d.) samples from the Gumbel $(0,1)$ distribution and stack them into the vector,

$\boldsymbol{g}_j$ for $j=1$ to J. The sum signal $\log(\boldsymbol{\pi}_j)+\boldsymbol{g}_j$ is then multiplied by the inverse of a (positive) hyperparameter, τ, known as the *temperature* in the relevant literature. This scaled signal is then fed into a *Softmax* unit. Consequently, the i-th entry of the j-th sample vector, $\boldsymbol{z}_{j\,\text{samp}}$, is calculated as ($i=1$ to N)

$$z_{j\,\text{samp}}^{(i)} = \frac{\exp\Big(\big(\log(\pi_j^{(i)}) + g_j^{(i)}\big)/\tau\Big)}{\sum_{\ell=1}^{N}\exp\Big(\big(\log(\pi_j^{(\ell)}) + g_j^{(\ell)}\big)/\tau\Big)} \in [0,1], \quad (17)$$

where $\pi_j^{(i)}$ and $g_j^{(i)}$ are the i-th entries of the vectors, $\boldsymbol{\pi}_j$ and $\boldsymbol{g}_j$, respectively. The lower the τ becomes, the closer behavior to an *Argmax* is achieved. For $\tau=1$, the unmodified Softmax is achieved. By letting $\tau\to\infty$, a uniform distribution is obtained over N categories. Thus, the temperature, τ, must be chosen carefully, as either too small or too large values of it may lead to a poor performance (due to either experiencing a very rapid change in the gradients or an excessively smoothed behavior). Finally, the decoder is a standard Feed-Forward DNN.

The end-to-end chain that is illustrated in Fig. 3 extends the conventional structure of VAEs. Particularly, in a conventional VAE, first we go from the source, x, to a latent representation, z, by an encoder, and then we try to retrieve / recover the source from that latent representation by a decoder. In contrast, here, we go from *noisy* observations, y_j for $j=1$ to J, of the source, x, to the latent representations, z_j for $j=1$ to J, by J (local) encoders, and then try to recover the source, x, from all latent representations by a joint decoder.

C. Neural Networks & Supervised Learning

Neural Networks are nonlinear functions with the trainable parameters that are called weights, here $\boldsymbol{\theta}_j$ for $j=1$ to J and $\boldsymbol{\phi}$ for (local) encoders and the joint decoder, respectively. These weights are adapted w.r.t. a loss function, here $-\mathcal{L}^{\text{DNN}}$ in (16). Given a finite data set of the source signal realizations and the pertinent noisy observations, the weights of the encoder DNNs and the joint decoder DNN are updated by *back-propagation*, when performing some form of *Gradient Descent*. The goal is to *jointly* train the (local) encoders and the (joint) decoder by minimizing the loss function. Finally, it should be mentioned that the latent priors, $r_{\boldsymbol{\psi}_j}(\mathsf{z}_j)$ for $j=1$ to J, although having their own sets of parameters (i.e., the probabilities of different categories / clusters), are not implemented by DNNs.

VI. Numerical Results

We consider a standard, equiprobable 16-QAM (Quadrature Amplitude Modulation) source signaling ($\sigma_{\mathsf{x}}^2=10$) over $J=3$ AWGN (Additive White Gaussian Noise) access channels with the (same) noise variance, σ_{n}^2. To conduct the training, we use 10^6 samples, with a batch size of 10^4, and a maximum of 10^4 epochs. We apply the *Early Stopping* to obtain the best weights. Furthermore, we set the learning rate to 10^{-5}, and make use of the *Adam* optimizer [20]. The detailed configurations of the encoder DNNs and the (joint) decoder DNN have been given in Table I. Specifically, for every (local) encoder and the joint decoder, a *Multi-Layer Perceptron (MLP)* has been exploited with three hidden layers featuring the *ReLU (Rectified Linear Unit)* activation function. For (local) encoder DNNs, the output activation function is linear to form the log probabilities, i.e., $\log(\boldsymbol{\pi}_j)$. For the joint decoder, the output activation function is a Softmax on different symbols of the source alphabet.

TABLE I
The configuration of encoder DNNs, decoder DNN, and priors.

Denotation	# of Hidden Layers	Width of Layers	# of Weights
$p_{\boldsymbol{\theta}_j}(\mathsf{z}_j\|\mathsf{y}_j)$	3	300, 200, 100	81200+101×N
$q_{\boldsymbol{\phi}}(\mathsf{x}\|\mathsf{z}_{1:3})$	3	300, 200, 100	82216+900×N
$r_{\boldsymbol{\psi}_j}(\mathsf{z}_j)$	0	0	N

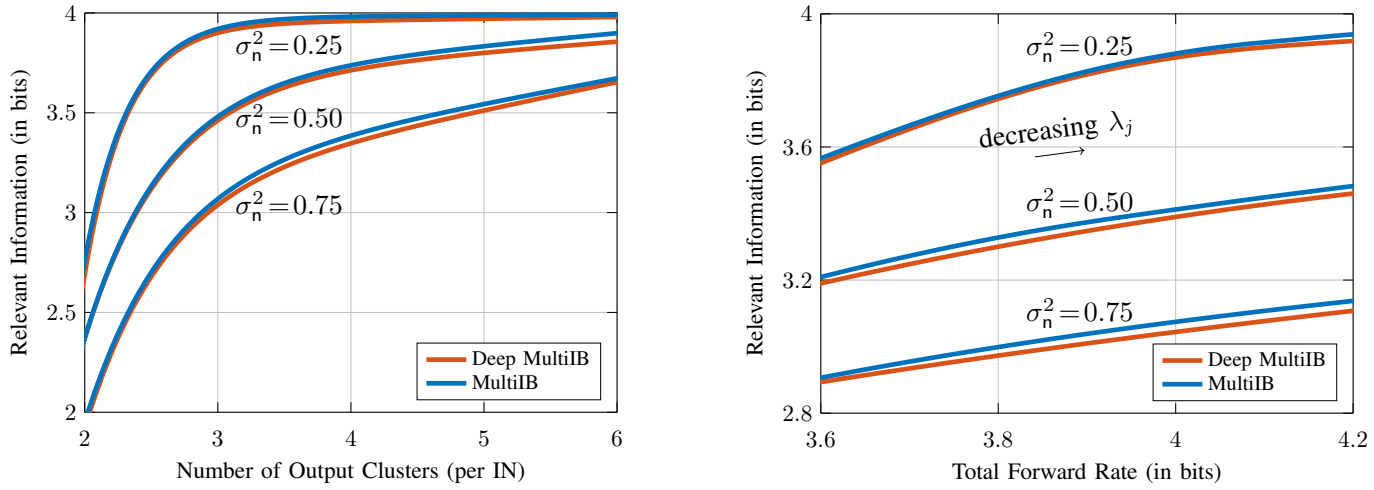
In Fig. 4, we present the relevant information, $I(\mathsf{x};\mathsf{z}_{1:3})$, by the model-based *MultiIB* and the data-driven *Deep MultiIB* vs. a) the cardinality of the compressors' output alphabet, N, and b) the total forward rate, $\sum_{j=1}^{3} I(\mathsf{y}_j;\mathsf{z}_j)$, for different values of the access links' noise variance, namely, $\sigma_{\mathsf{n}}^2=0.25, 0.50, 0.75$. To obtain the pertinent curves for MultiIB, the best outcome out of 100 trials (with different initialization) was chosen per parameter set. Regarding the Deep MultiIB, the training was conducted for each parameter set, and the weights were saved and used for inference without retraining.

Focusing on Fig. 4a, it is observed that the obtained relevant information, $I(\mathsf{x};\mathsf{z}_{1:3})$, increases by increasing the number of output clusters, N, per branch (for a given access links' noise variance, σ_{n}^2). This is an expected behavior, as by increasing N, we loosen the compression bottleneck, and consequently, we allow a larger flow of information throughout the system. Likewise, it is directly seen that for a given number of output clusters, N, the relevant information, $I(\mathsf{x};\mathsf{z}_{1:3})$, increases by decreasing the access links' noise variance, σ_{n}^2. This is also expected, as by deceasing σ_{n}^2, the capacities of access links increase. Therefore, larger information about the source signal, x, is flown into the system. As the main takeaway, it is readily observed that the data-driven Deep MultiIB, performs (almost) on par with the SotA model-based MultiIB, without requiring the full prior knowledge of the joint statistics, $p(\mathsf{x},\mathbf{y}_{1:3})$.

Focusing on Fig. 4b, analogous behaviors are observed as the ones already discussed for Fig. 4a, and similar justifications are equally applicable. Explicitly, by decreasing the trade-off parameter, λ_j for $j=1,2,3$, the focus is steered towards the preservation of the relevant information (at the cost of higher total forward rates). The main takeaway here as well is the fact that the overall "*information-compression*" dynamics of the SotA model-based MultiIB is obtainable by the devised data-driven Deep MultiIB, only with a (finite) sample set.

VII. Summary

In this work, we focused on a generic multiterminal remote source coding scenario which appears in a broad variety of real-world applications. Explicitly, multiple *noisy* observations from a user / source signal must be quantized at some intermediate nodes, before getting forwarded to a processing unit via several *error-free* and *rate-limited* links. To design the (local)



(a) Fixed trade-off value $\lambda_j = 0.01$

(b) Fixed output cardinality $N = 4$

Fig. 4. The relevant information of MultiIB and Deep MultiIB vs. a) number of output clusters and b) total forward rate. 16-QAM source signaling ($\sigma_x^2 = 10$) over AWGN access channels ($\sigma_n^2 = 0.25, 0.50, 0.75$), with a) $\lambda_j = 0.01$ for $j = 1, 2, 3$, and the temperature $\tau = 2$ and b) $N = 4$, $0.25 \leq \lambda_j \leq 0.9$ for $j = 1, 2, 3$, and the temperature $\tau = 0.5$.

compressors, we applied the *Information Bottleneck* principle and selected the *Mutual Information* as the fidelity criterion. After concisely discussing the SotA model-based solution, we presented our purely data-driven approach, the *Deep MultiIB*, that extends the well-known concepts from (generative) latent variable models, especially, the *Variational Auto-Encoders* [9], and *Deep Variational Information Bottleneck* [11]. Explicitly, we derived a tractable *variational lower-bound* on the original objective functional and presented a *learning architecture* over which the design problem is addressed by the joint training of the encoder DNNs and the decoder DNN. Through some numerical investigations, we further showed that our devised data-driven approach performs (almost) on par with the SotA model-based solution, without requiring the prior knowledge of the joint statistics of input signals. This clearly indicates the importance of Deep MultiIB, especially, in applications where the input statistics are either unavailable or hard to estimate.

Acknowledgment

This was partly funded by the German ministry of education and research (BMBF) under grants 16KISK068 (6G-TakeOff), 16KISK016 (Open6GHub), and 16KISK109 (6G-ANNA).

References

[1] 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, Sep. 1999.

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

[3] P. Harremoës and N. Tishby, "The Information Bottleneck Revisited or How to Choose a Good Distortion Measure," in *IEEE International Symposium on Information Theory*, Nice, France, Jun. 2007.

[4] 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, Jan. 2020.

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

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

[7] 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 Journal of the Communications Society*, vol. 1, pp. 646–660, May 2020.

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

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

[10] 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.

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

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

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

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

[15] S. Hassanpour, D. Wübben, and A. Dekorsy, "Generalized Distributed Information Bottleneck for Fronthaul Rate Reduction at the Cloud-RANs Uplink," in *IEEE Global Communications Conference*, Taipei, Taiwan, Dec. 2020.

[16] ——, "A Novel Approach to Distributed Quantization via Multivariate Information Bottleneck Method," in *IEEE Global Communications Conference*, Waikoloa, HI, USA, Dec. 2019.

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

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

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

[20] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv 2014*, arXiv:1412.6980.