Estimation of Rate-Distortion Function for Computing with Decoder Side Information

Heasung Kim, Hyeji Kim, and Gustavo de Veciana
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX, USA
Email: {heasung.kim, hyeji.kim, deveciana}@utexas.edu

***Abstract*—"THIS PAPER IS ELIGIBLE FOR THE STUDENT PAPER AWARD." There has been growing interest in computing rate-distortion functions for real-world data, as they can provide a theoretical benchmark for compression problems. However, a generalized form of rate-distortion that includes side information and coding for computing has been underexplored, despite its relevance in modern compression problems. To address this gap, we propose a new method for estimating the rate-distortion function for computing with side information, using a Lagrangian framework with neural network-parametrized encoding and decoding strategies. This approach enables targeting specific points on the rate-distortion curve through gradient-based optimization. Our methodology is validated in synthetic environments where rate-distortion functions are known, ensuring accuracy in estimation. Additionally, we extend its application to practical, high-dimensional channel state information compression scenarios. We provide rate-distortion estimation results on these scenarios, which in turn enables us to quantify the usefulness of side information in the practical scenarios.**

I. Introduction

The rate-distortion function [1], which characterizes the optimal rate-distortion trade-off, serves as a theoretical benchmark for assessing the effectiveness of compression algorithms, as highlighted in recent studies [2]–[5]. However, accurately computing Shannon's information measures, such as entropy and mutual information which form the basis of rate-distortion function, is notably challenging. This is particularly true in scenarios involving real-world distributions where one must rely solely on samples without any additional knowledge of the distributions, or in cases involving high-dimensional input sources. Closed-form solutions for these measures are generally limited to specific circumstances, e.g., Gaussian sources.

A. Computing rate-distortion functions

An approach to numerically compute the rate-distortion functions and associated information measures for general distributions has been devised based on iterative algorithms in 1972 by Blahut [6] and Arimoto [7]. Known collectively as the Blahut–Arimoto algorithms, they have been adapted to address multiterminal source coding settings [8].

However, these conventional iterative approaches face limitations, especially when applied to high-dimensional or continuous sources [3]. To overcome these challenges, recent studies have explored solutions utilizing neural networks or advanced optimization techniques. The Restricted Boltzmann Machines is integrated with neural networks to estimate rate-distortion functions [2]. In [3], rate-distortion function duality concepts, e.g., [9], is utilized in estimation methods. A sandwich bound for rate-distortion function is introduced in [4] through distribution parameterization with neural networks. Notably, the Wasserstein gradient descent algorithm proposed in [5], has demonstrated state-of-the-art performance for rate-distortion estimation, without relying on neural networks.

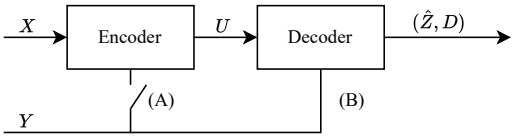

Fig. 1. Coding for Computing with Side Information. We consider a configuration where switch (A) remains open while switch (B) is closed, permitting access to side information at the decoder. The decoder aims to compute a function $Z = g(X, Y)$; we let $\hat{Z}$ and D denote the decoder's output and distortion, respectively.

B. Rate-distortion function for computing with side information

The rate-distortion function concept can be extended to encompass scenarios where side information, correlated with the input source, is available at the decoder, or at both the encoder and decoder, as illustrated in Figure 1. This adaptation is widely recognized as the Wyner-Ziv rate-distortion function [10]. Additionally, the notion of the rate-distortion is further broadened by considering communication systems where the goal is to compute a function of the source. Such applications of the Wyner-Ziv rate-distortion function are commonly referred to as *Coding for Computing* [11].

Such a broader perspective of the rate-distortion function is crucial for evaluating compression algorithms in practical scenarios and understanding side information's role in various contexts. It also offers a way to quantify the relevance of different types of side information for various sources.

While recent compression techniques increasingly incorporate side information [12]–[14] with specific objectives, research on rate-distortion estimation with side information, especially for continuous or large-dimensional distributions, remains limited. Prior studies have focused on discrete sources and side information [15], extending the Blahut-Arimoto Algorithm but often struggle with high-dimensional distributions

particularly when there is no a priori knowledge of the distributions.

Our contributions address these gaps through a neural network-based estimation method for the rate-distortion function for computing with side information, along with applicable methodologies:

C. Contributions

We present a generalized framework for estimating the rate-distortion function for computing with side information. We formulate a Lagrangian loss function where the minimization of this function is achieved through specific encoding and decoding schemes that can achieve point(s) on the rate-distortion function. Our algorithm focuses on minimizing the loss by parameterizing the conditional distributions of the codewords for a given source and side information, as well as the decoder. The algorithm is designed to alternatively update these parameters to efficiently minimize the Lagrangian loss.

The effectiveness of our algorithm is validated through numerical evaluation, particularly in cases where the rate-distortion function is known, such as with correlated Gaussian distributions for the source and side information. These simulations demonstrate that our algorithm consistently provides a precise estimation of the rate-distortion function. Extending beyond the synthetic data, we also apply our approach to practical scenarios. This includes assessing the possible gains when side information is available when considering channel state information compression problems and illustrating the practical relevance and applicability of our method to high-dimensional sources.

II. System Model and Proposed Method

Consider the communication system illustrated in Fig. 1, where switch (A) is open and (B) is closed. The primary source and side information pair (X, Y) is assumed to be independently and identically distributed (i.i.d.), following a joint distribution $p_{X,Y}(x, y)$. Here, x and y are realizations of X and Y from the domains $\mathcal{X}$ and $\mathcal{Y}$, respectively. The codeword is represented by U from the domain $\mathcal{U}$ and the output of the decoder is $\hat{Z} \in \mathcal{Z}$, with D denoting a distortion level and d being a distortion measure defined as a mapping as $d : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}^+$. For readability, we also let $\hat{X}$ denote the output of the decoder when the system aims to reconstruct the original information X.

In our general framework we assume the objective is to reconstruct a function $Z = g(X, Y)$, where Z is not necessarily identical to X. In this setting, the corresponding rate-distortion function determines the minimum necessary rate to compute $g(X, Y)$ within a given distortion threshold D. The rate-distortion function, denoted $R_{\text{D,C}}$, is given as follows [16].

Definition 1 (Rate-distortion function for computing with side information)**.**

$$R_{\text{D,C}}(D) = \min_{q_{U|X}(u|x), f(u,y):\ \mathbb{E}[d(Z,\hat{Z})] \le D} I(X; U|Y) \quad (1)$$

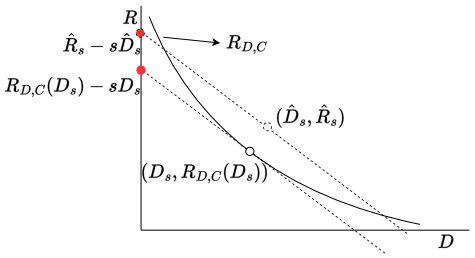

Fig. 2. $R_{\text{D,C}}$ is convex with respect to distortion D. For a given slope s, minimizing the y-intercept of a line originating from an achievable point $(\hat{D}_s, \hat{R}_s)$ in the rate-distortion region leads to a new y-intercept, which corresponds to a line that is tangent to the $R_{\text{D,C}}$ curve at point(s) with the same slope s.

where $q_{U|X}(u|x)$ is a conditional probability distribution of U given X. Z and $\hat{Z}$ are the desired function output and the decoder output, respectively. f is a decoder taking u and side information y as an input pair as $f(u, y) = \hat{z}$.

In this paper, we focus on developing a method to estimate $R_{\text{D,C}}(D)$ for a general function, particularly in scenarios where the joint distribution $p_{X,Y}(x, y)$ is unknown and a dataset of N data points $(x_i, y_i)_{i=1}^N$ that are sampled from $p_{X,Y}(x, y)$ is available. This setup is typical in real-world contexts, where the exact distribution underlying a dataset is often not known.

We start with a Lagrangian formulation to address the optimization problem defined in (1) by exploiting convexity and non-increasing property of $R_{\text{D,C}}(D)$ with respect to the distortion D. We can formulate an optimization problem for finding the vertical intercept of the tangent with slope $s(\le 0)$ to the rate-distortion curve as follows.

$$R_{\text{D,C}}(D_s) - sD_s = \min_{q_{U|X}, f} \{I(X; U|Y) - s\mathbb{E}[d(Z, \hat{Z})]\}. \quad (2)$$

For a given slope s and a corresponding achievable (distortion, rate) pair, $(\hat{D}_s, \hat{R}_s)$ illustrated in Fig. 2, the y-intercept at this line is $\hat{R}_s - s\hat{D}_s$. This intercept is equivalent to $I(X; U|Y) - s\mathbb{E}[d(Z, \hat{Z})]$, attained by the specific encoding and decoding schemes associated with $q_{U|X}$, f corresponding to $(\hat{D}_s, \hat{R}_s)$. This y-intercept can be minimized through optimization, adjusting the encoding and decoding schemes accordingly.

Due to the convexity of $R_{\text{D,C}}$, the lowest achievable value of the vertical intercept corresponds to $R_{\text{D,C}}(D_s) - sD_s$ where the distortion D_s and rate $R_{\text{D,C}}(D_s)$ is a point lies on the $R_{\text{D,C}}$ curve itself. By determining a point on the $R_{\text{D,C}}$ curve for each slope s and then varying s, we can estimate the $R_{\text{D,C}}$ curve.

To facilitate estimation using a given dataset, we reformulate the optimization term as follows.

$$\min_{q_{U|X}, f} \left\{ \mathbb{E}_{X,Y,U}\left[\log \frac{q_{U|X}(U|X)}{q_{U|Y}(U|Y)} \right] - s\mathbb{E}[d(Z, \hat{Z})] \right\}, \quad (3)$$

where $q_{U|Y}(u|y) = \sum_{x \in \mathcal{X}} p_{X|Y}(x|y) q_{U|X}(u|x)$ (when X is a discrete random variable) and $\hat{Z} = f(U, Y)$. This formulation enables the computation of expectation terms using Monte Carlo estimation with data points following the distribution $q_{X,Y,U}(x, y, u)$. Here, we use the notation q to represent a

Algorithm 1 Estimation of Rate-Distortion Function for Computing with Side Information at Decoder

1: **Input:** Slope s, dataset $\{x_i, y_i\}_{i=1}^N$, initialized sets of parameters $\boldsymbol{\theta}_{\text{po}}$, $\boldsymbol{\theta}_{\text{pr}}$, $\boldsymbol{\theta}_{\text{dec}}$
2: **for** $t = 0$ to T **do**
3: Sample minibatch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^b$ and sample $\{u_i\}_{i=1}^b$ from $\{q_{U|X=x_i}\}_{i=1}^b$
4: Compute $\nabla L_1 = \nabla \frac{1}{b} \sum_{i=1}^b [\log(q_{U|X}(u_i|x_i; \boldsymbol{\theta}_{\text{po}}) - \log q_{U|Y}(u_i|y_i; \boldsymbol{\theta}_{\text{pr}})] - s[d(g(x_i, y_i), f(u_i, y_i; \boldsymbol{\theta}_{\text{dec}}))]$
5: Update $\boldsymbol{\theta}_{\text{po}} \leftarrow \boldsymbol{\theta}_{\text{po}} - \nabla_{\boldsymbol{\theta}_{\text{po}}} L_1$ and $\boldsymbol{\theta}_{\text{dec}} \leftarrow \boldsymbol{\theta}_{\text{dec}} - \nabla_{\boldsymbol{\theta}_{\text{dec}}} L_1$
6: **for** $t' = 0$ to T' **do**
7: Sample minibatch $\mathcal{B}' = \{(x_i, y_i)\}_{i=1}^b$ and sample $\{u_i\}_{i=1}^b$ from $\{q_{U|X=x_i}\}_{i=1}^b$
8: Compute $\nabla L_2 = \nabla \frac{1}{b} \sum_{i=1}^b [\log(q_{U|X}(u_i|x_i; \boldsymbol{\theta}_{\text{po}}) - \log q_{U|Y}(u_i|y_i; \boldsymbol{\theta}_{\text{pr}})]$
9: Update $\boldsymbol{\theta}_{\text{pr}} \leftarrow \boldsymbol{\theta}_{\text{pr}} - \nabla_{\boldsymbol{\theta}_{\text{pr}}} L_2$

probability distribution influenced by $q_{U|X}$ and f, while p has been used to denote distributions independent of $q_{U|X}$ and f.

To proceed with this approach, we parameterize the key components of the optimization problem using a neural network based model. First, we represent the conditional distribution $q_{U|X}(u|x)$ as $q_{U|X}(u|x; \boldsymbol{\theta}_{\text{po}})$ where $\boldsymbol{\theta}_{\text{po}}$ denotes a set of parameters for $q_{U|X}$. Similarly, we parameterize the decoding function with a set of parameters $\boldsymbol{\theta}_{\text{dec}}$ as $f(u, y; \boldsymbol{\theta}_{\text{dec}})$.

It should be noted that the parameterization of $q_{U|X}(u|x; \boldsymbol{\theta}_{\text{po}})$ directly determines the related marginal and joint distributions, such as $q_{U|X,Y}$, $q_{U|Y}$, and $q_{X,Y,U}$ under the fixed $p_{X,Y}$. These distributions, governed by the parameter set $\boldsymbol{\theta}_{\text{po}}$, are thus denoted as $q_{U|X,Y;\boldsymbol{\theta}_{\text{po}}}$, $q_{U|Y;\boldsymbol{\theta}_{\text{po}}}$, and $q_{X,Y,U;\boldsymbol{\theta}_{\text{po}}}$.

In summary, we begin with the Lagrangian optimization problem for the rate-distortion function, also known as the supporting hyperplane method. We employ neural networks to parameterize the key components of our loss function. This optimization strategy draws parallels with the conventional Blahut-Arimoto algorithms [6], [7] in terms of formulating the Lagrangian loss, while also drawing inspiration from recent works [4], which has achieved state-of-the-art results in rate-distortion estimation through neural networks.

In the following subsections, we delve into a comprehensive explanation of our proposed algorithm, detailing the steps and techniques involved. We will also discuss the methods used for parameterizing the components in our framework.

A. Algorithm

The proposed method is detailed in Algorithm 1. This algorithm iteratively computes the gradient of the loss function (3) over T training iterations and updates the relevant parameters to minimize the the loss.

Line 3. Specifically, in each iteration, a minibatch with size b, $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^b$, is sampled. To estimate the expectation $\mathbb{E}_{X,Y,U}[\log q_{U|X}(U|X) - \log q_{U|Y}(U|Y)]$, it is necessary to generate data point triples (x_i, y_i, u_i) following the distribution $p_{X,Y}(x, y)q_{U|X}(u|x; \boldsymbol{\theta}_{\text{po}})$. For each sampled pair (x_i, y_i), a corresponding u_i is drawn from the distribution $q_{U|X}(u|x; \boldsymbol{\theta}_{\text{po}})$. This sampling results in triples (x_i, y_i, u_i) that adhere to the joint distribution $q_{X,Y,U}(x, y, u) = p_{X,Y}(x, y)q_{U|X}(u|x; \boldsymbol{\theta}_{\text{po}})$.

Utilizing these samples, we compute the average gradient of the loss function, which involves the computation of the expected value of $\log \frac{q_{U|X}(U|X)}{q_{U|Y}(U|Y)}$. This corresponds to $\log \frac{q_{U|X}(U|X;\boldsymbol{\theta}_{\text{po}})}{q_{U|Y;\boldsymbol{\theta}_{\text{po}}}(U|Y)}$ based on the parameterization where $q_{U|Y;\boldsymbol{\theta}_{\text{po}}}$ is formulated as

$$\begin{aligned} q_{U|Y;\boldsymbol{\theta}_{\text{po}}}(u|y) &= \sum_{x \in \mathcal{X}} p_{X|Y}(x|y) q_{U|X,Y;\boldsymbol{\theta}_{\text{po}}}(u|x, y) \\ &= \sum_{x \in \mathcal{X}} p_{X|Y}(x|y) q_{U|X}(u|x; \boldsymbol{\theta}_{\text{po}}). \end{aligned} \tag{4}$$

The efficient computation of $q_{U|Y;\boldsymbol{\theta}_{\text{po}}}$ is critical, as it needs to be executed for multiple instances to obtain the average of the log probability. However, this computation of (4) presents a substantial challenge due to the unknown nature of the distribution $p_{X|Y}$, with only sample-based access available. Furthermore, using sampling approaches for the estimation of the sum over $\mathcal{X}$ is non-trivial when domain $\mathcal{X}$ is a high-dimensional space and the data instances are limited. To address this issue, we leverage the following lemma, with its proof detailed in Appendix B.

Lemma 1. *Consider a fixed set of parameters $\boldsymbol{\theta}_{po}$ and scenario where the side information Y is available only at the decoder. Then we have*

$$\arg\min_{\hat{q}_{U|Y}} \mathbb{E}_{X,Y,U} \left[\log \frac{q_{U|X}(U|X; \boldsymbol{\theta}_{po})}{\hat{q}_{U|Y}(U|Y)} \right] = q_{U|Y;\boldsymbol{\theta}_{po}}. \tag{5}$$

Based on this lemma, we conclude that instead of executing the summation in (4) to derive $q_{U|Y;\boldsymbol{\theta}_{\text{po}}}$ for a given $q_{U|X}(u|x; \boldsymbol{\theta}_{\text{po}})$, we can model the distribution of U given Y as $q_{U|Y}(u|y; \boldsymbol{\theta}_{\text{pr}})$ where $\boldsymbol{\theta}_{\text{pr}}$ denotes a set of free parameters and then use the parameterized distribution $q_{U|Y}(u|y; \boldsymbol{\theta}_{\text{pr}})$ as an argument for the problem (5). The solution of (5) will lead to $q_{U|Y}(u|y; \boldsymbol{\theta}_{\text{pr}}) = q_{U|Y;\boldsymbol{\theta}_{\text{po}}}(u|y)$ as long as the parametrization of $q_{U|Y}(u|y; \boldsymbol{\theta}_{\text{pr}})$ is expressive enough.

Lines 4-5. By using the parametrized functions $q_{U|X}(u|x; \boldsymbol{\theta}_{\text{po}})$, $q_{U|Y}(u|y; \boldsymbol{\theta}_{\text{pr}})$, and $f(u, y; \boldsymbol{\theta}_{\text{dec}})$, in Line 4, we compute the gradient of (3). Subsequently, in Line 5, the parameters $\boldsymbol{\theta}_{\text{po}}$ and $\boldsymbol{\theta}_{\text{dec}}$ are updated to minimize the loss.

Lines 6-9. At the end of each iteration, we update $q_{U|Y}(u|y; \boldsymbol{\theta}_{\text{pr}})$ by solving (5) based on the newly updated $q_{U|X}(u|x; \boldsymbol{\theta}_{\text{po}})$ to correctly compute the main loss function (3) in the subsequent iteration. Problem (5) can be solved through gradient descent updates of the set of parameters $\boldsymbol{\theta}_{\text{pr}}$ as described in Lines 7-9 of Algorithm 1. More specifically,

for each inner-iteration (occurring T' times), we sample a minibatch and obtain pairs $\{(x_i, y_i, u_i)\}_{i=1}^{b}$. We then update $\boldsymbol{\theta}_{\text{pr}}$ to minimize the objective in (5). Practically, we have found that setting $T' = 1$ and reusing the same minibatch $\mathcal{B}$ for $\mathcal{B}'$ not only offers computational efficiency but also provides a tight upper bound on the rate-distortion function relative to theoretical optimality (as detailed in Sec. III).

B. Parameterization

In Algorithm 1, we utilize three distinct parameterized models: $q_{U|X}(u|x;\boldsymbol{\theta}_{\text{po}})$, $q_{U|Y}(u|y;\boldsymbol{\theta}_{\text{pr}})$, and $f(u, y;\boldsymbol{\theta}_{\text{dec}})$. A conventional approach to parameterizing distributions involves assuming a specific distribution form and then parameterizing its moments, such as the mean and variance. Various parameterization setups exist, including Gaussian, uniform distribution-based parameterizations, and more sophisticated forms relevant to modern machine learning research [17]. In our study, we opt for Gaussian distributions for parameterization. For example, in sampling from the distribution $q_{U|X}(u|x;\boldsymbol{\theta}_{\text{po}})$, the random variable U is assumed to follow a Gaussian distribution characterized by mean $\mu(x;\boldsymbol{\theta}_{\text{po}})$ and variance $\Sigma(x;\boldsymbol{\theta}_{\text{po}})$, both of which depend on the given realization x. The functions μ and Σ can be designed in various ways, depending on the specifics of the problem, where they take x as input and output the corresponding mean and variance.

We provide more details on the implementation in Sec. III. The choice of parameterization and the construction of these functions yield a point that represents an upper bound on the rate-distortion curve. This is because the variable spaces for the minimization problem in (3) is constrained by the assumptions inherent in the chosen distribution models. Thus, while these parameterizations facilitate the computational tractability of the problem, they also inherently define the limits of the solution space explored in the optimization process.

III. Numerical Evaluation

In order to evaluate our algorithm's efficacy, our initial step involves scenarios where the true rate-distortion function for computing with side information is known in closed form. Beyond these environments, we consider estimating the rate-distortion function for practical problem associated with channel state information (CSI) compression [18], incorporating side information.

A. 2-Component White Gaussian Noise

We adapt a scenario from [11, Sec. 21.1], featuring a 2-component White Gaussian Noise (2-WGN(P, ρ)) source, where (X, Y) forms pairs of i.i.d. jointly Gaussian random variables. Each pair in the sequence $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ has zero mean ($\mathbb{E}[X] = \mathbb{E}[Y] = 0$), equal variance ($\mathbb{E}[X^2] = \mathbb{E}[Y^2] = P$), and a correlation coefficient $\rho = \mathbb{E}[XY]/P$. With a squared error distortion measure d and a function $g(X, Y) = (X + Y)/2$, $R_{\text{D,C}}$ is given by

$$R_{\text{D,C}}(D) = \max\left\{\frac{1}{2}\log\left(\frac{P(1-\rho^2)}{4D}\right), 0\right\}. \tag{6}$$

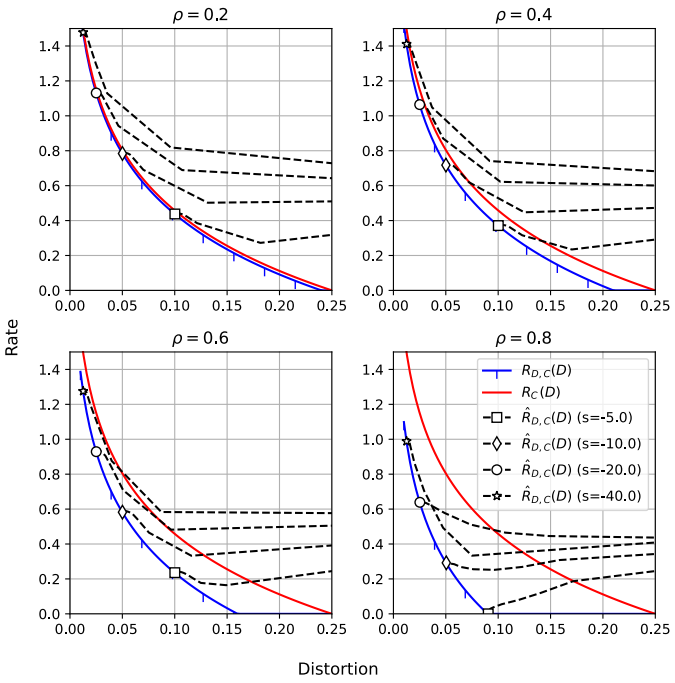

Fig. 3. Compression of Gaussian sources: Rate-distortion functions for computing with side information for various ρ and the estimated points.

To implement our approach, we employed a multi-layer perceptron (MLP) to model $q_{U|X}(u|x;\boldsymbol{\theta}_{\text{po}})$, $q_{U|Y}(u|y;\boldsymbol{\theta}_{\text{pr}})$, and $f(u, y;\boldsymbol{\theta}_{\text{dec}})$. Specifically, for $q_{U|X}(u|x;\boldsymbol{\theta}_{\text{po}})$ and $q_{U|Y}(u|y;\boldsymbol{\theta}_{\text{pr}})$, we use a single-layer MLP that takes an n-dimensional input and outputs a $2n$-dimensional vector, half for mean and half for variance, to model an n-dimensional independent multivariate Gaussian distribution. For $f(u, y;\boldsymbol{\theta}_{\text{dec}})$, we used a 2-layer MLP with leaky ReLU activation, which takes (u, y) as an input and outputs an n-dimensional $\hat{z}$.

In Fig. 3, we set $P = 1, n = 100$, and provide simulation results for various ρ values in $\{0.2, 0.4, 0.6, 0.8\}$. Each subplot displays the $R_{\text{D,C}}$ curves, alongside four rate-distortion points estimated by our algorithm for different slopes s. We also plot R_{C} curves, which refers to the rate-distortion function for computing without side information, which can be obtained by setting $\rho = 0$. y-axis has natural units (Nats) and x-axis represents mean squared error distortion. The dashed lines associated with $\hat{R}_{\text{D,C}}(D)$ corresponds to the learning trajectory, i.e., the achieved (distortion, rate) points during the training process.

Our algorithm consistently estimates points on $R_{\text{D,C}}$ within a small tolerance of less than 1e-3. A decrease in the s value corresponds to points on the left side of the curve, indicating higher rates and lower distortion. As can be seen, a higher correlation ρ results in a larger gap between $R_{\text{C}}(D)$ and $R_{\text{D,C}}(D)$, and our method effectively estimates points on $R_{\text{D,C}}$ regardless of various ρ values.

B. Applications to CSI Compression

This subsection focuses on the compression of Frequency Division Duplex Downlink (DL) Channel State Information (CSI), an area of growing interest in wireless research [18].

1) Setup: The objective is to compress the DL CSI, X, at the User Equipment (UE) side. The UE then transmits this compressed information, or codeword U, to the Base Station (BS). The aim is to minimize the Normalized Mean Squared Error (NMSE), defined as $\mathbb{E}[\|X - \hat{X}\|_2^2/\|X\|_2^2]$, where $\hat{X}$ is the decoder output and $\|\cdot\|_2$ is elementwise square norm.

To enhance compression efficiency, uplink (UL) CSI can be utilized as side information Y. This is based on the observation that UL CSI is typically acquired (available) via pilot transmissions from the UE to BS, and is correlated with DL CSI due to frequency-invariant characteristics [19], [20].

2) Simulation environment configuration: A CSI instance X characterized by the setting (n_{tx},n_{sc}), with $n_{\text{tx}} = 8$ represents the number of transmit antennas and $n_{\text{sc}} = 667$ denotes the number of subcarriers. UL CSI also has the same parameters. Our numerical evaluation use the Quasi Deterministic Radio channel generator [21]. We model channel distributions with 3GPP-3D antenna configurations, setting DL and UL center frequencies at 1.91GHz and 2.11GHz, respectively. UEs are randomly placed within a 300m diameter area around a centrally located BS, in an urban microcell environment with non-line-of-sight conditions.

3) Training configuration: We employ the Adam optimizer with a learning rate varying from 5e-4 to 1e-6 and a minibatch size of 100.

4) Parameterization of distributions: CSI instances are preprocessed by converting to the angular-delay domain via Inverse Fast Fourier Transform (IFFT) and trimming high-delay near-zero regions, following existing CSI preprocessing methods [22], [23]. This results in an 8×32 complex valued matrix, with 8 angular and 32 cropped delay components. We utilize inception block-based [24] encoding and decoding schemes [25] for the distribution parameterization. The encoder outputs a $2\times8\times32$ matrix, divided into mean matrix of 8×32 and variance matrix of 8×32 for $q_{U|X}(u|x;\boldsymbol{\theta}_{\text{po}})$. The same structure is used for $q_{U|Y}(u|y;\boldsymbol{\theta}_{\text{pr}})$. The decoder, $f(u,y;\boldsymbol{\theta}_{\text{dec}})$, takes two 8×32 complex valued matrices (codeword and side information) as input, processes them through a linear layer followed by the inception block-based decoder, outputting an 8×32 complex matrix. This output undergoes zero-padding and FFT for spatial frequency domain recovery.

5) A constructive neural CSI compression algorithm: For further analysis, we implement CSI compression algorithms using fixed rates (codeword lengths) with the same architecture used in Sec. III-B4. Consider l_{cl} sized binary codeword for the compression. For efficient implementation, we take 16 as a new base and consider codewords of length $l = l_{\text{cl}}/\log_2(16)$ to maintain cardinality. The encoder outputs a vector $\mathbf{U}_e \in \mathbb{R}^{\frac{l_{\text{cl}}}{\log_2(16)}\times N_{\text{Ebd}}}$ where N_{Ebd} is an embedding dimension. This vector is quantized using a trainable codebook of 16 different N_{Ebd}-dimensional vectors, based on [26]. The encoder transmits indices of these vectors via a wireless link to the BS, forming codeword U. The BS reconstructs $\mathbf{U}_e$ using these indices and the corresponding vectors in the codebook. The same decoder modules are then applied. We vary binary codeword lengths $l_{\text{cl}} \in \{64, 128, 256, 512\}$, $N_{\text{Ebd}} = 8$, set the loss function as the NMSE, and employ the same optimization techniques to reduce distortion.

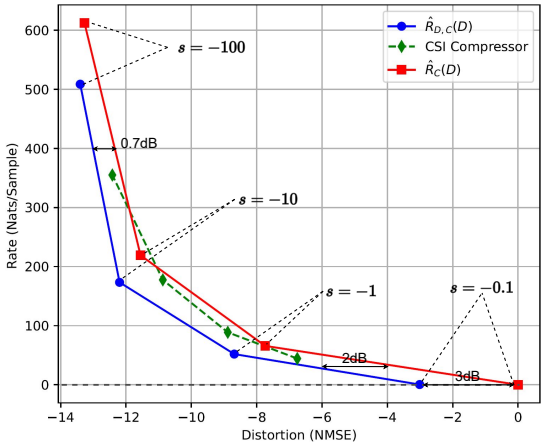

Fig. 4. CSI Compression: Comparison of the estimated rate-distortion function, estimated rate-distortion function with side information, and (distortion, rate) points achieved by the neural compression algorithm.

6) Results: In Figure 4, we illustrate the estimated rate-distortion function $\hat{R}_{\text{D,C}}$ along with $\hat{R}_{\text{C}}$, which is the estimated R_{C} by [4] and using the same neural architectures but which ignores the side information. By adjusting s values (-0.1, -1, -10, -100), we explore distortion levels from 0dB to approximately -14dB, connecting these points linearly to serve as an upper bound for the true rate-distortion curves.

As expected, introducing UL CSI for DL CSI compression is beneficial as $\hat{R}_{\text{D,C}} < \hat{R}_{\text{C}}$, especially at lower CSI feedback rates. For instance, with no DL CSI transmission (0 nats/sample), the BS can still retrieve reasonable information from UL CSI, achieving -3dB NMSE. At a feedback rate of 40 Nats/Sample, the gain from UL CSI side information is approximately 2dB. This advantage diminishes with increased feedback resources; for example, at 400 Nats/Sample, the gain is around 0.7dB.

The neural compression algorithm, incorporating side information, achieved a rate-distortion curve situated between $\hat{R}_{\text{D,C}}$ and $\hat{R}_{\text{C}}$. Given that $\hat{R}_{\text{D,C}}$ establishes an upper bound of $R_{\text{D,C}}$, the discrepancy between $R_{\text{D,C}}$ and the real CSI compression algorithm's performance signals room for improvement. For example, in the case of 177 Nats/sample, we may anticipate an improvement exceeding 1dB. Notably, this gap is less pronounced in scenarios with lower rates, allowing one to have a conjecture that the actual performance of the CSI compression algorithms is closer to $\hat{R}_{\text{D,C}}$.

IV. Discussion

In this paper, we propose a new algorithm for estimating the generalized rate-distortion function, with a specific emphasis on the rate-distortion function for computing with side information. This approach can offer estimated rates for given distortion levels and also enables the formulation of reliable conjectures about the benefits of side information at varying compression rates. Such a methodology is anticipated to be valuable in practical system design, allowing system designers to effectively measure the potential gains from side information against its processing costs through informed estimations.

REFERENCES

[1] T. Berger, "Rate-distortion theory," *Wiley Encyclopedia of Telecommunications*, 2003.

[2] Q. Li and Y. Chen, "Rate distortion via deep learning," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 456–465, 2019.

[3] E. Lei, H. Hassani, and S. S. Bidokhti, "Neural estimation of the rate-distortion function with applications to operational source coding," *IEEE Journal on Selected Areas in Information Theory*, 2023.

[4] Y. Yang and S. Mandt, "Towards empirical sandwich bounds on the rate-distortion function," *The International Conference on Learning Representations (ICLR)*, 2022.

[5] Y. Yang, S. Eckstein, M. Nutz, and S. Mandt, "Estimating the rate-distortion function by wasserstein gradient descent," *arXiv preprint arXiv:2310.18908*, 2023.

[6] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE transactions on Information Theory*, vol. 18, no. 4, pp. 460–473, 1972.

[7] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 14–20, 1972.

[8] Y. Uğur, I. E. Aguerri, and A. Zaidi, "A generalization of blahut-arimoto algorithm to compute rate-distortion regions of multiterminal source coding under logarithmic loss," in *2017 IEEE Information Theory Workshop (ITW)*. IEEE, 2017, pp. 349–353.

[9] A. Dembo and L. Kontoyiannis, "Source coding, large deviations, and approximate pattern matching," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1590–1615, 2002.

[10] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[11] A. El Gamal and Y.-H. Kim, *Network information theory*. Cambridge university press, 2011.

[12] S. Ayzik and S. Avidan, "Deep image compression using decoder side information," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer, 2020, pp. 699–714.

[13] N. Mital, E. Özyılkan, A. Garjani, and D. Gündüz, "Neural distributed image compression using common information," in *2022 Data Compression Conference (DCC)*. IEEE, 2022, pp. 182–191.

[14] Y. Huang, B. Chen, S. Qin, J. Li, Y. Wang, T. Dai, and S.-T. Xia, "Learned distributed image compression with multi-scale patch matching in feature domain," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4322–4329.

[15] F. Dupuis, W. Yu, and F. M. Willems, "Blahut-arimoto algorithms for computing channel capacity and rate-distortion with side information," in *International Symposium onInformation Theory, Proceedings.* IEEE, 2004, p. 179.

[16] H. Yamamoto, "Wyner-ziv theory for a general function of the correlated sources (corresp.)," *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 803–807, 1982.

[17] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[18] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Overview of deep learning-based csi feedback in massive mimo systems," *IEEE Transactions on Communications*, vol. 70, no. 12, pp. 8017–8045, 2022.

[19] D. Vasisht, S. Kumar, H. Rahul, and D. Katabi, "Eliminating channel feedback in next-generation cellular networks," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 398–411.

[20] D. Han, J. Park, and N. Lee, "Fdd massive mimo without csi feedback," *arXiv preprint arXiv:2302.04398*, 2023.

[21] "Quasi deterministic radio channel generator, user manual and documentation," *Tech. Rep.*, vol. v2.6.1, 2021.

[22] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[23] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based csi feedback approach for time-varying massive mimo channels," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 416–419, 2018.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[25] Z. Lu, J. Wang, and J. Song, "Multi-resolution csi feedback with deep learning in massive mimo system," in *IEEE International Conference on Communications*. IEEE, 2020, pp. 1–6.

[26] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

APPENDIX A
PROBLEM FORMULATION (3)

The definition of conditional mutual information leads to the following formulation

$$\min_{q_{U|X}(u|x),f} \Big\{ I(X;U|Y) - s\mathbb{E}[d(Z,\hat{Z})] \Big\} \tag{7}$$

$$= \min_{q_{U|X}(u|x),f} \Big\{ \sum_{\mathcal{X},\mathcal{Y},\mathcal{U}} q_{X,Y,U}(x,y,u) \log \frac{p_Y(y) q_{X,Y,U}(x,y,u)}{p_{X,Y}(x,y) q_{Y,U}(y,u)} - s\mathbb{E}[d(Z,\hat{Z})] \Big\} \tag{8}$$

This can be equivalently expressed as

$$\min_{q_{U|X}(u|x),f} \Big\{ \mathbb{E}\Big[\log \frac{q_{U|X,Y}(U|X,Y)}{q_{U|Y}(U|Y)} \Big] - s\mathbb{E}[d(Z,\hat{Z})] \Big\} \tag{9}$$

Here, the expectation is taken with respect to the joint distribution of (X,Y,U). For given realizations of X and Y, the conditional distribution of the codeword U is determined solely by X based on the communication model that we deal with. This restriction arises from the system model, which does not allow for the codeword to be controlled based on side information. Consequently, this simplifies to $q_{U|X,Y}(U|X,Y) = q(U|X)$, thereby completing the proof.

APPENDIX B
PROOF OF LEMMA 1

We start with the following equation:

$$q_{U|X}(U|X;\boldsymbol{\theta}_{\text{po}}) = q_{U|X,Y;\boldsymbol{\theta}_{\text{po}}}(U|X,Y). \tag{10}$$

This equation stems from the premise that the distribution of codeword U is deterministic on X when it is given. Following this, we have

$$\begin{aligned}
&\arg\min_{\hat{q}_{U|Y}} \mathbb{E}_{X,Y,U}\left[\log \frac{q_{U|X}(U|X;\boldsymbol{\theta}_{\text{po}})}{\hat{q}_{U|Y}(U|Y)} \right] \\
&= \arg\min_{\hat{q}_{U|Y}} \mathbb{E}_{X,Y}[\mathbf{KL}(q_{U|X}(U|X;\boldsymbol{\theta}_{\text{po}}) \| \hat{q}_{U|Y}(U|Y))] \\
&= \arg\min_{\hat{q}_{U|Y}} \mathbb{E}_{X,Y}[\mathbf{KL}(q_{U|X,Y;\boldsymbol{\theta}_{\text{po}}}(U|X,Y) \| \hat{q}_{U|Y}(U|Y))] \\
&= \arg\min_{\hat{q}_{U|Y}} \mathbb{E}_{X,Y}\Big[\sum_{\mathcal{U}} q_{U|X,Y;\boldsymbol{\theta}_{\text{po}}}(u|X,Y) \log \frac{q_{U|X,Y;\boldsymbol{\theta}_{\text{po}}}(u|X,Y)}{\hat{q}_{U|Y}(u|Y)} \Big] \\
&= \arg\min_{\hat{q}_{U|Y}} \mathbb{E}_{Y}\Big[\sum_{\mathcal{X}} \sum_{\mathcal{U}} p_{X|Y}(x|Y) q_{U|X,Y;\boldsymbol{\theta}_{\text{po}}}(u|x,Y) \log \frac{q_{U|X,Y;\boldsymbol{\theta}_{\text{po}}}(u|x,Y)}{\hat{q}_{U|Y}(u|Y)} \Big] \\
&= \arg\min_{\hat{q}_{U|Y}} \mathbb{E}_{Y}\Big[\sum_{\mathcal{X}} \sum_{\mathcal{U}} q_{U,X|Y;\boldsymbol{\theta}_{\text{po}}}(u,x|Y) \log \frac{q_{U|X,Y;\boldsymbol{\theta}_{\text{po}}}(u|x,Y)}{\hat{q}_{U|Y}(u|Y)} \Big] \\
&= \arg\min_{\hat{q}_{U|Y}} \mathbb{E}_{Y}\Big[\sum_{\mathcal{X}} \sum_{\mathcal{U}} q_{U,X|Y;\boldsymbol{\theta}_{\text{po}}}(u,x|Y) \log \frac{q_{U|X}(u|x;\boldsymbol{\theta}_{\text{po}})}{\hat{q}_{U|Y}(u|Y)} \Big] \\
&= \arg\max_{\hat{q}_{U|Y}} \mathbb{E}_{Y}\Big[\sum_{\mathcal{X}} \sum_{\mathcal{U}} q_{U,X|Y;\boldsymbol{\theta}_{\text{po}}}(u,x|Y) \log \hat{q}_{U|Y}(u|Y) \Big] \\
&= \arg\max_{\hat{q}_{U|Y}} \mathbb{E}_{Y}\Big[\sum_{\mathcal{U}} q_{U|Y;\boldsymbol{\theta}_{\text{po}}}(u|Y) \log \hat{q}_{U|Y}(u|Y) \Big].
\end{aligned} \tag{11}$$

As both $q_{U|Y;\boldsymbol{\theta}_{\text{po}}}$ and $\hat{q}_{U|Y}$ are probability distributions, applying Gibbs' inequality completes the proof.