Neural Estimation of the Rate-Distortion Function for Massive Datasets

Eric Lei, Hamed Hassani, Shirin Saeedi Bidokhti
Department of Electrical and Systems Engineering, University of Pennsylvania
{elei, hassani, saeedi}@seas.upenn.edu

Abstract
A fundamental question in designing lossy data compression schemes is how well one can do in comparison with the rate-distortion function, which describes the known theoretical limits of lossy compression. Motivated by the empirical success of deep neural network (DNN) compressors on large, real-world data, we investigate methods to estimate the rate-distortion function on such data, which would allow comparison of DNN compressors with optimality. While one could use the empirical distribution of the data and apply the Blahut-Arimoto algorithm, this approach presents several computational challenges when the datasets are large and high-dimensional, such as the case of modern image datasets. Instead, we re-formulate the rate-distortion objective, and solve the resulting functional optimization problem using neural networks. We provide experimental results on popular image datasets, and provide theoretical evidence why our method can accurately estimate the rate-distortion function. Additionally, we show that the rate-distortion achievable by DNN compressors are within several bits of the rate-distortion function. Lastly, we connect the rate-distortion objective and entropic optimal transport, and describe a method to implement an operational lossy compression scheme with guarantees on the achievable rate and distortion.

Index Terms
Lossy compression, rate-distortion theory, neural networks, generative models, optimal transport

I. INTRODUCTION
Driven by advances in deep neural network (DNN) compression schemes, rapid progress has been made in finding high-performing lossy compression schemes for large, high-dimensional datasets that remain practical [1]–[4]. While these methods have empirically shown to outperform classical compression schemes for real-world data (e.g. images), it remains unknown as to how well they perform in comparison to the fundamental limit, which is given by the rate-distortion function. To investigate this question, one approach is to examine a stylized data source with a known probability distribution that is analytically tractable, such as the sawbridge random process, as done in [5]. This allows for a closed-form solution of the rate-distortion function; one can then compare it with empirically achievable rate and distortion of DNN compressors trained on realizations of the source. However, this approach does not evaluate DNN compressors on true sources of interest, such as real-world images, for which architectural choices such as convolutional layers have been engineered [6]. Thus, evaluating the rate-distortion function on these sources is paramount to understanding the efficacy of DNN compressors on real-world data.

Consider an independent and identically-distributed (i.i.d.) data source $X \sim P_X$, where $P_X$ is a probability distribution supported on alphabet $\mathcal{X}$. Let $\mathcal{Y}$ be the reproduction alphabet, and $d: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ be a distortion function on the input and output alphabets. The asymptotic limit on the minimum number of bits required to achieve a distortion $D$ is given by the rate-distortion function [7]–[9], defined as

$$R(D) := \inf_{P_{Y|X}: \mathbb{E}_{P_{X,Y}}[d(X,Y)] \leq D} I(X; Y)$$

Any rate-distortion pair $(R, D)$ satisfying $R > R(D)$ is achievable by some lossy source code, and no code can achieve a rate-distortion less than $R(D)$. It is important to note that $R(D)$ is achievable only under asymptotic blocklengths, whereas DNN compressors are typically one-shot, as compressing i.i.d. blocks for real-world datasets may not be feasible. However, the one-shot achievable region is known to be within $\log(R(D) + 1) + O(1)$ bits of $R(D)$ [10], and thus even in the one-shot setting, $R(D)$ remains an appropriate measure of the fundamental limits.

There are several immediate challenges when computing $R(D)$ for large-scale data. Even when the distribution of $P_X$ is known, the analytical form of the rate-distortion function has been difficult to evaluate, and has been characterized only on specific sources. This prohibits an analytical derivation using a density estimate of $P_X$ (which are also not sample-efficient in high dimensions) in most cases. Computational methods such as the Blahut-Arimoto (BA) algorithm seem to be better suited for our setting; however, as will be shown, BA provides inaccurate estimates and fails to scale with large datasets.

The work of Eric Lei was supported by a NSF Graduate Research Fellowship.
optimizers take the following closed form: 

\[
R(D) = \inf_{\beta} D_{\beta}(P_X, Q_Y) = D_{\beta}(P_X | | P_X \otimes Q_Y) + \beta \mathbb{E} [d(X, Y)]
\]

where \(D_{\beta}(P_X, Q_Y) = \sum_{y \in Y} p_Y(y) d_{\beta}(P_X, Q_Y(y))\) is the rate-distortion function at rate \(R = D_{\beta}(P_X, Q_Y)\), with \(D_{\beta}(P_X, Q_Y(y))\) a convex and strictly decreasing function of \(\beta\) [9]. The minimizers \(P^*_X, Q^*_Y\) of (2) yields a unique point \(R^*_\beta = D_{\beta}(P^*_X \| | P^*_X \otimes Q^*_Y)\), and the zero-distortion rate (with squared-error distortion) achieved by DNN compressors.

\[R^*_\beta = \inf_{\beta} D_{\beta}(P_X, Q_Y) = D_{\beta}(P_X, Q_Y) + \beta \mathbb{E} [d(X, Y)]\]

for empirical distributions. Hence, getting an accurate estimate for the rate-distortion function at rate \(R\) on the true MNIST distribution, via this method, requires at least \(n = 60,000\) samples, which is clearly infeasible. Finally, the distortion corresponding to \(R = 0\), known as \(\alpha\), is given by \(\mathbb{E}_{P_X} ||X - \mu_X||^2\).

A. Blahut-Arimoto Fails to Scale

Let \(D_{\text{KL}}(\mu || \nu)\) be the Kullback-Leibler (KL) divergence, defined as \(\mathbb{E}_\mu[\log \frac{d\nu}{d\mu}]\) when the density \(\frac{d\nu}{d\mu}\) exists and \(+\infty\) otherwise. Due to the convex and strictly decreasing properties [9] of \(R(D)\), it suffices to fix some \(\beta > 0\), and solve the following double minimization problem:

\[\min_{\beta} \left\{ \min_{P_X, Q_Y} D_{\beta}(P_X, Q_Y) + \beta \mathbb{E} [d(X, Y)] \right\}
\]

Lemma 1 (Double-Minimization Form, cf. [9, Ch. 10], [11]). The optimizers \(P_{X,Y}^{(\beta)}, Q_Y^{(\beta)}\) of

\[R_{\beta}(D) := \inf_{\beta} \inf_{P_X, Q_Y} D_{\beta}(P_X, Q_Y) + \beta \mathbb{E} [d(X, Y)]\]

yields a unique point \(R^*_\beta = D_{\beta}(P^*_X \| | P^*_X \otimes Q^*_Y)\) and \(D_{\beta} = \mathbb{E}_{P_X} [d(X, Y)]\) on the positive-rate regime of the rate-distortion curve, i.e. \(R(D_{\beta}) = R^*_\beta\), such that \(D_{\beta} < D_{\text{max}}\) where \(R(D_{\text{max}}) = 0\).

The Blahut-Arimoto (BA) solves (2) by alternating steps on \(P_{Y|X}\) and \(Q_Y\) until convergence. In discrete settings, the optimizers take the following closed form:

\[p(y|x) = \frac{r(y|x)e^{-\beta d(y, \hat{y})}}{\sum_{\hat{y} \in Y} r(\hat{y})e^{-\beta d(x, \hat{y})}}, \quad \forall x \in X, y \in Y\]

\[r(y) = \sum_{x \in X} p_X(x)p(y|x), \quad \forall y \in Y\]

Even though BA requires knowledge of the source distribution, one can use the empirical distribution \(\hat{P}_n = \frac{1}{n} \sum_{i=1}^n \delta_{X_i}\) as a proxy. This, however, does not scale in the case of modern datasets. Consider the setting when \(X = \mathbb{R}^m\) is continuous. Applying BA requires discretization of the input and output alphabets. In many cases, this would require acute knowledge of how to discretize \(\mathbb{R}^m\) to form an appropriate reconstruction alphabet \(Y\), and even if one could, it might result in computational complexity that grows, potentially exponentially, with \(m\). One would need to store a \(n \times |Y|\) matrix for the conditional PMFs and a \(|Y|\)-sized vector for the output marginal PMF, which may not fit in memory depending on the number of data points or the choice of discretization. For example, in image compression, where we assume each \(X_i \in \mathbb{R}^m\) to be a single image realization, \(Y \subseteq \mathbb{R}^m\). Even for 8-bit grayscale images, full precision quantization would require \(2^8 \cdot m\) points, and although one could provide better discretization schemes, they may still require an intractable number of points.

To demonstrate, we attempt to apply discretized BA to MNIST digits in Fig. 1, and plot its estimated curve in comparison to rate-distortion (with squared-error distortion) achieved by DNN compressors. Specifically, our source is the empirical MNIST distribution \(P_n\), and we discretize \(Y\) to be exactly the support of our data, i.e. \(Y = \{X_i\}_{i=1}^n\). While this scheme should converge to the true rate-distortion function as \(n \to \infty\) [12], we see that even for \(n = 60,000\), this fails to capture the general trend of the DNN compressors. Additionally, the zero-distortion rate \(R(0)\), which is given by the entropy of the source when the source is discrete, is equal to \(\log_2(n)\) for empirical distributions. Hence, getting an accurate estimate for the rate-distortion function at rate \(R\) on the true MNIST distribution, via this method, requires at least \(2^R\) samples, which is clearly infeasible. Finally, the distortion corresponding to \(R = 0\), known as \(D_{\text{max}}\), that BA estimates is far from the optimal given by \(\mathbb{E}_{P_X} ||X - \mu_X||^2\) – see Section IV for more details. This showcases the inaccuracy of BA in estimating the rate-distortion function even with relatively large number of samples. In contrast, our method, which provides the estimate \(\hat{R}_{\alpha}(D)\), does not exhibit these failures and is able to generalize to the true MNIST distribution.

Fig. 1: Inaccuracy of discretized Blahut-Arimoto in comparison to our method, \(\hat{R}_{\alpha}(D)\), for computing the rate distortion curve on the MNIST dataset. DNN compressors provide codes that lie in the achievable region. See text for details.
B. Related Work

To the best of the our knowledge, there are no known works in the literature that directly attempt to efficiently and accurately estimate $R(D)$ for large-scale data. Aside from the classical works of Arimoto [13] and Blahut [14], the most directly related work is [15], in which a mapping from the Lebesgue measure on $[0, 1]$ to $\mathcal{Y}$ is used to represent the reproduction distribution, and $[0, 1]$ is discretized rather than $\mathcal{Y}$ itself. In [12], the authors analyze theoretical properties of the plug-in estimator for $R(D)$. Both references do not provide a method that can demonstrably scale to modern datasets.

A related area of work lies in the generative modeling literature, where the rate-distortion trade-off is often used to evaluate generative models and unsupervised learning algorithms. The most relevant work is [16], where the authors take a rate-distortion perspective to evaluate the performance of generative adversarial networks (GANs) and variational autoencoders (VAEs). In their formulation, they assume the trained generative model is the output $Y$-marginal of the rate-distortion objective, and find an upper bound on the rate-distortion needed to reproduce the generative model, not the true rate-distortion function of the source.

Much of the other work in this area [17]–[19] use variational bounds on the rate-distortion for the purposes of representation learning, and lack a direct connection to fundamental limits of lossy compression.

C. Contributions

As opposed to the aforementioned approaches in Sec I, we take a step back and reformulate the rate-distortion objective into a min-max objective using duality, building on results from [20]. As will be shown, this alleviates many of the issues plaguing previous methods. Our contributions are as follows.

- We propose an estimator which we show is strongly consistent, and provide a corresponding algorithm to compute $R(D)$ from samples for a broad class of distortion measures.
- We empirically show that these methods provide accurate estimates of $R(D)$ on MNIST and FashionMNIST datasets, and that DNN autoencoder compressors achieve a rate-distortion within a few bits of our estimate.
- We establish a connection between the rate-distortion objective and entropic optimal transport, which provides an interpretable and simple upper bound on $R(D)$.
- Leveraging one-shot lossy source coding results, we describe how to implement a one-shot operational compressor with guarantees on the achievable rate and distortion.

In essence, we provide a scalable method to estimate the rate-distortion function on real-world datasets, and provide evidence that it is accurate using known properties of $R(D)$; we also demonstrate the achievable rate-distortion of state-of-the-art (one-shot) DNN compressors to be within several bits of our estimate. These findings open up further avenues for research. In particular, it remains open whether the gap between the performance of DNN compressors and the asymptotic $R(D)$ function is due to their one-shot nature and if we could close this gap by developing DNN compressors that perform block coding. It also remains open as to whether or not our proposed one-shot codes described in Sec. V will work in practice and could outperform DNN compressors.

II. Problem Formulation

Our goal is to estimate the rate-distortion function $R(D)$ of some source $P_X$. However, we only have access to $n$ samples $X_1, \ldots, X_n$ drawn i.i.d. from $P_X$, and do not assume any other knowledge about its distribution.

As opposed to BA, which solves the double minimization problem (2) in closed form, we use the dual form of the rate-distortion function. We first note that the constrained version of the inner minimization problem in (2) is known as the rate function in the literature [12], [20], i.e.

$$R(Q_Y, D) := \inf_{P_{Y|X}} \text{KL}(P_{X,Y} \| P_X \otimes Q_Y),$$

which exhibits the following dual characterization.

**Lemma 2** (Rate Function Duality, [20, Sec. 2]). The rate function can be equivalently expressed as follows.

$$R(Q_Y, D) = \sup_{\beta \leq 0} \beta \bar{D} - \mathbb{E}_{P_X} \left[ \log \mathbb{E}_{Q_Y} \left[ e^{\beta \bar{d}(X,Y)} \right] \right]$$

Therefore, $R(D)$ is equivalent to $\inf_{Q_Y} R(Q_Y, D)$ and can be expressed as a min-max problem,

$$R(D) = \inf_{Q_Y} \sup_{\beta \leq 0} \beta \bar{D} - \mathbb{E}_{P_X} \left[ \log \mathbb{E}_{Q_Y} \left[ e^{\beta \bar{d}(X,Y)} \right] \right]$$
Algorithm 1 Neural Estimator of the Rate-Distortion Function

Input: Distortion constraint $D$, sample size $B$, number of steps $T$, learning rate $\eta$
Initial generator neural network $G_\theta : \mathcal{Z} \to \mathcal{Y}$

for $t = 1, 2, \ldots, T$ do
    Sample $\{x_i\}_{i=1}^B \overset{i.i.d.}{\sim} P_X$
    Sample $\{z_j\}_{j=1}^B \overset{i.i.d.}{\sim} P_Z$
    Define $\kappa_{i,j}(\tilde{\beta}, \theta_t) := \exp\left(\tilde{\beta} d(x_i, G_\theta(z_j))\right)$
    Solve $D = \frac{1}{B} \sum_{i=1}^B d(x_i, G_\theta(z_j)) \frac{B \kappa_{i,j}(\tilde{\beta}, \theta_t)}{\sum_{j=1}^B \kappa_{i,j}(\tilde{\beta}, \theta_t)}$ for $\tilde{\beta}^*$
    $\theta_{t+1} \leftarrow \theta_t - \eta \nabla_\theta \left(-\frac{1}{B} \sum_{i=1}^B \log \frac{1}{B} \sum_{j=1}^B \kappa_{i,j}(\tilde{\beta}^*, \theta_t)\right)$
end for

which is far more amenable to estimation from samples, since we can approximate expectations with empirical averages for both $P_X$, $Q_\mathcal{Y}$ independently. Furthermore, the inner problem is concave, scalar, and has a unique solution. To solve the inner max, first-order stationary conditions yield \[20, \text{Sec. 2.2}\]

$\nabla_\beta \left[ \frac{1}{B} \sum_{i=1}^B d(x_i, G_\theta(z_i)) \frac{B \kappa_{i,j}(\tilde{\beta}, \theta_t)}{\sum_{j=1}^B \kappa_{i,j}(\tilde{\beta}, \theta_t)} \right] = 0$

Remark 1. A naive approach is to parametrize $Q_\mathcal{Y}$ using neural networks in the double minimization form in (2). In fact, the mapping approach [15] uses a similar idea, where the space $\mathcal{Z}$ is discretized rather than $\mathcal{Y}$, and $G$ can be optimally fit to find the best discretization of $\mathcal{Y}$. One could then solve the inner minimization in the first BA step (3) using batched samples as uniform empirical distributions, similar to the method in Sinkhorn GANs [21]. However, doing so still suffers from needing a number of samples exponential in $R$, since the KL term is still computed on a discrete distribution.

III. NEURAL ESTIMATION OF THE RATE-DISTORTION FUNCTION

We propose to parametrize the output marginal distribution $Q_\mathcal{Y}$ using architectural choices similar to those used in the GAN literature [22]. Specifically, let $P_\mathcal{Z}$ be some simple base distribution over $\mathcal{Z}$ and let $G : \mathcal{Z} \to \mathbb{R}^m$ be a function belonging to a function class $\mathcal{G}$. Then, representing distributions $Q_\mathcal{Y}$ with the pushforward $G_* P_\mathcal{Z}$, we can optimize over functions in $\mathcal{G}$, and arrive at

$$R_\mathcal{G}(D) := \inf_{G \in \mathcal{G}} \sup_{\tilde{\beta} \leq 0} \tilde{\beta} D - \mathbb{E}_{P_X} \left[ \log \mathbb{E}_{P_Z} e^{\tilde{\beta} d(X, G(Z))} \right]$$ \hspace{1cm} (9)

The equivalence of this (under certain assumptions on $P_\mathcal{Z}$) with $R(D)$ is justified in [15]. In practice, we only have access to samples $X_1, \ldots, X_n$ drawn i.i.d. from $P_X$, and must estimate (9) from the empirical distribution $\hat{P}_X^{(n)} := \frac{1}{n} \sum_{i=1}^n \delta_{X_i}$. Leveraging the expressive power of neural networks, we choose $\mathcal{G}$ to be the class of functions parametrized by neural networks, and arrive at the following estimator (NERD).

Definition 1 (Neural Estimator of the Rate-Distortion Function (NERD)). Let $\mathcal{G} := \{G_\theta\}_{\theta \in \Theta}$ be a class of functions parametrized by a neural network. NERD is given by

$$\hat{R}_{\mathcal{G}}(D)_n := \inf_{\theta \in \Theta} \sup_{\tilde{\beta} \leq 0} \tilde{\beta} D - \mathbb{E}_{\hat{P}_X^{(n)}} \left[ \log \mathbb{E}_{P_Z} e^{\tilde{\beta} d(X, G_\theta(Z))} \right]$$ \hspace{1cm} (10)

$$= \inf_{\theta \in \Theta} \sup_{\tilde{\beta} \leq 0} \tilde{\beta} D - \frac{1}{n} \sum_{i=1}^n \log \mathbb{E}_{P_Z} e^{\tilde{\beta} d(X_i, G_\theta(Z_i))}$$ \hspace{1cm} (11)

The next theorem shows that NERD is a strongly consistent estimator for the rate distortion function. The proof is provided in Appendix A.

Theorem 1 (Strong consistency of NERD). Suppose the alphabets are $X = \mathcal{Y} = \mathbb{R}^m$, $P_Z$ is supported on $\mathcal{Z} \subseteq \mathbb{R}^l$, where $\mathcal{Z}$ is compact, and $\mathcal{G} \subseteq C^1(\mathcal{Z})$. Also, suppose that $d$ is continuous, bounded by some constant $M > 0$, and satisfies $d(x, y) \leq c||x - y||_\infty$ for some $c > 0$. Then the NERD estimator in (11) is strongly consistent, i.e.

$$\lim_{n \to \infty} \hat{R}_{\mathcal{G}}(D)_n = R_{\mathcal{G}}(D) \text{ almost surely.}$$ \hspace{1cm} (12)
To use NERD, following (11), one can simply sample batches from \( P_X \) and \( P_Z \), solve the inner max of (7) by solving (8) for \( \beta^* \), and take a gradient step over the DNN parameters. The full algorithm is given in Algorithm 1.

### IV. Experimental Results

In our experiments we use MNIST digits and Fashion MNIST (FMNIST) images to represent our source \( X \sim P_X \). In both cases, we have \( n = 60,000 \) i.i.d. samples from \( P_X \). Furthermore, we have that \( X := [0,1]^m \), where \( m = 784 \), and use the squared-error distortion function on vectors, i.e. \( d(x, y) = \|x - y\|^2_2 \). We set the base distribution as \( P_Z = N(0, I_{100}) \) and parametrize \( G_\theta : \mathbb{R}^{100} \to [0,1]^m \) with a deep convolutional architecture similar to the generator architecture used in DCGAN [23].

#### A. Comparison with DNN Compressors

We use Alg. 1 to solve (11) using a variant of stochastic gradient descent via the backpropagation algorithm [24]. We plot the estimated rate-distortion curves in Fig. 2 for FMNIST and Fig. 1 for MNIST. In both cases, the curve indeed satisfies the convex and strictly decreasing properties of the rate-distortion function. Using our estimate of \( R(D) \), we can use DNN compressors of the autoencoder type [1]–[3] to see how they perform compared to the fundamental limits. In both cases, we see that DNN compressors closely follow the same trend as the estimated rate-distortion function, and are within several bits of optimality inside the achievable region. However, it remains difficult to conclude in this case whether or not DNN compressor are optimal on these datasets. The gap of several bits could be potentially attributed to the fact that the DNN compressors are one-shot, whereas \( R(D) \) is achievable only under asymptotic blocklengths. While one-shot achievable regions of \( R > R(D) + \log(R(D) + 1) + 5 \) are known [10], lower bounds tighter than \( R(D) \) are not as clear. Either way, it remains to be seen whether other computationally feasible source codes could be designed to perform closer to the rate-distortion limit.

#### B. Effect of Arbitrary Output Marginal

We compare solving (11) to a baseline scheme that uses Blahut-Arimoto on discretized input and output alphabets. Specifically, we let \( X = \{X_1, \ldots, X_n \in [0,1]^m \} \) and let the source PMF be \( \frac{1}{n} \sum_{i=1}^n \delta_{X_i}(x) \) and choose a discretization for \( Y \subseteq [0,1]^m \) to define an output marginal PMF for Blahut-Arimoto. We choose the discretization for \( Y \) to be the same as the source, i.e. \( Y = \{X_1, \ldots, X_n \in [0,1]^m \} \) is exactly the support of the samples. Such a scheme should converge to the true rate-distortion function as \( n \to \infty \) assuming the true continuous alphabets are both \([0,1]^m \). However, we demonstrate that an intractable number of samples will be needed to accurately estimate \( R(D) \). Firstly, we are limited by the number of samples (60,000 at most with both datasets). Even with a large number of samples, we see that, given in Fig. 1 and 2, doing so does not work particularly well, and the trend is completely off compared to NERD and the DNN codes. It fails to extrapolate to the true rate-distortion function of the true source, and traces the rate-distortion curve for the discrete uniform empirical distribution which we see achieves zero distortion at \( R = H(P_X^{(n)}) = \log_2 n \). As \( n \) grows larger, we would expect the curve traced by this scheme to “rotate” clockwise to the true rate-distortion curve (which requires infinite rate at zero-distortion for continuous sources), but this scheme can only rotate to where the zero-distortion rate reaches \( \log_2(60,000) \approx 15.87 \). In contrast, NERD is able to follow the same trend of the operational rate-distortion curve estimated by DNN compressors, and matches known characteristics of \( R(D) \) as described in the next section.
C. Samples from the Optimal Reproduction Distribution

We now illustrate generated samples from the optimal reproduction distribution, parametrized by the trained neural network, and show that it indeed aligns with the behavior of the rate-distortion function. Let $D_{\text{max}} := \min_{y \in \mathcal{Y}} \mathbb{E}_{P_X}[d(X, y)]$ be the distortion achievable at zero rate [11, Ch. 9], i.e. $R(D_{\text{max}}) = 0$. This is the best distortion that can be possibly achieved when there is no information about the source, where the reproduction is simply the best constant estimate of $X$. When $d$ is squared-error, and $\mathcal{X} = \mathcal{Y}$, the best constant estimate is the mean $\mu_X = \mathbb{E}_{P_X}[X]$, with $D_{\text{max}} = \mathbb{E}_{P_X}[\|X - \mu_X\|^2]$. In the MNIST case, the samples generated from the generator neural network at $R = 0.28$ bits, shown in Fig. 3a, consistently generate the “mean image”. Computing $D_{\text{max}}$ with an empirical average turns out to be $\approx 56$ (under mild preprocessing of the MNIST dataset), which matches the zero-rate point in Fig. 1. In contrast, samples generated from a trained generator at higher rate, shown in Fig. 3b, appear more similar to the original MNIST images and produce more modes of the distribution. A similar phenomenon occurs with FMNIST as well, shown in Appendix ??.

V. UPPER BOUNDS OF $R(D)$ AND OPERATIONAL SCHEMES

In this section, we first describe an upper bound on the rate-distortion function via entropic optimal transport (OT), and then describe how it can help construct an operational lossy compression scheme.

A. Upper Bound via Entropic Optimal Transport

The upper bound is most easily seen when considering the combined rate-distortion objective $RD(\beta)$ in (2).

**Proposition 1** (Entropic OT Upper Bound). Given source $P_X$ on $\mathcal{X}$, reconstruction space $\mathcal{Y}$, and distortion measure $d$, $RD(\beta) \leq \inf_{Q_Y \in \mathcal{P}(\mathcal{Y})} \beta \cdot W_1(P_X, Q_Y)$ (13)

The RHS objective is the entropic OT problem $W_\epsilon(\mu, \nu) := \inf_{\pi \in \Pi(\mu, \nu)} \mathbb{E}_\pi[d(X, Y)] + \epsilon \mathbb{D}_{\text{KL}}(\pi \| \mu \otimes \nu)$, (14)

where $\Pi(\mu, \nu)$ is the set of couplings between $\mu$ and $\nu$.

**Remark 2.** The inner minimization problem in $RD(\beta)$ in (2) only has a marginal constraint on $P_X$, whereas the inner minimization in Prop. 1 has an additional marginal constraint on $Q_Y$ as well.
Under discrete settings, (14) can be solved efficiently via Sinkhorn iterations [25]. We call the upper bound in (13) RD-Sinkhorn. In order to illustrate the behavior of RD-Sinkhorn, we first examine the rate-distortion curve of a discretized Gaussian source over 6 atoms, shown in Fig. 4. In such a setting, we can compute the true rate-distortion curve and RD-Sinkhorn easily and exactly, via Blahut-Arimoto and Sinkhorn iterations respectively. Although RD-Sinkhorn traces a curve that is not convex, we see that for large portions of the rate-distortion curve, the gap is small and RD-Sinkhorn empirically provides a good estimate of the true rate-distortion function. We then apply RD-Sinkhorn to MNIST by parametrizing $Q_Y$ with a generator neural network, similar to the work in [21], and compare it with NERD in Fig. 5.

**B. Operational Scheme via the PFR**

We now describe how the entropic OT upper bound can be used to implement an operational scheme via the Poisson functional representation (PFR) [10]. For any channel $Q_{Y|X}$, we can construct a one-shot code using $Q_{Y|X}$ via the PFR that will achieve a rate of $I(P_X, Q_{Y|X})$ and distortion $\mathbb{E}_{P_X Q_{Y|X}}[d(X,Y)]$. As described in [10, Sec. 4], if $Q_Y$ is the output marginal of $P_X Q_{Y|X}$, this can be done by generating the same marked Poisson process $\{(Y_i, T_i)\}_{i=1}^{\infty}$, where $Y_i \overset{i.i.d.}{\sim} P_Y$, $T_i \overset{i.i.d.}{\sim} \text{Exp}(1)$, at both the encoder and decoder (assuming shared common randomness). To compress $X$, the encoder encodes $k = \arg \min_i T_i \cdot \mathbb{E}_d Q_{Y|X}(T_i)$ with a prefix code which the decoder can losslessly recover, and output $Y_k$. To use the PFR, we need three things: (i) the conditional density $q(y|x)$ of $Q_{Y|X}$, (ii) the density $p_Y(y)$ of $Q_Y$, and (iii) the ability to generate samples from $Q_Y$. We already have (iii), and can get (ii) using generative models with density estimates. We outline our approach for (i) below.

**Remark 3.** One way to implement the PFR and achieve a rate-distortion given by optimizers of RD-Sinkhorn is through the entropic OT dual [26]:

$$
W_\pi^*(Q_Y, P_X) = \sup_{f,g \in \mathcal{C}(X)} \mathbb{E}_{P_X}[f(X)] + \mathbb{E}_{P_Y}[g(Y)] - \beta^{-1} \left( \mathbb{E}_{P_X \otimes P_Y} \left[ e^{\beta(f(X)+g(Y)-d(X,Y))} \right] - 1 \right)
$$

Solving this yields optimal dual potentials $f^*, g^*$ that satisfy $\pi_{X,Y}^* = e^{\beta(f^* \otimes g^*-d)}(P_X \otimes Q_Y)$ [27]. Hence, if one knows the $Q_Y$ density $q(y)$, the conditional density for $Q_{Y|X}$ can be recovered from the coupling via $q^*(y|x) = e^\frac{1}{\beta}(f^*(x)+g^*(y)-d(x,y)) \cdot q(y)$. Finding $f^*$ and $g^*$ can be done by neural estimators for statistical divergences [28]–[30] or through first-order stationary conditions [27].

**VI. Conclusion**

In this paper, we propose a new algorithm for computing the rate-distortion function for real-world data. We use an alternative formulation of the rate-distortion objective which is amenable to parametrization with neural networks and provide an algorithm that is sample and computationally efficient. We empirically show that it accurately estimates the rate-distortion function for large datasets. Finally, we provide upper bounds and describe a method for operational one-shot codes.

**References**

Theorem 2 (Theorem 1 in text). Suppose the alphabets are $X = Y = \mathbb{R}^m$, $P_Z$ is supported on $Z \in \mathbb{R}^l$, where $Z$ is compact, and $G \subseteq C^l(Z)$. Also, suppose that $d$ is continuous, bounded by some constant $M > 0$, and satisfies $d(x, y) \leq c \|x - y\|_\infty$ for some $c > 0$. Then the NERD estimator in (11) is strongly consistent, i.e.,

$$
\lim_{n \to \infty} R_{\theta_n}(D)_n = R_G(D) \text{ almost surely.}
$$

Proof. Fix $\epsilon > 0$. Define

$$
R(G, D) := \sup_{\beta \leq 0} \beta D - \mathbb{E}_{P_X} \left[ \log \mathbb{E}_{P_Z} e^{\beta d(X, G(Z))} \right].
$$
and let

$$f(G, \beta) := \mathbb{E}_{P_X \otimes P_Y} \left[ d(X, G(Z)) \frac{\exp(\beta d(X, G(Z)))}{\mathbb{E}_{Z' \sim P_Z} [\exp(\beta d(X, G(Z)))]} \right] - D$$

From [20], $f(\beta, G)$ is continuously differentiable and hence by the implicit function theorem, there is a continuously differentiable function $\beta(G)$ such that $f(\beta(G), G) = 0$. Since $\beta(G)$ provides the optimizer of $R(G, D)$ [20], we have that

$$R(G, D) = \beta(G) D - \mathbb{E}_{P_X} \left[ \log \mathbb{E}_{P_Z} [\exp(\beta(G) d(X, G(Z)))] \right].$$

By definition, $R^*_G(D) = \inf_G R(G, D)$, so $\exists \tilde{G} \in G$ s.t. $R(\tilde{G}, D) < R_G(D) + \epsilon/4$. We would like to find a $G_\theta$ such that $|R(\tilde{G}, D) - R(G_\theta, D)| < \epsilon/4$. Indeed, we have that

$$|R(\tilde{G}, D) - R(G_\theta, D)| \leq D|\beta(\tilde{G}) - \beta(G_\theta)| + \mathbb{E}_{P_X} \left[ \log \mathbb{E}_{P_Z} \left[ \frac{\exp(\beta(\tilde{G}) d(X, \tilde{G}(Z)))}{\exp(\beta(G_\theta) d(X, G_\theta(Z)))} \right] \right] \leq \epsilon/4.$$  

The first term can be bounded via continuity of $\beta(G)$ and universal approximation results in [31, Thm. 3.2]. Specifically, there is some $\delta_1 > 0$ such that $\|\tilde{G} - G\|_\infty \leq \delta_1$ implies $|\beta(\tilde{G}) - \beta(G)| < \epsilon/4$. Since we assume $\tilde{G}$ is defined on a compact domain, we can find a neural network $G_\theta$ with $l$ input neurons, $m$ output neurons, and an arbitrary number of $l + m + 2$-neuron hidden layers [31, Thm. 3.2] such that $\|G_\theta - \tilde{G}\|_\infty \leq \delta_1$. As for the second term, we may bound it as follows:

$$\mathbb{E}_{P_X} \left[ \log \mathbb{E}_{P_Z} \left[ \frac{\exp(\beta(\tilde{G}) d(X, \tilde{G}(Z)))}{\exp(\beta(G_\theta) d(X, G_\theta(Z)))} \right] \right] \leq \mathbb{E}_{P_X} \left[ \log \mathbb{E}_{P_Z} \left[ \frac{\exp(\beta(\tilde{G}) d(X, \tilde{G}(Z)))}{\exp(\beta(G_\theta) d(X, G_\theta(Z)))} \right] - 1 \right] \leq \mathbb{E}_{P_X} \left[ \exp \left( \sup_{z \in Z} \beta(\tilde{G}) d(X, \tilde{G}(z)) - \beta(G_\theta) d(X, G_\theta(z)) \right) \right] $$

where we use the fact that $\log x \leq x - 1$ as well as the inequality $\mathbb{E}[\phi(W)] = \mathbb{E} [\phi(W) \frac{\phi(W)}{\mathbb{E}[\phi(W)]}] \leq \mathbb{E}[\phi(W)] \sup_{w \in W} \frac{\phi(w)}{\mathbb{E}[\phi(W)]}$ for any random variable $W$ supported on $W$ and any functions $\phi, \phi'$. We may further bound the exponent via

$$\sup_z \left( \beta(\tilde{G}) d(X, \tilde{G}(z)) - \beta(G_\theta) d(X, G_\theta(z)) \right) = \sup_z \left| d(X, G_\theta(z)) \beta(\tilde{G}) - \beta(G_\theta) \right| + \sup_z \left( \beta(\tilde{G}) d(X, \tilde{G}(z)) - d(X, G_\theta(z)) \right) \leq M \sup_z \left| \beta(\tilde{G}) - \beta(G_\theta) \right| + \sup_z \left( \beta(\tilde{G}) d(X, \tilde{G}(z)) - d(X, G_\theta(z)) \right)$$

where we use the fact that $d$ is bounded by $M$. To bound this by $\log(\epsilon/8)$, the first term can again be bounded via continuity, i.e. $\exists \delta_2 > 0$ such that $|\beta(\tilde{G}) - \beta(G_\theta)| \leq \frac{\log(\epsilon/8)}{2M \delta_2}$. The second term can be controlled to error $\frac{1}{2} \log(\epsilon/8)$ by choosing a neural network $G_\theta$ such that $\|G_\theta - \tilde{G}\|_\infty \leq \frac{\log(\epsilon/8)}{2M \delta_2}$. Again via [31], we have that the second term in (17) is bounded by $\mathbb{E}_{P_X} [\exp(\log(\epsilon/8))] = \epsilon/8$. Thus if we choose the neural network $G_\theta$ (with aforementioned size) such that $\|G_\theta - \tilde{G}\|_\infty < \min \left( \delta_1, \delta_2, \frac{\log(\epsilon/8)}{2M \delta_2} \right)$, we have that $|R(\tilde{G}, D) - R(G_\theta, D)| < \epsilon/4$. This can be done via [31, Thm. 3.2]. Since $R(G_\theta, D)$ is an upper bound of $R_G(D)$, we have that

$$|R_G(D) - R_{G_\theta}(D)| := |R_G(D) - \inf_{G_\theta} R(G_\theta, D)| < \epsilon/2$$

by the triangle inequality. Applying the strong consistency result for the parametric problem in [12], $\exists N \in \mathbb{N}$ s.t. $\forall n \geq N$, $|R_{G_\theta}(D)_n - R_{G_\theta}(D)| \leq \epsilon/2$ almost surely, and so again by triangle inequality, we have that $|R_{G_\theta}(D)_n - R_G(D)| < \epsilon$ for all $n \geq N$, almost surely.

**Proposition 2** (Proposition 1 in text). Given source $P_X$ on $\mathcal{X}$, reconstruction space $\mathcal{Y}$, and distortion measure $d$, $\mathbf{R}(\beta) \leq \inf_{Q_Y \in P(\mathcal{Y})} \beta \cdot W_\beta \left( P_X, Q_Y \right)$

The RHS objective is the entropic OT problem

$$W_\beta(\mu, \nu) := \inf_{\pi \in \Pi(\mu \otimes \nu)} \mathbb{E}_\pi [d(X, Y)] + \epsilon \text{KL}(\pi || \mu \otimes \nu),$$

where $\Pi(\mu, \nu)$ is the set of couplings between $\mu$ and $\nu$. 

\(\square\)
Proof. Scaling by $\text{RD}(\beta)$ by $1/\beta$,
\[
\beta^{-1} \text{RD}(\beta) = \inf_{P_{Y|X}} \mathbb{E}_{P_{X,Y}}[d(X,Y)] + \beta^{-1} I(X;Y)
\]
\[
= \inf_{Q_Y} \inf_{P_{Y|X}} \mathbb{E}_{P_{X,Y}}[d(X,Y)] + \beta^{-1} \text{D}_{\text{KL}}(P_{X,Y}||P_X \otimes Q_Y)
\]
\[
= \inf_{Q_Y} \inf_{\pi_{X,Y}: \pi_X = P_X} \mathbb{E}_{P_{X,Y}}[d(X,Y)] + \beta^{-1} \text{D}_{\text{KL}}(P_{X,Y}||P_X \otimes Q_Y)
\]
By enforcing an additional output marginal constraint in the inner minimization, we get the upper bound
\[
\beta^{-1} \text{RD}(\beta) \leq \inf_{Q_Y} \inf_{\pi_{X,Y}: \pi_X = P_X} \{ \mathbb{E}_{\pi_{X,Y}}[d(X,Y)] + \beta^{-1} \text{D}_{\text{KL}}(\pi_{X,Y}||P_X \otimes Q_Y) \}
\]
\[
= \inf_{Q_Y} W_1(Q_Y, P_X)
\]
where the inner problem’s feasible set contains couplings $\pi_{X,Y} \in \Pi(P_X, Q_Y)$ that marginalize to $P_X$ and $Q_Y$. \hfill \Box