TOWARDS EMPIRICAL SANDWICH BOUNDS ON THE RATE-DISTORTION FUNCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Rate-distortion (R-D) function, a key quantity in information theory, characterizes the fundamental limit of how much a data source can be compressed subject to a fidelity criterion, by *any* compression algorithm. As researchers push for ever-improving compression performance, establishing the R-D function of a given data source is not only of scientific interest, but also sheds light on the possible room for improving compression algorithms. Previous work on this problem relied on distributional assumptions on the data source (Gibson, 2017) or only applied to discrete data. By contrast, this paper makes the first attempt at an algorithm for sandwiching the R-D function of a general (not necessarily discrete) source requiring only i.i.d. data samples. We estimate R-D sandwich bounds on Gaussian and high-dimension banana-shaped sources, as well as GAN-generated images. Our R-D upper bound on natural images indicates room for improving the performance of state-of-the-art image compression methods by 1 dB in PSNR at various bitrates.

1 INTRODUCTION

From storing astronomical images captured by the Hubble telescope, to delivering familiar faces and voices over video chats, data compression, i.e., communicating the "same" information but with less bits, is commonplace and indispensable to our digital life, and even arguably lies at the heart of intelligence (Mahoney, 2009). While for lossless compression, there exist practical algorithms that can compress any discrete data arbitrarily close to the information theory limit (Ziv & Lempel, 1977; Witten et al., 1987), no such universal algorithm has been found for *lossy* data compression (Berger & Gibson, 1998), and significant research efforts have dedicated to lossy compression algorithms for various data. Recently, deep learning has shown promise for learning lossy compressors from raw data examples, with continually improving compression performance often matching or exceeding traditionally engineered methods (Minnen et al., 2018; Agustsson et al., 2020; Yang et al., 2020a).

However, there are fundamental limits to the performance of any lossy compression algorithm, due to the inevitable trade-off between *rate*, the average number of bits needed to represent the data, and the *distortion* incurred by lossy representations. This trade-off is formally described by the rate-distortion (R-D) function, for a given *source* (i.e., the data distribution of interest; referred to as such in information theory) and distortion metric. The R-D function characterizes the best theoretically achievable rate-distortion performance by any compression algorithm, which can be seen as a lossy-compression counterpart and generalization of Shannon entropy for lossless compression.

Despite its fundamental importance, the R-D function is generally unknown analytically, and establishing it for general data sources, especially real world data, is a difficult problem (Gibson, 2017). The default method for computing R-D functions, the Blahut-Arimoto algorithm (Blahut, 1972; Arimoto, 1972), only works for discrete data with a known probability mass function and has a complexity exponential in the data dimensionality. Applying it to an unknown data source requires discretization (if it is continuous) and estimating the source probabilities by a histogram, both of which introduce errors and are computationally infeasible beyond a couple of dimensions. Previous work characterizing the R-D function of images and videos (Hayes et al., 1970; Gibson, 2017) all assumed a statistical model of the source, making the results dependent on the modeling assumptions.

In this work, we make progress on this old problem in information theory using tools from machine learning, and introduce new algorithms for upper and lower bounding the R-D function of a *general* (i.e., discrete, continuous, or neither), *unknown* memoryless source. More specifically,

1. Our upper bound draws from the deep generative modeling toolbox, and is closely related to a class of β-VAEs in learned data compression (Ballé et al., 2017); we clarify how these models optimize a model-independent upper bound on the source rate-distortion function.
2. We theoretically derive a general lower bound on the R-D function that can in principle be optimized by gradient ascent. Facing the difficulty of the problem involving global optimization, we restrict to a squared error distortion and obtain an approximate algorithm.
3. We experimentally show that our upper bound can recover the R-D function of Gaussian sources, and obtain non-trivial sandwich bounds on high-dimension data with low intrinsic dimension. In particular, our sandwich bounds on 128×128 GAN-generated images shed light on the effectiveness of learned compression approaches (Ballé et al., 2021).
4. Although we currently cannot verify its tightness, our estimated R-D upper bound on natural images indicates possible room for improving the image compression performance of state-of-the-art methods by one dB in PSNR, at various bitrates.

We begin by reviewing the prerequisite rate-distortion theory in Section 2, then describe our upper and lower bound algorithms in Section 3 and Section 4, respectively. We discuss related work in Section 5, report experimental results in Section 6, and conclude in Section 7.

2 Background

Rate-distortion (R-D) theory deals with the fundamental trade-off between the average number of bits per sample (*rate*) used to represent a data source X and the *distortion* incurred by the lossy representation $\hat{X}$. It asks the following question about the limit of lossy compression: for a given data source and a distortion metric (a.k.a., a fidelity criterion), what is the minimum number of bits (per sample) needed to represent the source at a tolerable level of distortion, regardless of the computation complexity of the compression procedure? The answer is given by the rate-distortion function $R(D)$. To introduce it, let the source and its reproduction take values in the sets $\mathcal{X}$ and $\hat{\mathcal{X}}$, conventionally called the *source* and *reproduction alphabets*, respectively. We define the data source formally by a random variable $X \in \mathcal{X}$ following a (usually unknown) distribution P_X, and assume a distortion metric $\rho : \mathcal{X} \times \hat{\mathcal{X}} \to [0, \infty)$ has been given, such as the squared error $\rho(x, \hat{x}) = \|x - \hat{x}\|^2$. The rate-distortion function is then defined by the following constrained optimization problem,

$$R(D) = \inf_{Q_{\hat{X}|X}:\, \mathbb{E}[\rho(X,\hat{X})]\leq D} I(X; \hat{X}), \tag{1}$$

where we consider all random transforms $Q_{\hat{X}|X}$ whose expected distortion is within the given threshold $D \geq 0$, and minimize the mutual information between the source X and its reproduced representation $\hat{X}$ [1]. Shannon's lossy source coding theorems (Shannon, 1948; 1959) gave operational significance to the above mathematical definition of $R(D)$, as the minimum achievable rate with which any lossy compression algorithm can code i.i.d. data samples at a distortion level within D.

The R-D function thus gives the tightest lower bound on the rate-distortion performance of any lossy compression algorithm, and can inform the design and analysis of such algorithms. If the operational distortion-rate performance of an algorithm lies high above the source $R(D)$-curve $(D, R(D))$, then further performance improvement may be expected; otherwise, its performance is already close to theoretically optimal, and we may focus our attention on other aspects of the algorithm. As the R-D function does not have an analytical form in general, we propose to estimate it from data samples, making the standard assumption that various expectations w.r.t. the true data distribution P_X exist and can be approximated by sample averages. For a discrete source, this assumption automatically holds, and the R-D function also provides a lower bound on the Shannon entropy of discrete data.

3 Upper Bound Algorithm

For a known discrete source, the Blahut-Arimoto (BA) algorithm converges to $R(D)$ from above by fixed-point equations. For a general unknown source, it is not clear how this can be done. We

[1] Both the expected distortion and mutual information terms are defined w.r.t. the joint distribution $P_X Q_{\hat{X}|X}$. We formally describe the general setting of the paper, including the technical definitions, in Appendix A.1.

propose to solve the underlying variaitonal problem approximately by gradient descent. In exchange for generality and scalability, we lose the optimality guarantee of the BA algorithm, only arriving at a stochastic upper bound of $R(D)$ in general. By R-D theory (Cover & Thomas, 2006), every (distortion, rate) pair lying above the $R(D)$-curve is in principle realizable by a (possibly expensive) compression algorithm; an upper bound on $R(D)$, therefore, reveals what kind of (improved) R-D performance is theoretically possible, without suggesting how it can be practically achieved.

Variational Formulation. We adopt the same unconstrained variational objective as the Blahut-Arimoto (BA) algorithm (Blahut, 1972; Arimoto, 1972), in its most general form,

$$\mathcal{L}(Q_{\hat{X}|X}, Q_{\hat{X}}, \lambda) := \mathbb{E}_{x\sim P_X}[KL(Q_{\hat{X}|X=x} \| Q_{\hat{X}})] + \lambda \mathbb{E}_{P_X Q_{\hat{X}|X}}[\rho(X, \hat{X})], \tag{2}$$

where $Q_{\hat{X}}$ is an arbitrary probability measure on $\hat{\mathcal{X}}$ and $KL(\cdot\|\cdot)$ denotes the Kullback-Leibler (KL) divergence. This objective can be seen as a Lagrangian relaxation of the constrained problem defining $R(D)$, where the first (*rate*) term is a variational upper bound on the mutual information $I(X; \hat{X})$, and the second (*distortion*) term enforces the distortion tolerance constraint in Eq. 1. For each fixed $\lambda > 0$, a global minimizer of $\mathcal{L}$ yields a point $(\mathcal{R}, \mathcal{D})$ on the $R(D)$ curve (Csiszar, 1974), where $\mathcal{R}$ and $\mathcal{D}$ are simply the two terms of $\mathcal{L}$ evaluated at the optimal $(Q_{\hat{X}|X}, Q_{\hat{X}})$. Repeating this minimization for various λ then traces out the $R(D)$ curve. Based on this connection, the BA algorithm carries out the minimization by coordinate descent on $\mathcal{L}$ via fixed-point equations, each time setting $Q_{\hat{X}|X}$ to be optimal w.r.t. $Q_{\hat{X}}$ and vice versa; the sequence of alternating distributions can be shown to converge and yield a point on the $R(D)$ curve (Csiszar, 1974). Unfortunately, the BA algorithm only applies when $\mathcal{X}$ and $\hat{\mathcal{X}}$ are finite, and the source distribution P_X known (or estimated from data samples) in the form of a vector of probabilities $P_X(x)$ of every state $x \in \mathcal{X}$. Moreover, the algorithm requires storage and running time exponential in the data dimensionality, since it operates on exhaustive tabular representations of $P_X, \rho, Q_{\hat{X}|X}$, and $Q_{\hat{X}}$. The algorithm therefore quickly becomes infeasible on data with more than a couple of dimensions, not to mention high-dimension data such as natural images. The fixed-point equations of the BA algorithm are known in general settings (Rezaei et al., 2006), but when $\mathcal{X}$ or $\hat{\mathcal{X}}$ is infinite (such as in the continuous case), it is not clear how to carry them out or exhaustively represent the measures $Q_{\hat{X}}$ and $Q_{\hat{X}|X=x}$ for each $x \in \mathcal{X}$.

Proposed Method. To avoid these difficulties, we propose to apply (stochastic) gradient descent on $\mathcal{L}$ w.r.t. flexibly parameterized variational distributions $Q_{\hat{X}|X}$ and $Q_{\hat{X}}$. The distributions can be members from any variational family, as long as $Q_{\hat{X}|X=x}$ is absolutely continuous w.r.t. $Q_{\hat{X}}$ for (P_X-almost) all $x \in \mathcal{X}$. This technical condition ensures their KL divergence is well defined. This is easily satisfied, when $\hat{\mathcal{X}}$ is discrete, by requiring the support of $Q_{\hat{X}|X=x}$ to be contained in that of $Q_{\hat{X}}$ for all $x \in \mathcal{X}$; and in the continuous case, by representing both measures in terms of probability density functions (e.g., normalizing flows (Kobyzev et al., 2021)). In this work we represent $Q_{\hat{X}}$ and $Q_{\hat{X}|X}$ by parametric distributions, and predict the parameters of each $Q_{\hat{X}|X=x}$ by an *encoder* neural network $\phi(x)$ as in amortized inference (Kingma & Welling, 2014). Given i.i.d. X samples, we optimize the parameters of the variational distributions by SGD on $\mathcal{L}$; at convergence, the estimates of rate and distortion terms of $\mathcal{L}$ yields a point that in expectation lies on an R-D upper bound $R_U(D)$.

The variational objective $\mathcal{L}$ (Eq. 2) closely resembles the negative ELBO (NELBO) objective of a β-VAE (Higgins et al., 2017), if we regard the reproduction alphabet $\hat{\mathcal{X}}$ as the "latent space". The connection is immediate, when $\hat{\mathcal{X}}$ is continuous and a squared error ρ specifies the density of a Gaussian likelihood $p(x|\hat{x}) \propto \exp(-\|x - \hat{x}|^2)$. However, unlike in data compression, where $\hat{\mathcal{X}}$ is determined by the application (and often equal to $\mathcal{X}$ for a full-reference distortion), the latent space in a (β-)VAE typically has a lower dimension than $\mathcal{X}$; a *decoder* network is then used to parameterize a likelihood model in the data space. To capture this setup, we introduce a new, arbitrary latent space $\mathcal{Z}$ on which we define variational distributions $Q_{Z|X}, Q_Z$, and a (possibly stochastic) decoder function $\omega : \mathcal{Z} \to \hat{\mathcal{X}}$. This results in an extended objective (with $\mathcal{L}$ being the special case of an identity ω),

$$J(Q_{Z|X}, Q_Z, \omega, \lambda) := \mathbb{E}_{x\sim P_X}[KL(Q_{Z|X=x} \| Q_Z)] + \lambda \mathbb{E}_{P_X Q_{Z|X}}[\rho(X, \omega(Z))]. \tag{3}$$

How does this relate to the original rate-distortion problem? We note that the same results from rate-distortion theory apply, once we identify a new distortion function $\rho_\omega(x, z) := \rho(x, \omega(z))$ and treat $\mathcal{Z}$ as the reproduction alphabet. Then for each fixed decoder, we may define a ω-dependent

rate-distortion function, $R_\omega(D) := \inf_{Q_{Z|X}:\mathbb{E}[\rho_\omega(X,Z)]\leq D} I(X;Z)$. The minimum of J w.r.t. the variational distributions produces a point on the $R_\omega(D)$ curve. Moreover, as a consequence of the data processing inequality $I(X;Z) \geq I(X;\omega(Z))$, we can prove that $R_\omega(D) \geq R(D)$ for any ω (Theorem A.3). Moreover, the inequality is tight for a bijective ω, offering some theoretical support for the use of sub-pixel instead of upsampled convolutions in the decoder of image compression autoencoders (Theis et al., 2017; Cheng et al., 2020). We can now minimize the NELBO objective Eq. 3 w.r.t. parameters of $(Q_{Z|X}, Q_Z, \omega)$ similar to training a β-VAE, knowing that we are optimizing an upper bound on the information R-D function of the data source. This can be seen as a generalization to the lossless case (with a countable $\mathcal{X}$), where minimizing the NELBO minimizes an upper bound on the Shannon entropy of the source (Frey & Hinton, 1997), the limit of lossless compression.

The tightness of our bound depends on the choice of variational distributions. The freedom to define them over any suitable latent space $\mathcal{Z}$ can simplify the modeling task (of which there are many tools (Salakhutdinov, 2015; Kobyzev et al., 2021)). e.g., we can work with densities on a continuous $\mathcal{Z}$, even if $\hat{\mathcal{X}}$ is high-dimensional and discrete. We can also treat Z as the concatenation of sub-vectors $[Z_1, Z_2, ..., Z_L]$, and parameterize Q_Z in terms of simpler component distributions $Q_Z = \prod_{l=1}^{L} Q_{Z_l|Z_{<l}}$ (similarly for $Q_{Z|X}$). We exploit these properties in our experiments on images.

4 Lower Bound Algorithm

Without knowing the tightness of an R-D upper bound, we could be wasting time and resources trying to improve the R-D performance of a compression algorithm, when it is in fact already close to the theoretical limit. This would be avoided if we could find a matching *lower* bound on $R(D)$. Unfortunately, the problem turns out to be much more difficult computationally. Indeed, every compression algorithm, or every pair of variational distributions $(Q_{\hat{X}}, Q_{\hat{X}|X})$ yields a point above $R(D)$. Conversely, establishing a lower bound requires disproving the existence of *any* compression algorithm that can conceivably operate below the $R(D)$ curve. In this section, we derive an algorithm that can in theory produce arbitrarily tight R-D lower bounds. However, as an indication of its difficulty, the problem requires *globally* maximizing a family of partition functions. By restricting to a continuous reproduction alphabet and a squared error distortion, we make some progress on this problem and demonstrate useful lower bounds on data with low intrinsic dimension (see Sec. 6.2).

Dual characterization of $R(D)$. While upper bounds on $R(D)$ arise naturally out of its definition as a minimization problem, a variational lower bound requires expressing $R(D)$ through a *maximization* problem. For this, we introduce a "conjugate" function as the optimum of the Lagrangian Eq. 2 ($Q_{\hat{X}}$ is eliminated by replacing the rate upper bound with the exact mutual information $I(X;\hat{X})$):

$$F(\lambda) := \inf_{Q_{\hat{X}|X}} I(X;\hat{X}) + \lambda\mathbb{E}[\rho(X,\hat{X})]. \tag{4}$$

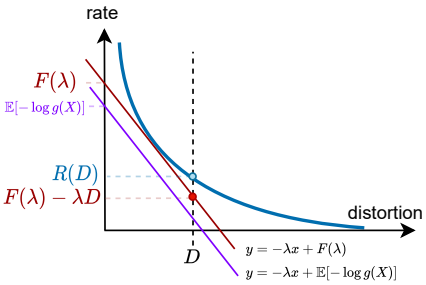

Figure 1: The geometry of the R-D lower bound problem. For a given slope $-\lambda$, we seek to maximize the R-axis intercept, $\mathbb{E}[-\log g(X)]$, over all $g \geq 0$ functions admissible according to Eq. 6.

As illustrated by Fig. 1, $F(\lambda)$ is the maximum R-axis intercept of a straight line with slope $-\lambda$, among all such lines that lie below or tangent to $R(D)$; the R-D curve can then be found by taking the upper envelope of lines with slope $-\lambda$ and R-intercept $F(\lambda)$, i.e., $R(D) = \max_{\lambda\geq 0} F(\lambda) - \lambda D$. This key result is captured mathematically by Lemma A.1, and the following:

Theorem 4.1. *(Csiszár, 1974) Under basic conditions (e.g., satisfied by a bounded ρ; see Appendix A.2), it holds that* (all expectations below are with respect to the data source r.v. $X \sim P_X$)

$$F(\lambda) = \max_{g(x)} \{\mathbb{E}[-\log g(X)]\} \tag{5}$$

where the maximization is over $g : \mathcal{X} \to [0, \infty)$ satisfying the constraint

$$\mathbb{E}\left[\frac{\exp(-\lambda\rho(X,\hat{x}))}{g(X)}\right] = \int \frac{\exp(-\lambda\rho(x,\hat{x}))}{g(x)} dP_X(x) \leq 1, \forall \hat{x} \in \hat{\mathcal{X}} \tag{6}$$

In other words, every admissible g yields a lower bound of $R(D)$, via an underestimator of the intercept $\mathbb{E}[-\log g(X)] \leq F(\lambda)$. We give the origin and history of this result in related work Sec. 5.

Proposed Unconstrained Formulation. The constraint in Eq. 6 is concerning – it is a family of possibly infinitely many constraints, one for each $\hat{x}$. To make the problem easier to work with, we propose to eliminate the constraints by the following transformation. Let g be defined in terms of another function $u(x) \geq 0$ and a scalar c depending on u, such that

$$g(x) := cu(x), \quad \text{where } c := \sup_{\hat{x} \in \hat{\mathcal{X}}} \Psi_u(\hat{x}), \quad \text{and } \Psi_u(\hat{x}) := \mathbb{E}\left[\frac{\exp -\lambda\rho(X, \hat{x})}{u(X)}\right]. \tag{7}$$

This reparameterization of g is without loss of generality, and can be shown to always satisfy the constraint in Eq. 6. While this form of g bears a superficial resemblance to an energy-based model (LeCun et al., 2006), with $\frac{1}{c}$ resembling a normalizing constant, there is an important difference: $c = \sup_{\hat{x}} \Psi_u(\hat{x})$ is in fact the supremum of a family of "partition functions" $\Psi_u(\hat{x})$ indexed by $\hat{x}$; we thus refer to c as the *sup-partition function*. Although all these quantities have λ-dependence, we omit this from our notation since λ is a fixed input parameter (as in the upper bound algorithm).

Consequently, F is now the result of *unconstrained* maximization over all u functions, and we obtain a lower bound on it by restricting u to a subset of functions with parameters θ (e.g., neural networks),

$$F(\lambda) = \max_{u \geq 0}\{\mathbb{E}[-\log u(X)] - \log \sup_{\hat{x} \in \hat{\mathcal{X}}} \Psi_u(\hat{x})\} \geq \max_{\theta}\{\mathbb{E}[-\log u_\theta(X)] - \log \sup_{\hat{x} \in \hat{\mathcal{X}}} \Psi_\theta(\hat{x})\}$$

For convenience, define the maximization objective by

$$\ell(\theta) := \mathbb{E}[-\log u_\theta(X)] - \log c(\theta), \quad c(\theta) := \sup_{\hat{x} \in \hat{\mathcal{X}}} \Psi_\theta(\hat{x}). \tag{8}$$

Then, it can in principle be maximized by stochastic gradient ascent using samples from P_X. However, computing the sup-partition function $c(\theta)$ poses serious computation challenges: even evaluating $\Psi_\theta(\hat{x})$ for a single $\hat{x}$ value involves a potentially high-dimensional integral w.r.t. P_X; this is only exacerbated by the need to globally optimize w.r.t. $\hat{x}$, an NP-hard problem even in one-dimension.

Proposed Method. To tackle this problem, we propose a sample-based over-estimator of the sup-partition function inspired by the IWAE estimator (Burda et al., 2015). Fix u_θ for now. Noting that $\Psi(\hat{x}) := \mathbb{E}[\psi(X, \hat{x})]$ is an expectation w.r.t. P_X, where $\psi(x, \hat{x}) := \frac{\exp -\lambda\rho(x,\hat{x})}{u(x)}$, one may then naturally consider a plug-in estimator for c, replacing the expectation by a sample estimate of $\Psi(\hat{x})$. Formally, given a sequence of i.i.d. random variables $X_1, X_2, ... \sim P_X$, we define the estimator $C_k := \sup_{\hat{x}} \frac{1}{k}\sum_i \psi(X_i, \hat{x})$ for each $k \geq 1$. We can then prove (see Theorem A.4 and proof in Appendix) that $\mathbb{E}[C_1] \geq \mathbb{E}[C_2] \geq ... \geq c$, i.e., C_k is on average an over-estimator of the sup-partition function c; and like the Importance-Weighted ELBO (Burda et al., 2015), the bias of the estimator decreases monotonically as $k \to \infty$, and that under continuity assumptions, C_k is asymptotically unbiased and converges to c. In light of this, we replace c by $\mathbb{E}[C_k]$ and obtain a k-sample under-estimator of the objective $\ell(\theta)$ (which in turn underestimates $F(\lambda)$):

$$\ell_k(\theta) := \mathbb{E}[-\log u_\theta(X)] - \log \mathbb{E}[C_k]; \quad \text{moreover, } \ell_1(\theta) \leq \ell_2(\theta) \leq ... \leq \ell(\theta).$$

Unfortunately, we still cannot apply stochastic gradient ascent to ℓ_k, as two more difficulties remain. First, C_k is still hard to compute, as it is defined through a global maximization problem. We note that by restricting to a suitable ρ and $\hat{X} = X$, the maximization objective of C_k has the form of a kernel density estimate (KDE). For a squared error distortion, this becomes a Gaussian mixture density, $\frac{1}{k}\sum_i \psi(x_i, \hat{x}) \propto \sum_i \pi_i \exp(-\lambda\|x_i - \hat{x}\|^2)$, with centroids defined by the samples $x_1, ..., x_k$, and mixture weights $\pi_i = (ku(x_i))^{-1}$. The global mode of a Gaussian mixture can generally be found by hill-climbing from each of the k centroids, except in rare artificial examples (Carreira-Perpinan, 2000; 2020); we therefore use this procedure to compute C_k, but note that other methods exist (Lee et al., 2019; Carreira-Perpinan, 2007). The second difficulty is that even if we could estimate $\mathbb{E}[C_k]$ (with samples of C_k computed by global optimization), the objective ℓ_k requires an estimate of its logarithm; a naive application of Jensen's inequality $-\log \mathbb{E}[C_k] \geq \mathbb{E}[-\log C_k]$ results in an *over*-estimator (as does the IWAE estimator), whereas we require a lower bound. Following Poole et al. (2019), we underestimate $-\log(x)$ by its linearization around a scalar parameter $\alpha > 0$, resulting in the following lower bound objective:

$$\tilde{\ell}_k(\theta) := \mathbb{E}[-\log u_\theta(X)] - \mathbb{E}[C_k]/\alpha - \log \alpha + 1. \tag{9}$$

$\ell_k(\theta)$ can finally be estimated by sample averages, and yields a lower bound on the optimal intercept $F(\lambda)$ by the chain of inequalities, $\tilde{\ell}_k(\theta) \leq \ell_k(\theta) \leq \ell(\theta) \leq F(\lambda)$. A trained model u_{θ^*} then yields an R-D lower bound, $R_L(D) = -\lambda D + \tilde{\ell}_k(\theta^*)$. We give the detailed algorithm in Appendix A.4.

5 Related Work

Machine Learning: The past few years have seen significant progress in applying machine learning to data compression. For lossless compression, explicit likelihood models (van den Oord et al., 2016; Hoogeboom et al., 2019) directly lead themselves to entropy coding, and bits-back coding techniques are actively being developed for efficient compression with latent variable models (Townsend et al., 2019; Kingma et al., 2019; Ho et al., 2019; Ruan et al., 2021). In the lossy domain, Theis et al. (2017); Ballé et al. (2017) showed that a particular type of VAE can be trained to perform data compression using the same objective as Eq. 3. The variational distributions in such a model have shape restrictions to simulate quantization and entropy coding (Ballé et al., 2017). Our upper bound is directly inspired by this line of work, and suggests that such a model can in principle compute the source R-D function when equipped with sufficiently expressive variational distributions and a "rich enough" decoder (see explanation in Sec. 3). We note however not all compressive autoencoders admit a probabilistic formulation (Theis et al., 2017); recent work has found training with hard quantization to improve compression performance (Minnen & Singh, 2020), and methods have been developed (Agustsson & Theis, 2020; Yang et al., 2020b) to reduce the gap between approximate quantization at training time and hard quantization at test time. Departing from compressive autoencoders, Yang et al. (2020c) and Flamich et al. (2020) applied regular Gaussian VAEs to data compression to exploit the flexiblility of variable-width variational posterior distributions. The REC algorithm from Flamich et al. (2020) can in theory transmit a sample of $Q_{\hat{X}|X}$ with a rate close to the rate term of the NELBO-like Eq. 3, but comes with non-negligible overhead. Our experiment in Sec. 6.3 points to the theoretically possible gain in image compression performance from this approach. Agustsson & Theis (2020) proved the general difficulty of this approach without assumptions on the form of $Q_{\hat{X}|X}$, and showed that the particular case of a uniform $Q_{\hat{X}|X}$ leads to efficient implementation based on dithered quantization.

Information theory has also broadly influenced unsupervised learning (Alemi et al., 2018; Poole et al., 2019) and representation learning (Tishby et al., 2000). The Information Bottleneck method (Tishby et al., 2000) was directly motivated as a more general R-D theory and borrows from the BA algorithm. Alemi et al. (2018) analyzed the relation between generative modeling and representation learning with a similar R-D Lagrangian to Eq. 2, but used an abstract, model-dependent distortion $\rho(\hat{x}, x) := -\log p(x|\hat{x})$ with an arbitrary $\hat{\mathcal{X}}$ and without considering a data compression task. Recently, Huang et al. (2020) proposed to evaluate decoder-based generative models by computing a restricted version of $R_\omega(D)$ (with $Q_{\hat{X}}$ fixed to a Gaussian); our result in Sec. 3 ($R_\omega(D) \geq R(D)$) gives a principled way to interpret and compare these model-dependent R-D curves.

Information Theory: While the BA algorithm (Blahut, 1972; Arimoto, 1972) solves the problem of computing the $R(D)$ of a known discrete source, no tools currently exist for the general and unknown case. Riegler et al. (2018) share our goal of computing $R(D)$ of a general source, but still require the source to be known analytically and supported on a known reference measure. Harrison & Kontoyiannis (2008) consider the same setup as ours of estimating $R(D)$ of an unknown source from samples, but focus on purely theoretical aspects, assuming prefect optimization. They prove statistical consistency of such estimators for a general class of alphabets and distortion metrics, assuring that our stochastic bounds on $R(D)$ optimized from data samples, when given unlimited computation and samples, can converge to the true $R(D)$. Perhaps closest in spirit to our work is Gibson (2017), who estimate lower bounds on $R(D)$ of speech and video using Gaussian autoregressive models of the source. However, the correctness of their bounds depends on the modeling assumptions.

A variational lower bound on $R(D)$ was already proposed by Shannon (1959), and later extended (Berger, 1971) to the present version similar to Theorems 4.1 and A.2. In the finite-alphabet case, the maximization characterization of $R(D)$ directly follows from taking the Lagrange dual of its standard definition in Eq. 1 (which is itself a convex optimization problem); the dual problem can then be solved by convex optimization (Chiang & Boyd, 2004), but faces the same limitations as the BA algorithm. A rigorous proof of Thereom 4.1 for general alphabets is more involved, and is based on repeated applications of information divergence inequalities (Csiszár, 1974; Gray, 2011).

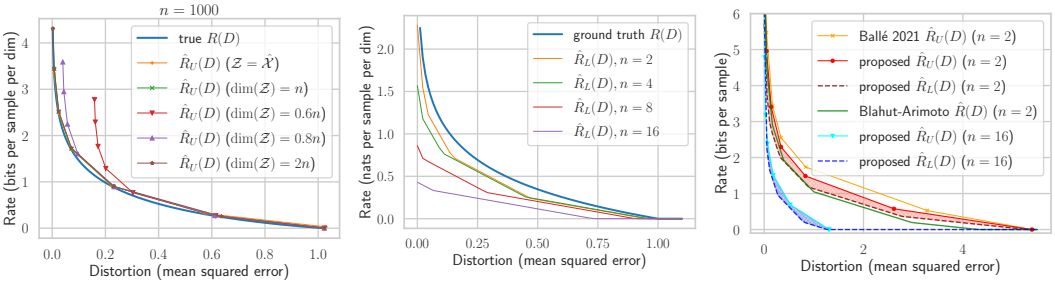

Figure 2: **Left**: R-D upper bound estimates on a randomly generated 1000-dimensional Gaussian source. **Middle**: R-D lower bound estimates on standard Gaussian sources with increasing dimensions. **Right**: Various R-D estimates on the 2D banana-shaped source, as well as its projection to $\mathbb{R}^{16}$.

6 Experiments

We experiment on four types of sources. On **random Gaussian sources**, we show that our upper bound algorithm can converge to the *exact* R-D function; our lower bounds, however, become increasingly loose in higher dimensions, and our experiments with varying k offers some insight into this issue. We obtain tight sandwich bounds on a **banana-shaped source** (Ballé et al., 2021) and its higher-dimension projections, demonstrating that our lower bound algorithm can handle high-dimension data with a low *intrinsic* dimension. We further confirm this by obtaining sandwich bounds on **realistic GAN-generated images** with varying intrinsic dimension, and use them to assess the performance of common learned image compression methods. Finally, we estimate bounds on the R-D function of **natural images**. The intrinsic dimension is still too high for our lower bounds to be useful. Our upper bounds, on the other hand, suggest one dB (in PSNR) of theoretical improvement in the compression quality of state-of-the-art image compression methods, at various bitrates. We provide extensive experimental details and additional results in Appendix sections A.5 and A.6.

6.1 Gaussian sources

We start by testing our algorithms on the factorized Gaussian distribution, one of the few sources with an analytical R-D function. We randomly generate the Gaussian sources in increasing dimensions.

For the upper bound algorithm, we let $Q_{\hat{X}}$ and $Q_{\hat{X}|X}$ be factorized Gaussians with learned parameters, predicting the parameters of $Q_{\hat{X}|X}$ by a 1-layer MLP encoder. As shown in Fig. 2-Left, on a $n = 1000$ dimensional Gaussian (the results are similar across all the n we tested), our upper bound (**yellow** curve) accurately recovers the analytical R-D function. We also considered optimizing the variational distributions in a latent space $\mathcal{Z}$ with varying dimensions, using an MLP decoder to map from $\mathcal{Z}$ to $\hat{\mathcal{X}}$ (see Sec. 3). The resulting bounds show similarly good agreement with the ground truth R-D function when the latent dimension matches or exceeds the data dimension n (**green**, **brown**), but become increasingly loose as the latent dimension drops below n (**red** and **purple** curves), demonstrating the importance of a rich enough latent space for a tight R-D bound, as suggested by our Theorem A.3.

For the lower bound algorithm, we parameterize $\log u$ by a 2-layer MLP, and study the effect of source dimension n and the number of samples k used in our estimator C_k (and objective $\tilde{\ell}_k$). To simplify comparison of R-D results across different source dimensions, here we consider standard Gaussian sources, whose R-D curve does not vary with n if we scale the rate by $\frac{1}{n}$ (i.e., rate per sample per dimension); the results on randomly generated Gaussians are similar. First, we fix $k = 1024$ and examine the resulting lower bounds; as shown in Fig. 2-Middle, the bounds quickly become loose with increasing source dimension. This is likely due to the over-estimation bias of our estimator C_k for the sup-partition function, which causes under-estimation in the objective $\tilde{\ell}_k$. While C_k is defined similarly to an M-estimator (Van der Vaart, 2000), analyzing its convergence behavior is not straightforward, as it depends on the function u being learned. Empirically, we observe the bias of C_k is amplified by an increase in the source dimension n or λ, such that an increasingly large k is required to effectively train a u model. In additional experiments, we found that for Gaussian sources, an increasing large k (seemingly exponential in n) is needed to reduce the gap in the lower bound. See results in Fig. 4, as well as a detailed discussion of this issue, in Appendix Sec. A.5.2.

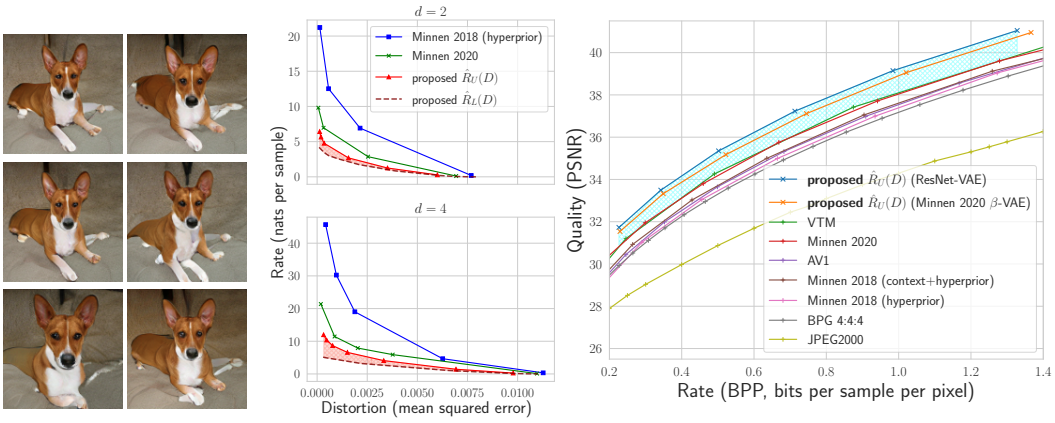

Figure 3: **Left**: 128×128 GAN-generated images, with intrinsic dimension $d = 4$. **Middle**: Bounds on $R(D)$ of GAN images, for $d = 2$ (top) and $d = 4$ (bottom). **Right**: Quality-rate curves of ours and state-of-the-art image compression methods on Kodak (1993), corresponding to R-D upper bounds.

6.2 High-dimension Sources with Low Intrinsic Dimension

The quickly deteriorating lower bound on higher-dimension Gaussians makes our goal of sandwiching the $R(D)$ of a general source seem hopeless. However, real-world data often contains much more structure than Gaussian noise. Growing research points to the low *intrinsic* dimension of real-world data such as natural images, and identifies it as a key to the success of learning algorithms (Pope et al., 2021). Echoing these findings, our experiments demonstrate that we can, in fact, obtain tightly matching lower and upper bounds on high-dimension data with a sufficiently low *intrinsic* dimension.

First, we consider the 2D banana-shaped source from Ballé et al. (2021). Our upper bound algorithm used the same MLP autoencoder as the compression model of Ballé et al. (2021), and parameterized the variaitonal distributions by normalizing flows. Our lower bound algorithm used a simple 3-layer MLP for $\log u$. As shown in Fig. 2-Right, our R-D upper bound (**red**) lies below the operational R-D curve of Ballé et al. (2021) (**orange**), and on average of 0.18 bits above our lower bound (**maroon**), with the sandwiched region shaded in red. The BA algorithm (**green**) largely agrees with our R-D estimates, but likely underestimates the true R-D in the high distortion region due to the finite discretization region. Next, we mapped the banana source to increasingly high dimensions by a randomly sampled linear transform. The BA algorithm is no longer feasible, even for $n = 4$. As shown in Fig. 2-Right, we still obtain tight sandwich bounds in $n = 16$ dimension (**blue**); the results are similar for all other n we tested, up to 1000. Unlike in the Gaussian experiment, where increasing n required (seemingly exponentially) larger k for a good lower bound, here a constant $k = 2048$ worked well for all n. The difference is that despite the high dimension, here the data still lies on a low-dimension manifold of the ambient space, a key property that has been observed to facilitate learning from data (Pope et al., 2021).

Following this observation, we experimented on GAN-generated images with controlled intrinsic dimension, and obtain R-D sandwich bounds to assess the performance of actual image compression algorithms. We borrowed the setup from Pope et al. (2021), and experimented on 128×128 images of `basenji` generated by a BigGAN. The intrinsic dimension d of the images are controlled by zeroing out all except d dimensions of the noise input to the GAN. As shown in Fig. 3-Left, the images appear highly realistic, showing the dog with subtly different body features and in various poses. We implemented a 6-layer ResNet-VAE (Kingma et al., 2016) for our upper bound model, and a simple convolutional network for our lower bound model. Fig. 3-Middle plots our resulting R-D bounds and the sandwiched region (**red**), along with the operational R-D curves (**blue, green**) of two learned image compression methods (Minnen et al., 2018; Minnen & Singh, 2020) trained on the GAN images, for $d = 2$ and $d = 4$. We see that despite the high dimensionality ($n = 128 \times 128 \times 3$), the images require few nats to compress; e.g., for $d = 4$, we estimate $R(D)$ to be between ~ 4 and 8 nats per sample when $D = 0.001$ (30 dB in PSNR). Notably, the R-D curve of Minnen & Singh (2020) stays roughly within a factor of 2 above our estimated true $R(D)$ region. Both neural compression methods show improved performance as d decreases from 4 to 2, following the same

trend as our R-D bounds. This demonstrates the ability of learned compression methods to exploit the low intrinsic dimension of data, contributing to its success over traditional methods such as JPEG (whose R-D curve on this data does not appear to vary with d, and lies orders of magnitude higher).

6.3 Natural Images

To establish upper bounds on the R-D function of natural images, we define variational distributions $(Q_Z, Q_{Z|X})$ on a Euclidean latent space for simplicity, and parameterize them as well as a learned decoder ω via hierarchical VAEs. We borrow the convolutional autoencoder architecture of a state-of-the-art image compression model (Minnen & Singh, 2020), and set the variational distributions to be factorized Gaussians with learned means and variances (we still use the deep factorized hyperprior, but no longer convolve it with a uniform prior). We also reuse our (β-)ResNet-VAE model with 6 layers of latent variables from the GAN experiments (Sec. 6.2). We trained models with mean-squared error (MSE) distortion and various λ on the COCO 2017 (Lin et al., 2014) images, and evaluated them on the Kodak (1993) and Tecnick (Asuni & Giachetti, 2014) datasets. Following image compression conventions, we report the rate in bits-per-pixel, and the quality (i.e., negative distortion) in PSNR averaged over the images for each (λ, model) pair [2]. The resulting *quality-rate* (Q-R) curves can be interpreted as giving upper bounds on the R-D functions of the image-generating distributions. We plot them in Fig. 3, along with the Q-R performance (in actual bitrate) of various traditional and learned image compression methods (Ballé et al., 2017; Minnen et al., 2018; Minnen & Singh, 2020), for the Kodak dataset (see similar results on Tecnick in Appendix Fig. 9). Our β-VAE model based on (Minnen & Singh, 2020) (**orange**) lies on average 0.7 dB higher than the operational Q-R curves of the original model (**red**) and VTM (**green**). With a deeper latent hierarchy, our (β-)ResNet-VAE gives a higher Q-R curve (**blue**) that lies on average 1 dB above the state-of-the-art Q-R curves (gap shaded in blue). We leave it to future work to investigate which choice of autoencoder architecture and variational distributions are most effective, as well as how the theoretical R-D performance of such a β-VAE can be realized by a practical compression algorithm (see discussions in Sec. 5).

7 Discussions

In this work, we proposed machine learning techniques to computationally bound the rate-distortion function of a data source, a key quantity that characterizes the fundamental performance limit of all lossy compression algorithms, but is largely unknown. Departing from prior work in the information theory community (Gibson, 2017; Riegler et al., 2018), our approach applies broadly to general data sources and requires only i.i.d. samples, making it more suitable for real-world application.

Our upper bound method is a gradient descent version of the classic Blahut-Arimoto algorithm, and closely relates to (and extends) variational autoencoders from neural lossy compression research. Our lower bound method optimizes a dual characterization of the R-D function, which has been known for some time but seen little application outside of theoretical work. Due to difficulties involving global optimization, our lower bound currently requires a squared error distortion for tractability in the continuous case, and is only tight on data sources with a low *intrinsic* dimension. We hope that a better understanding of the lower bound problem will lead to improved algorithms in the future.

To properly interpret bounds on the R-D function, we emphasize that the significance of the R-D function is two-fold: 1. for a given distortion tolerance D, no coding procedure can operate with a rate less than $R(D)$, and that 2. this rate is asymptotically achievable by some (potentially expensive) procedure. Therefore, a lower bound makes a universal statement about what kind of rate-distortion performance is "too good to be true". The story is more subtle for the upper bound, due to the asymptotic nature of $R(D)$. The achievability proof relies on a random coding procedure that jointly compresses multiple data samples in increasingly long blocks (Shannon, 1959). When compressing at a finite block length b (e.g., $b = 1$ when compressing individual images), $R(D)$ is generally no longer achievable, due to a rate overhead that scales like $b^{-\frac{1}{2}}$ (Kostina & Verdú, 2012). Extending our work to the setting of finite block lengths could be another useful future direction.

[2] Technically, to compute an R-D upper bound with MSE ρ, the distortion needs to be evaluated by averaging MSE (instead of PSNR) on samples.

ETHICS AND REPRODUCIBILITY STATEMENTS

As our work deals with theoretical aspects of lossy data compression, we are not aware of any potential ethical implications. Our code and instructions for reproducing the datasets and experimental results can be found in this anonymous repo https://drive.google.com/drive/folders/1uI3q7sVU6T8iF2PoAjNXSAwVHpYLeyHC?usp=sharing.

REFERENCES

E. Agustsson and L. Theis. Universally Quantized Neural Compression. In *Advances in Neural Information Processing Systems 33*, 2020.

Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8503–8512, 2020.

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken ELBO. In *International Conference on Machine Learning*, pp. 159–168. PMLR, 2018.

Suguru Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, 1972.

N. Asuni and A. Giachetti. TESTIMAGES: A large-scale archive for testing visual devices and basic image processing algorithms (SAMPLING 1200 RGB set). In *STAG: Smart Tools and Apps for Graphics*, 2014. URL https://sourceforge.net/projects/testimages/files/OLD/OLD_SAMPLING/testimages.zip.

J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end Optimized Image Compression. In *International Conference on Learning Representations*, 2017.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational Image Compression with a Scale Hyperprior. *ICLR*, 2018.

J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici. Nonlinear transform coding. *IEEE Trans. on Special Topics in Signal Processing*, 15, 2021.

T Berger. Rate distortion theory, a mathematical basis for data compression (prentice-hall. *Inc. Englewood Cliffs, New Jersey*, 1971.

Toby Berger and Jerry D Gibson. Lossy source coding. *IEEE Transactions on Information Theory*, 44(6):2693–2723, 1998.

R. Blahut. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory*, 18(4):460–473, 1972. doi: 10.1109/TIT.1972.1054855.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2019.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Miguel A. Carreira-Perpinan. Mode-finding for mixtures of gaussian distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1318–1323, 2000.

Miguel A. Carreira-Perpinan. Gaussian mean-shift is an em algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):767–776, 2007. doi: 10.1109/TPAMI.2007.1057.

Miguel A. Carreira-Perpinan. How many modes can a Gaussian mixture have, 2020. URL https://faculty.ucmerced.edu/mcarreira-perpinan/research/GMmodes.html.

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. *arXiv preprint arXiv:2001.01568*, 2020.

Mung Chiang and Stephen Boyd. Geometric programming duals of channel capacity and rate distortion. *IEEE Transactions on Information Theory*, 50(2):245–258, 2004.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*, volume 2. John Wiley & Sons, 2006.

I. Csiszar. On the computation of rate-distortion functions (corresp.). *IEEE Transactions on Information Theory*, 20(1):122–124, 1974. doi: 10.1109/TIT.1974.1055146.

Imre Csiszár. On an extremum problem of information theory. *Studia Scientiarum Mathematicarum Hungarica*, 9, 01 1974.

G. Flamich, M. Havasi, and J. M. Hernández-Lobato. Compressing Images by Encoding Their Latent Representations with Relative Entropy Coding, 2020. Advances in Neural Information Processing Systems 34.

Brendan J. Frey and Geoffrey E. Hinton. Efficient stochastic source coding and an application to a bayesian network source model. *The Computer Journal*, 40(2_and_3):157–165, 1997.

Jerry Gibson. Rate distortion functions and rate distortion function lower bounds for real-world sources. *Entropy*, 19(11):604, 2017.

Robert M Gray. *Entropy and information theory*. Springer Science & Business Media, 2011.

Matthew T. Harrison and Ioannis Kontoyiannis. Estimation of the rate–distortion function. *IEEE Transactions on Information Theory*, 54(8):3757–3762, Aug 2008. ISSN 0018-9448. doi: 10.1109/tit.2008.926387. URL http://dx.doi.org/10.1109/TIT.2008.926387.

J. Hayes, A. Habibi, and P. Wintz. Rate-distortion function for a gaussian source model of images (corresp.). *IEEE Transactions on Information Theory*, 16(4):507–509, 1970. doi: 10.1109/TIT.1970.1054496.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2(5):6, 2017.

Jonathan Ho, Evan Lohn, and Pieter Abbeel. Compression with flows via local bits-back coding. In *Advances in Neural Information Processing Systems*, pp. 3874–3883, 2019.

Emiel Hoogeboom, Jorn Peters, Rianne van den Berg, and Max Welling. Integer discrete flows and lossless compression. In *Advances in Neural Information Processing Systems*, pp. 12134–12144, 2019.

Sicong Huang, Alireza Makhzani, Yanshuai Cao, and Roger Grosse. Evaluating lossy compression rates of deep generative models. *International Conference on Machine Learning*, 2020.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

D. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pp. 4743–4751, 2016.

Friso H Kingma, Pieter Abbeel, and Jonathan Ho. Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables. *arXiv preprint arXiv:1905.06845*, 2019.

Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, Nov 2021. ISSN 1939-3539. doi: 10.1109/tpami.2020.2992934. URL http://dx.doi.org/10.1109/TPAMI.2020.2992934.

Kodak. PhotoCD PCD0992, 1993. URL http://r0k.us/graphics/kodak/.

Victoria Kostina. When is shannon's lower bound tight at finite blocklength? In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 982–989. IEEE, 2016.

Victoria Kostina and Sergio Verdú. Fixed-length lossy compression in the finite blocklength regime. *IEEE Transactions on Information Theory*, 58(6):3309–3338, 2012.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Jasper C. H. Lee, Jerry Li, Christopher Musco, Jeff M. Phillips, and Wai Ming Tai. Finding the mode of a kernel density estimate, 2019.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Matt Mahoney. Rationale for a large text compression benchmark. *Retrieved (Oct. 1st, 2021) from: http://mattmahoney.net/dc/rationale.html*, 2009.

Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2): 583–601, 2002.

D. Minnen and S. Singh. Channel-wise autoregressive entropy models for learned image compression. In *IEEE International Conference on Image Processing (ICIP)*, 2020.

D. Minnen, J. Ballé, and G. D. Toderici. Joint Autoregressive and Hierarchical Priors for Learned Image Compression. In *Advances in Neural Information Processing Systems 31*. 2018.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pp. 2338–2347, 2017.

Y Polyanskiy and Y Wu. Lecture notes on information theory. 2014.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5171–5180. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/poole19a.html.

Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XJk19XzGq2J.

Farzad Rezaei, NU Ahmed, and Charalambos D Charalambous. Rate distortion theory for general sources with potential application to image compression. *International Journal of Applied Mathematical Sciences*, 3(2):141–165, 2006.

Erwin Riegler, Günther Koliander, and Helmut Bölcskei. Rate-distortion theory for general sets and measures. *arXiv preprint arXiv:1804.08980*, 2018.

Yangjun Ruan, Karen Ullrich, Daniel Severo, James Townsend, Ashish Khisti, Arnaud Doucet, Alireza Makhzani, and Chris J Maddison. Improving lossless compression rates via monte carlo bits-back coding. In *International Conference on Machine Learning*, 2021.

Ruslan Salakhutdinov. Learning deep generative models. *Annual Review of Statistics and Its Application*, 2:361–385, 2015.

T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. PixelCNN++: A pixelcnn implementation with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.

C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27: 379–423, 1948.

CE Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec., March 1959*, 4:142–163, 1959.

L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy Image Compression with Compressive Autoencoders. In *International Conference on Learning Representations*, 2017.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

James Townsend, Tom Bird, and David Barber. Practical lossless compression with latent variables using bits back coding. *arXiv preprint arXiv:1901.04866*, 2019.

A. van den Oord, N. Kalchbrenner, O. Vinyals, A. Graves L. Espeholt, and K. Kavukcuoglu. Conditional Image Generation with PixelCNN Decoders. In *Advances in Neural Information Processing Systems 29*, pp. 4790–4798, 2016.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Ian H Witten, Radford M Neal, and John G Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.

Ruihan Yang, Yibo Yang, Joseph Marino, and Stephan Mandt. Hierarchical autoregressive modeling for neural video compression. In *International Conference on Learning Representations*, 2020a.

Yibo Yang, Robert Bamler, and Stephan Mandt. Improving inference for neural image compression. In *Neural Information Processing Systems (NeurIPS), 2020*, 2020b.

Yibo Yang, Robert Bamler, and Stephan Mandt. Variational Bayesian Quantization. In *International Conference on Machine Learning*, 2020c.

Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 23(3):337–343, 1977.

A APPENDIX

A.1 TECHNICAL DEFINITIONS AND PREREQUISITES

In this work we consider the source to be represented by a random variable $X : \Omega \to \mathcal{X}$, i.e., a measurable function on an underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and P_X is the image measure of $\mathbb{P}$ under X. We suppose the source and reproduction spaces are standard Borel spaces, $(\mathcal{X}, \mathcal{A}_{\mathcal{X}})$ and $(\hat{\mathcal{X}}, \mathcal{A}_{\hat{\mathcal{X}}})$, equipped with sigma-algebras $\mathcal{A}_{\mathcal{X}}$ and $\mathcal{A}_{\hat{\mathcal{X}}}$, respectively. Below we use the definitions of standard quantities from Polyanskiy & Wu (2014).

Conditional distribution The notation $Q_{\hat{X}|X}$ denotes an arbitrary conditional distribution (also known as a Markov kernel), i.e., it satifies

1. For any $x \in \mathcal{X}$, $Q_{\hat{X}|X=x}(\cdot)$ is a probability measure on $\hat{\mathcal{X}}$;
2. For any measurable set $B \in \mathcal{A}_{\hat{\mathcal{X}}}$, $x \to Q_{\hat{X}|X=x}(B)$ is a measurable function on $\mathcal{X}$.

Induced joint and marginal measures Given a source distribution P_X, each test channel distribution $Q_{\hat{X}|X}$ defines a joint distribution $P_X Q_{\hat{X}|X}$ on the product space $\mathcal{X} \times \hat{\mathcal{X}}$ (equipped with the usual product sigma algebra, $\mathcal{A}_{\mathcal{X}} \times \mathcal{A}_{\hat{\mathcal{X}}}$) as follows:

$$P_X Q_{\hat{X}|X}(E) := \int_{\mathcal{X}} P_X(dx) \int_{\hat{\mathcal{X}}} \mathbf{1}\{(x, \hat{x}) \in E\} Q_{\hat{X}|X=x}(d\hat{x}),$$

for all measurable sets $E \in \mathcal{A}_{\mathcal{X}} \times \mathcal{A}_{\hat{\mathcal{X}}}$. The induced $\hat{x}$-marginal distribution $P_{\hat{X}}$ is then defined by

$$P_{\hat{X}}(B) = \int_{\mathcal{X}} Q_{\hat{X}|X=x}(B) P_X(dx),$$

for all measurable sets $\forall B \in \mathcal{A}_{\hat{\mathcal{X}}}$.

KL Divergence We use the general definition of Kullback-Leibler (KL) divergence between two probability measures P, Q defined on a common measurable space:

$$D(P\|Q) := \begin{cases} \int \log \frac{dP}{dQ} dP, & \text{if } P \ll Q \\ \infty, & \text{otherwise.} \end{cases}$$

$P \ll Q$ denotes that P is absolutely continuous w.r.t. Q (i.e., for all measurable sets E, $Q(E) = 0 \implies P(E) = 0$). $\frac{dP}{dQ}$ denotes the Radon-Nikodym derivative of P w.r.t. Q; for discrete distributions, we can simply take it to be the ratio of probability mass functions; and for continuous distributions, we can simply take it to be the ratio of probability density functions.

Mutual Information Given P_X and $Q_{\hat{X}|X}$, the mutual information $I(X;Y)$ is defined as

$$I(X; \hat{X}) := D(P_X Q_{\hat{X}|X} \| P_X \otimes P_{\hat{X}}) = \mathbb{E}_{x \sim P_X}[D(Q_{\hat{X}|X=x} \| P_{\hat{X}})],$$

where $P_{\hat{X}}$ is the $\hat{x}$-marginal of the joint $P_X Q_{\hat{X}|X}$, $P_X \otimes P_{\hat{X}}$ denotes the usual product measure, and $D(\cdot\|\cdot)$ is the KL divergence.

For the mutual information upper bound, it's easy to show that

$$\mathcal{I}_U(Q_{\hat{X}|X}, Q_{\hat{X}}) := \mathbb{E}_{x \sim P_X}[KL(Q_{\hat{X}|X=x} \| Q_{\hat{X}})] = I(X; \hat{X}) + KL(P_{\hat{X}} \| Q_{\hat{X}}), \tag{10}$$

so the bound is tight when $P_{\hat{X}} = Q_{\hat{X}}$.

Obtaining $R(D)$ through the Lagrangian. For each $\lambda \geq 0$, we define the Lagrangian by incorporating the distortion constraint in the definition of $R(D)$ through a linear penalty:

$$\mathcal{L}(Q_{\hat{X}|X}, \lambda) := I(X; \hat{X}) + \lambda \mathbb{E}_{P_X Q_{\hat{X}|X}}[\rho(X, \hat{X})], \tag{11}$$

and define its infimum w.r.t. $Q_{\hat{X}|X}$ by the function

$$F(\lambda) := \inf_{Q_{\hat{X}|X}} I(X;\hat{X}) + \lambda \mathbb{E}[\rho(X,\hat{X})]. \tag{12}$$

Geometrically, $F(\lambda)$ is the maximum of the R-axis intercepts of straight lines of slope $-\lambda$, such that they have no point above the $R(D)$ curve (Csiszár, 1974).

Define $D_{min} := \inf\{D' : R(D') < \infty\}$. Since $R(D)$ is convex, for each $D > D_{min}$, there exists a $\lambda \geq 0$ such that the line of slope $-\lambda$ through $(D, R(D))$ is tangent to the $R(D)$ curve, i.e.,

$$R(D') + \lambda D' \geq R(D) + \lambda(D) = F(\lambda), \quad \forall D'.$$

When this occurs, we say that λ *is associated to* D.

Consequently, the $R(D)$ curve is then the envelope of lines with slope $-\lambda$ and R-axis intercept $F(\lambda)$. Formally, this can be stated as:

Lemma A.1. (Lemma 1.2, Csiszár (1974); Lemma 9.7, Gray (2011)). *For every distortion tolerance $D > D_{min}$, where $D_{min} := \inf\{D' : R(D') < \infty\}$, it holds that*

$$R(D) = \max_{\lambda \geq 0} F(\lambda) - \lambda D \tag{13}$$

We can draw the following conclusions:

1. *For each $D > D_{min}$, the maximum above is attained iff λ is associated to D.*
2. *For a fixed λ, if $Q^*_{Y|X}$ achieves the minimum of $\mathcal{L}(\cdot, \lambda)$, then λ is associated to the point $(\mathcal{I}(Q^*_{Y|X}), \rho(Q^*_{Y|X}))$; i.e., there exists a line with slope $-\lambda$ that is tangent to the $R(D)$ curve at $(\mathcal{I}(Q^*_{Y|X}), \rho(Q^*_{Y|X}))$.*

A.2 Full Version of Theorem 4.1

Theorem A.2. (Theorem 1, Kostina (2016).) *Suppose that the following basic assumptions are satisfied.*

1. *$R(D)$ is finite for some D, i.e., $D_{min} := \inf\{D : R(D) < \infty\} < \infty$;*
2. *The distortion metric ρ is such that there exists a finite set $E \subset \hat{\mathcal{X}}$ such that*

$$\mathbb{E}[\min_{\hat{x} \in E} \rho(X, \hat{x})] < \infty$$

Then, for each $D > D_{min}$, it holds that

$$R(D) = \max_{g(x), \lambda} \{\mathbb{E}[-\log g(X)] - \lambda D\} \tag{14}$$

where the maximization is over $g(x) \geq 0$ and $\lambda \geq 0$ satisfying the constraint

$$\mathbb{E}\left[\frac{\exp(-\lambda \rho(X, \hat{x}))}{g(X)}\right] = \int \frac{\exp(-\lambda \rho(x, \hat{x}))}{g(x)} dP_X(x) \leq 1, \forall \hat{x} \in \hat{\mathcal{X}} \tag{15}$$

Note: the basic assumption 2 is trivially satisfied when the distortion ρ is bounded from above; the maximization over $g(x) \geq 0$ can be restricted to only $1 \geq g(x) \geq 0$. Unless stated otherwise, we use $\log$ base e in this work, so the $R(D)$ above is in terms of nats (per sample).

Theorem A.2 can be seen as a consequence of Theorem 4.1 in conjunction with Lemma A.1. We implemented an early version of our lower bound algorithm based on Theorem A.2, generating each R-D lower bound by fixing a target D value and optimizing over both λ and g as in Equation 14. However, the algorithm often diverged due to drastically changing λ. We therefore based our current algorithm on Theorem 4.1, producing R-D lower bounds by fixing λ and only optimizing over g (or u, in our formulation).

A.3 THEORETICAL RESULTS

Theorem A.3. (A suitable β-VAE defines an upper bound on the source R-D function). *Let $X \sim P_X$ be a memoryless source under distortion ρ. Let $\mathcal{Z}$ be any measurable space ("latent space" in a VAE), and $\omega : \mathcal{Z} \to \hat{\mathcal{X}}$ any measurable function ("decoder" in a VAE). This induces a new lossy compression problem with $\mathcal{Z}$ being the reproduction alphabet, under a new distortion function $\rho_\omega : \mathcal{X} \times \mathcal{Z} \to [0, \infty), \rho_\omega(x, z) = \rho(x, \omega(Z))$. Define the corresponding rate-distortion function*

$$R_\omega(D) := \inf_{Q_{Z|X}:\mathbb{E}[\rho_\omega(X,Z)]\leq D} I(X; Z) = \inf_{Q_{Z|X}:\mathbb{E}[\rho(X,\omega(Z))]\leq D} I(X; Z).$$

Then for any $D \geq 0$, $R_\omega(D) \geq R(D)$. A sufficient condition for $D \geq 0$, $R_\omega(D) = R(D)$ is for ω to be bijective.

Proof. Fix D. Take any admissible conditional distribution $Q_{Z|X}$ that satisfies $\mathbb{E}[\rho(X, \omega(Z))] \leq D$ in the definition of $R_\omega(D)$. Define a new kernel $Q_{\hat{X}|X}$ between $\mathcal{X}$ and $\hat{\mathcal{X}}$ by $Q_{\hat{X}|X=x} := Q_{Z|X=x} \circ \omega^{-1}, \forall x \in \mathcal{X}$, i.e., $Q_{\hat{X}|X=x}$ is the image measure of $Q_{Z|X=x}$ induced by ω. Applying data processing inequality to the Markov chain $X \xrightarrow{Q_{Z|X}} Z \xrightarrow{\omega} \hat{X}$, we have $I(X; Z) \geq I(X; \hat{X})$.

Moreover, since $Q_{\hat{X}|X}$ is admissible in the definition of $R(D)$, i.e.,

$$\mathbb{E}_{P_X Q_{\hat{X}|X}}[\rho(X, \hat{X})] = \mathbb{E}_{P_X Q_{Z|X}}[\rho(X, \omega(Z))] \leq D$$

we therefore have

$$I(X; Z) \geq I(X; \hat{X}) \geq R(D) = \inf_{Q_{\hat{X}|X}:\mathbb{E}[\rho(X,\hat{X})]\leq D} I(X; \hat{X}).$$

Finally, since $I(X; Z) \geq R(D)$ holds for any admissible $Q_{Z|X}$, taking infimum over such $Q_{Z|X}$ gives $R_\omega(D) \geq R(D)$.

To prove $R_\omega(D) = R(D)$ when ω is bijective, it suffices to show that $R(D) \geq R_\omega(D)$. We use the same argument as before. Take any admissible $Q_{\hat{X}|X}$ in the definition of $R(D)$. We can then construct a $Q_{Z|X}$ by the process $X \xrightarrow{Q_{\hat{X}|X}} \hat{X} \xrightarrow{\omega^{-1}} Z$. Then by DPI we have $I(X; \hat{X}) \geq I(X; Z)$. Morever, $Q_{Z|X}$ is admissible: $\mathbb{E}_{P_X Q_{Z|X}}[\rho(X, \omega(Z))] = \mathbb{E}_{P_X Q_{\hat{X}|X}}[\rho(X, \omega(\omega^{-1}(\hat{X}))] = \mathbb{E}_{P_X Q_{\hat{X}|X}}[\rho(X, \hat{X})] \leq D$. So $I(X; \hat{X}) \geq I(X; Z) \geq R_\omega(D)$. Taking infimum over such $Q_{\hat{X}|X}$ concludes the proof.

□

Remarks: Although $\omega^{-1}(\omega(\hat{X})) = \hat{X}$ for an injective ω, we needed $\omega(\omega^{-1}(\hat{X})) = \hat{X}$ in the proof of the other direction, which requires ω^{-1} to always be well-defined. Several learned image compression methods have advocated for the use of sub-pixel convolution, i.e, convolution followed by (invertible) reshaping of the results, over upsampled convolution in the decoder, in order to produce better reconstructions (Theis et al., 2017; Cheng et al., 2020). This can be seen as making the decoder more bijective, therefore reducing the gap of $R_\omega(D)$ over $R(D)$, in light of our above theorem.

Theorem A.4. (Basic properties of the proposed estimator C_k of the sup-partition function.) *Let $X_1, X_2, ... \sim P_X$ be a sequence of i.i.d. random variables. Let $\psi : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}$ be a measurable function. For each k, define the random variable $C_k := \sup_{\hat{x}} \frac{1}{k} \sum_i \psi(X_i, \hat{x})$. Then*

1. *C_k is an overestimator of the sup-partition function c, i.e.,*
   $\mathbb{E}[C_k] = \mathbb{E}_{X_1,...,X_k}[\sup_{\hat{x}} \frac{1}{k} \sum_i \psi(X_i, \hat{x})] \geq \sup_{\hat{x}} \mathbb{E}[\psi(X, \hat{x})] =: c$;
2. *The bias of C_k decreases with increasing k, i.e.,*
   $\mathbb{E}[C_1] \geq \mathbb{E}[C_2] \geq ... \geq \mathbb{E}[C_k] \geq \mathbb{E}[C_{k+1}] \geq ... \sup_{\hat{x}} \mathbb{E}[\psi(X, \hat{x})] = c$;
3. *If $\psi(x, \hat{x})$ is bounded and continuous in $\hat{x}$, and if $\hat{\mathcal{X}}$ is compact, then C_k is strongly consistent, i.e., C_k converges to c almost surely (as well as in probability, i.e., $\lim_{k\to\infty} \mathbb{P}(|C_k - c| > \epsilon) = 0, \forall \epsilon > 0$), and $\lim_{k\to\infty} \mathbb{E}[C_k] = c$.*

Proof. We prove each in turn:

1. $\mathbb{E}[C_k] = \mathbb{E}[\sup_{\hat{x}} \frac{1}{k}\sum_i \psi(X_i, \hat{x})] \geq \sup_{\hat{x}} \mathbb{E}[\frac{1}{k}\sum_i \psi(X_i, \hat{x})] = \sup_{\hat{x}} \mathbb{E}[\psi(X, \hat{x})] = c$
2. First, note that $\mathbb{E}[C_1] \geq \mathbb{E}[C_k]$ since

$$\mathbb{E}[C_1] = \mathbb{E}[\sup_{\hat{x}} \psi(X_1, \hat{x})] = \mathbb{E}[\frac{1}{k}\sum_i \sup_{\hat{x}} \psi(X_i, \hat{x})] \geq \mathbb{E}[\sup_{\hat{x}} \frac{1}{k}\sum_i \psi(X_i, \hat{x})] = \mathbb{E}[C_k]$$

We therefore have

$$\begin{aligned}
\mathbb{E}[C_{k+1}] &= \mathbb{E}[\sup_{\hat{x}} \frac{1}{k+1}\sum_{i=1}^{k+1} \psi(X_i, \hat{x})] \\
&= \mathbb{E}[\sup_{\hat{x}} \{\frac{1}{k+1}\sum_{i=1}^{k} \psi(X_i, \hat{x}) + \frac{1}{k+1}\psi(X_{k+1}, \hat{x})\}] \\
&\leq \mathbb{E}[\sup_{\hat{x}} \{\frac{1}{k+1}\sum_{i=1}^{k} \psi(X_i, \hat{x})\} + \sup_{\hat{x}} \{\frac{1}{k+1}\psi(X_{k+1}, \hat{x})\}] \\
&= \frac{k}{k+1}\mathbb{E}[C_k] + \frac{1}{k+1}\mathbb{E}[C_1] \\
&\leq \mathbb{E}[C_k]
\end{aligned}$$

3. The proof for this resembles that of Theorem 1 of Burda et al. (2015). We use standard arguments from probability theory and real analysis. Fix $\hat{x} \in \hat{\mathcal{X}}$, and consider the random variable $M_k = \frac{1}{k}\sum_{i=1}^{k} \psi(X_i, \hat{x})$. If ψ is bounded, then it follows from the Strong Law of Large Numbers that M_k converges to $\mathbb{E}[M_1] = \mathbb{E}[\psi(X, \hat{x})]$ almost surely; in other words, for every ω outside a set of measure zero,

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \psi(X_i(\omega)), \hat{x}) = \mathbb{E}[\psi(X(\omega), \hat{x})],$$

Then, for every such ω

$$\lim_{k\to\infty} \sup_{\hat{x}} \frac{1}{k}\sum_{i=1}^{k} \psi(X_i(\omega)), \hat{x}) = \sup_{\hat{x}} \lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \psi(X_i(\omega)), \hat{x}) = \sup_{\hat{x}} \mathbb{E}[\psi(X(\omega), \hat{x})],$$

where we used the fact that the sequence of continuous functions $\gamma_k(\hat{x}) := \frac{1}{k}\sum_{i=1}^{k} \psi(X_i(\omega)), \hat{x})$ converges pointwise to $\gamma(\hat{x}) := \mathbb{E}[\psi(X(\omega), \hat{x})]$ on a compact set $\hat{\mathcal{X}}$, so γ_k converges to γ also uniformly, so we are allowed to exchange limit and supremum, i.e., $\lim_{k\to\infty} \sup_{\hat{x}} \gamma_k(\hat{x}) = \sup_{\hat{x}} \lim_{k\to\infty} \gamma_k(\hat{x}) = \sup_{\hat{x}} \gamma(\hat{x})$. But the above equation precisely means that C_k converges to $c = \sup_{\hat{x}} \mathbb{E}[\psi(X, \hat{x})]$ almost surely. Therefore C_k also converges to c in probability, and $\lim_{k\to\infty} \mathbb{E}[C_k] = c$.

□

A.4 Proposed Lower Bound Algorithm

We give a pseudo-code implementation of the algorithm in Algorithm 1. The code largely follows Python semantics, and assumes an auto-differentiation package is available, such as GradientTape as provided in `Tensorflow`. We use γ_k to denote the function in the supremem definition of C_k , i.e., $\gamma_k(\hat{x}) := \frac{1}{k}\sum_{i=1}^{k} \exp\{-\lambda\rho(x_i, \hat{x})\}/u_\theta(x_i)$.

For a given $\lambda > 0$ and model $u_\theta : x \to \mathbb{R}^+$, the lower bound algorithm works by (stochastic) gradient ascent on the objective $\tilde{\ell}_k(\theta)$ (Eq. 9) w.r.t. the model parameters θ. To compute the gradient, we need an estimate of $\mathbb{E}[C_k]$ (ultimately $\log \mathbb{E}[C_k]$) as part of the objective, which requires us to draw m samples of C_k. In the pseudo-code, a separate set of m samples of C_k are also drawn to set the linear

Algorithm 1: Example implementation of the proposed algorithm for estimating rate-distortion lower bound $R_L(D)$.

1 **Requires:** $\lambda > 0$, model u_θ (e.g., a neural network) parameterized by θ, batch sizes k, m, and gradient ascent step size ϵ.
2 **while** $\tilde{\ell}_k$ *not converged* **do**
    `// Draw` $2m$ `samples of` C_k`; use the last` m `samples to set` α`, and the first` m `samples to estimate` $\mathbb{E}[C_k]$ `in` $\tilde{\ell}_k$`.`
3 **for** $j := 1$ **to** $2m$ **do**
4 Draw k data samples, $\{x_1^j, ..., x_k^j\}$
5 $\hat{x}^j, C_k^j :=$ compute_$C_k(\theta, \lambda, \{x_1, ..., x_k\})$
6 **end**
7 $\alpha := \frac{1}{m} \sum_{j=m+1}^{2m} C_k^j$
    `// Compute objective` $\tilde{\ell}_k(\theta)$ `and update` θ `by gradient ascent.`
8 with GradientTape() as tape:
9 $E := \frac{1}{m} \sum_{j=1}^{m} \gamma_k(\hat{x}^j, \theta, \{x_1^j, ..., x_k^j\})$ `// Estimate of` $\mathbb{E}[C_k]$
10 $\tilde{\ell}_k := -\frac{1}{m} \sum_{j=1}^{m} \frac{1}{k} \sum_{i=1}^{k} \log u_\theta(x_i^j) - \frac{1}{\alpha} E - \log \alpha + 1$
11 gradient := tape.gradient($\tilde{\ell}_k$, θ)
12 $\theta := \theta + \epsilon$ gradient
13 **end**
14 **Subroutine** $\gamma_k(\hat{x}, \theta, \lambda, \{x_1, ..., x_k\})$:
    `// Evaluate the global maximization objective at` $\hat{x}$`.`
15 Return $\frac{1}{k} \sum_{i=1}^{k} \exp\{-\lambda \rho(x_i, \hat{x})\} / u_\theta(x_i)$
16 **Subroutine** compute_$C_k(\theta, \lambda, \{x_1, ..., x_k\})$:
    `// Compute the global optimum of` $\gamma_k(\hat{x})$`, assuming squared distortion` $\rho(x, \hat{x}) \propto \|x - \hat{x}\|^2$`.`
17 opt_loss := $-\infty$
18 **for** $i := 1$ **to** k **do**
    `// Run gradient ascent from the` i`th mixture component.`
19 $\hat{x} := x_i$
20 **while** $\hat{x}$ *not converged* **do**
21 with GradientTape() as tape:
22 loss := $\gamma_k(\hat{x}, \theta, \lambda, \{x_1, ..., x_k\})$
23 gradient := tape.gradient(loss, $\hat{x}$)
24 $\hat{x} := \hat{x} + \epsilon$ gradient
25 **end**
26 **if** *loss* $>$ *opt_loss* **then**
27 opt_loss := loss
28 $\hat{x}^* := \hat{x}$
29 **end**
30 **end**
31 C_k = opt_loss
32 return $(\hat{x}^*, C_k)$

expansion point α. In our actual implementation, we set α to an exponential moving average of $\mathbb{E}[C_k]$ estimated from previous iterations (e.g., replacing line 7 of the pseudo-code by $\alpha := 0.2\alpha + 0.8E$), so only m samples of C_k need to be drawn for each iteration of the algorithm. We did not find this approximation to significantly affect the training or results.

Recall $C_k(\theta)$ is defined by the result of maximizing w.r.t. $\hat{x}$. When computing the gradient with respect to θ, we differentiate through the maximization operation by evaluating $C_k = \gamma_k(\hat{x}^*)$ on the forward pass, using the argmax $\hat{x}^*$ found by the subroutine compute_C_k. This is justified by appealing to a standard envelope theorem (Milgrom & Segal, 2002).

By Lemma A.1, each u_θ^* trained with a given value of λ yields a linear under-estimator of $R(D)$,

$$R_L^\lambda(D) = -\lambda D + \tilde{\ell}_k(\theta^*).$$

We obtain the final $R_L(D)$ lower bound by taking the upper envelope of all such lines, i.e.,

$$R_L(D) := \max_{\lambda \in \Lambda} R_L^\lambda(D),$$

where Λ is the set of λ values we trained with.

Tips and tricks:

To avoid numerical issues, we always parameterize $\log u$, and all the operations involving C_k and α are performed in the log domain; e.g., we optimize $\log \gamma_k$ instead of γ_k (which does not affect the result since $\log$ is monotonically increasing), and use $\mathrm{logsumexp}$ when taking the average of multiple C_k samples in the log domain.

During training, we only run an approximate version of the global optimization subroutine compute_C_k for efficiency. Instead of hill-climbing from each of the k mixture component centroids, we only do so from the t highest-valued centroids under the objective γ_k, for a small number of t (say 10). This approximation is not used when reporting the final results.

A.5 Additional Experimental Details and Results

A.5.1 Computational Aspects

Our methods are implemented using the `Tensorflow` library. Our experiments with learned image compression methods used the `tensorflow-compression`[3] library. Our experiments on Gaussian and banana-shaped sources were run on Intel(R) Xeon(R) CPUs, and experiments on images were run on Titan RTX GPUs. We used the Adam optimizer for gradient based optimization in all our experiments, typically setting the learning rate to $1e-4$. Training the β-VAEs for the upper bounds required from a few thousand gradient steps on the lower-dimension problems (under an hour), to a few million gradient steps on the image compression problem (a couple of days; similar to reported in Minnen & Singh (2020)). With our approximate mode finding procedure (starting hill climbing only from a small number of centroids, see Sec. A.4), the lower bound models only required a few thousand steps to train, even on the image experiments (under a day).

A.5.2 Gaussians

Data Here the data source is an n-dimensional Gaussian distribution with a diagonal covariance matrix, whose parameters are generated randomly. We sample each dimension of the mean uniformly randomly from $[-0.5, 0.5]$ and variance from $[0, 2]$. The ground-truth R-D function of the source is computed analytically by the reverse water-filling algorithm (Cover & Thomas, 2006).

Upper Bound Experiments For the $\mathcal{Z} = \hat{\mathcal{X}}$ (no decoder) experiment, we chose $Q_{\hat{X}}$ and $Q_{\hat{X}|X}$ to be factorized Gaussians; we let the mean and variance vectors of $Q_{\hat{X}}$ be trainable parameters, and predict the mean and variance of $Q_{\hat{X}|X=x}$ by one fully connected layer with $2n$ output units, using softplus activation for the n variance components.

For our experiments involving a decoder, we parameterize the variational distributions Q_Z and $Q_{Z|X}$ similarly to before, and use an MLP decoder with one hidden layer (with the number of units equal the data dimension n, and leaky ReLU activation) to map from the $\mathcal{Z}$ to $\hat{\mathcal{X}}$ space. We observe the best performance with a linear (or identity) decoder and a simple linear encoder; using deeper networks with nonlinear activation required more training iterations for SGD to converge, and often to a poorer upper bound. In fact, for a factorized Gaussian source, it can be shown analytically that the optional $Q^*_{Y|X=x}$ is a Gaussian whose mean depends linearly on x, and an identity (no) decoder is optimal.

Lower Bound Experiments In our lower bound experiments, we parameterize $\log u$ by an MLP with 2 hidden layers with $20n$ hidden units each and SeLU activation, where n is the dimension of

[3] https://github.com/tensorflow/compression

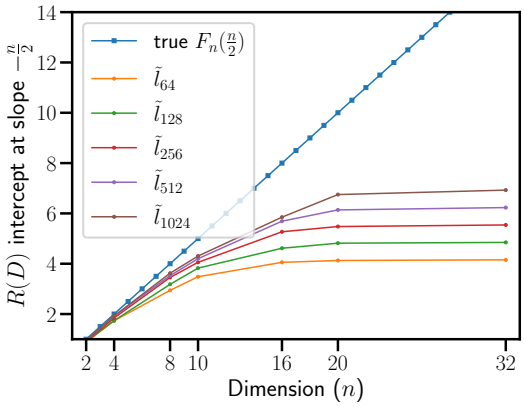

Figure 4: R-axis intercept estimates at (negative) slope $\lambda = \frac{n}{2}$ from our lower bound algorithm, trained with increasing n and k.

the Gaussian source. We fixed $k = 1024$ for the results with varying n in Figure 2-Middle. We vary both k and n in the additional experiment below.

To investigate the necessary k needed to achieve a tight lower bound, and its relation to λ or the Gaussian source dimension n, we ran a series of experiments with k ranging from 64 to 1024 on increasingly high dimensional standard Gaussian sources, each time setting $\lambda = \frac{n}{2}$. For an n-dimensional standard Gaussian, the true R-intercept, $F_n(\lambda)$, has an analytical form; in particular, $F_n(\frac{n}{2}) = \frac{n}{2}$. In Figure 4, we plot the final objective estimate, $\tilde{\ell}_k$, using the converged MLP model, one for each k and n. As we can see, the maximum achievable $\tilde{\ell}_k$ plateaus to the value $\log k$ as we increase n, and for sufficiently high dimension (e.g., $n = 20$ here), doubling k only brings out a constant ($\log 2$) improvement in the objective. This phenomenon relates to the over-estimation bias of C_k when k is too low compared to λ or the "spread" of the data, and can be understood as follows.

For a given sample size k, there is a "bad" regime produced by increasingly large λ (corresponding to an exceedingly narrow Gaussian kernel), or increasingly high (intrinsic) dimension of the data, so that the k data samples appear very far apart. Numerically, this is exhibited by very quick termination of the k hill-climbing runs when computing C_k (subroutine compute_C_k in Algorithm. 1), since the mixture components are well separated, and the mixture centroids are nearly stationary points of the mixture density. The value of the mixture density at each centroid x_i can be approximated as $\gamma_k(x_i) = \frac{1}{k}\frac{1}{u(x_i)} + \frac{1}{k}\sum_{j\neq i}\exp\{-\lambda\rho(x_j, x_i)\}/u_\theta(x_j) \approx \frac{1}{ku(x_i)}$, since $\exp\{-\lambda\rho(x_j, x_i)\} \approx 0$ for all $j \neq i$. The maximization problem defining C_k then essentially reduces to checking which mixture component is the highest, and returning the corresponding centroid (or a point very close to it), so that $C_k \approx \sup_{i=1,\dots,k} \frac{1}{ku(x_i)}$. This implies

$$\mathbb{E}[C_k] \approx \mathbb{E}[\sup_{i=1,\dots,k} \frac{1}{ku(X_i)}] \leq \mathbb{E}[\sup_x \frac{1}{ku(x)}] = \sup_x \frac{1}{ku(x)},$$

therefore

$$\begin{aligned}
\ell_k &:= \mathbb{E}[-\log u(X)] - \log \mathbb{E}[C_k] && (16)\\
&\leq \mathbb{E}[-\log u(X)] + \log k + \inf_x \log u(x) && (17)\\
&= \log k + \mathbb{E}[-\log u(X) + \inf_{x'} \log u(x')] \leq \log k. && (18)
\end{aligned}$$

So, when these approximations hold, the maximum achievable lower bound objective $\tilde{\ell}_k$ can never exceed $\log k$. On sources such as high-dimension (e.g., $n = 10000$) Gaussians, or 256×256 patches of natural images, λ needs to be on the order of 10^6 to target low-distortion region of the R-D curve, and the above analysis well describes the behavior of the lower bound algorithm.

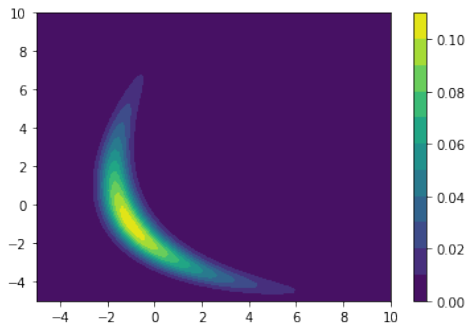

Figure 5: Density plot of the 2D banana-shaped distribution from Ballé et al. (2021).

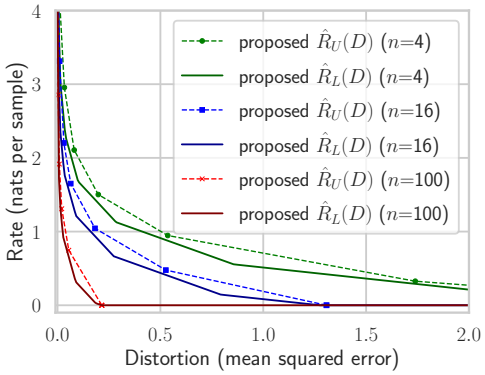

Figure 6: R-D sandwich bounds on higher-dimension random projections of the banana-shaped source.

A.5.3 Banana-shaped Source

Data The 2D banana-shaped source (Ballé et al., 2021) is a RealNVP transform of a 2D Gaussian [4]. See Figure 5 for a plot of its density function. We also consider mapping it to a higher dimension n by a $n \times 2$ matrix. We implemented this by simply passing samples of the banana source through a linear/dense MLP layer with n output units and no activation, with its weight matrix randomly drawn from a Gaussian by the Glorot initialization procedure.

Upper Bound Experiments For the 2D banana-shaped source, we base our β-VAE model architecture on the compressive autoencoder in Ballé et al. (2021), using a 2-dimensional latent space and two-layer MLPs for both the encoder and decoder, with 100 hidden units and softplus activation in each hidden layer. We parameterize Q_Z by a MAF (Papamakarios et al., 2017), and $Q_{Z|X}$ by an IAF (Kingma et al., 2016) (we found that a factorized Gaussian $Q_{Z|X}$ works just as well, as long as the MAF prior Q_Z is sufficiently powerful).

For the higher-dimension projections of the banana-shaped source, we use the same β-VAE as on the 2D source, but set the number of hidden units in each hidden layer as $100n$, where n is the dimension of the source. We cap the number of hidden units per layer at 2000 for $n > 20$, and do not find this to adversely affect the results.

Lower Bound Experiments We use an MLP with three hidden layers and SeLU activations for the $\log u$ model; as in the upper bound models, here we set the number of hidden units in each layer to be $100n$, and cap it at 1000. In all the experiments we set $k = 2048$.

As illustrated by Figure 6, the tightness of our sandwich bounds do not appear to be affected by the dimension n of the data (we verified this up to $n = 1000$), unlike on the Gaussian experiment where for a fixed k the lower bound became increasingly loose with higher n.

A.5.4 GAN-generated Images

Data We adopt the same setup as in the GAN experiment by Pope et al. (2021). We use a BigGAN (Brock et al., 2019) pretrained on ImageNet at 128×128 resolution (so that $n = 128 \times 128 \times 3 = 49152$), downloaded from `https://tfhub.dev/deepmind/biggan-deep-128/1`. To generate an image of an ImageNet category, we provide the corresponding one-hot class vector as well as a noise vector to the GAN. Following Poole et al. (2019), we control the intrinsic dimension d of the generated images by setting all except the first d dimensions of the 128-dimension noise vector to zero, and use a truncation level of 0.8 for the sample diversity factor. Since the GAN generates values in $[-1, 1]$, we rescale them linearly to $[0, 1]$ to correspond to images, so that the alphabets

[4] Source code taken from the authors' GitHub repo `https://github.com/tensorflow/compression/blob/66228f0faf9f500ffba9a99d5f3ad97689595ef8/models/toy_sources/toy_sources.ipynb`.

Figure 7: Random samples of `basenji` from a BigGAN, $d = 2$.

Figure 8: Random samples of `basenji` from a BigGAN, $d = 4$.

$\mathcal{X} = \hat{\mathcal{X}} = [0, 1]^n$. As in (Pope et al., 2021), we experimented with images from the ImageNet class `basenji`. In Figure 7 and 8, we plot additional random samples, for $d = 2$ and $d = 4$.

Upper Bound Experiments We implemented a version of ResNet-VAE following the appendix of Kingma et al. (2016), using the 3×3 convolutions of Cheng et al. (2020) in the encoder and decoder of our model. Our model consists of 6 layers of latent variables, and implements bidirectional inference using factorized Gaussian variational distributions at each stochastic layer. During an inference pass, an input image goes through 6 stages of convolution followed by downsampling (each time reducing the height and width by 2), and results in a stochastic tensor at each stage. In encoding order, the stochastic tensors have decreasing spatial extent and an increasing number of channels equal to 4, 8, 16, 32, 64, and 128. The latent tensor Z_0 at the topmost generative hierarchy is flattened and modeled by a MAF prior Q_{Z_0} (aided by the fact that all the images have a fixed shape).

Lower Bound Experiments We parameterize the $\log u$ model by a simple feedforward convolutional neural network. It contains three convolutional layers with 4, 8, and 16 filters, each time downsampling by 2, followed by a fully connected layer with 1024 hidden units. Again we use SeLU activation in the hidden units, and set $k = 2048$ in all the experiments. When evaluating the global optimization objective γ_k (see Algorithm 1), we loop through a small batch of 32 x_i sample at a time to avoid running out of GPU memory.

A.5.5 Natural Images

Data For signals such as images or video, which can have arbitrarily high spatial dimension, it is not immediately clear how to define i.i.d. samples. But since our focus is on image compression, we follow the common practice of training convolutional autoencoders on random 256×256-pixel crops of high resolution images as in learned image compression research (Ballé et al., 2017), noting that current methods cannot effectively exploit spatial redundancies larger than this scale (Minnen & Singh, 2020). As a representative dataset of natural images, we used images from the COCO 2017 (Lin et al., 2014) training set that are larger than 512×512, and downsampled each by a random factor in $[0.6, 1]$ to remove potential compression artifacts from JPEG. We trained image compression models from (Minnen et al., 2018; Minnen & Singh, 2020) on our dataset and were able to closely reproduce their performance.

Upper Bound Experiments Instead of working with variational distributions over the set of pixel values $\hat{\mathcal{X}} = \mathcal{X} = \{0, 1, ..., 255\}^n$, which would require specialized tools for representing and optimizing high-dimensional discrete distributions (Salimans et al., 2017; Hoogeboom et al., 2019; Maddison et al., 2016; Jang et al., 2016), here we parameterize the variational distributions in Euclidean latent spaces for simplicity (see Sec. 3).

We applied our upper bound algorithm based on two VAE architectures:

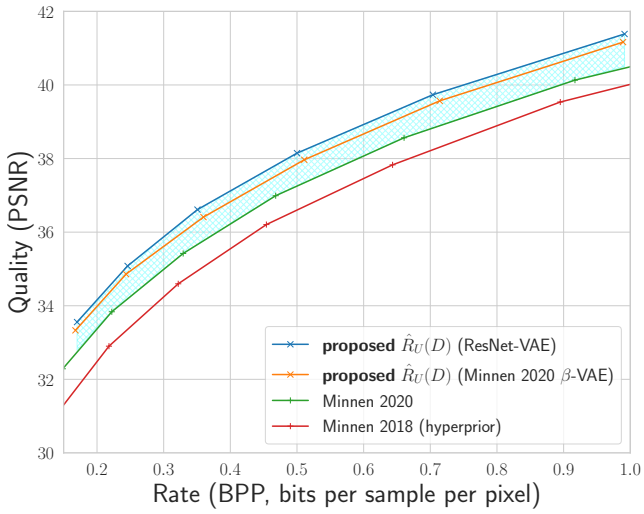

Figure 9: Quality-rate curves on the Tecnick dataset.

1. The learned image compression model from Minnen & Singh (2020) [5]. This corresponds to a hierarchical VAE with two layers of latent variables, with the lower level latents modeled by a channel-wise autoregressive prior conditioned on the top-level latents. We adapt it for our purpose, by using factorized Gaussians for the variational posterior distributions $Q_{Z|X}$ (instead of factorized uniform distributions used to simulate rounding in the original model), and no longer convolving the variational prior Q_Z with a uniform distribution as in the original model.

2. The ResNet-VAE from our experiment on GAN images. Following learned image compression models, we maintain the convolutional structure of the topmost latent tensor (rather than flattening it in the GAN experiments) in order to make the model applicable to images of variable sizes. Instead of a MAF, we model the topmost latent tensor by a deep factorized (hyper-)prior Q_{Z_0} as proposed by Ballé et al. (2018).

For the β-VAE based on Minnen & Singh (2020), $\dim(\mathcal{Z}) \approx 0.28 \dim(\mathcal{X})$; and for our β-ResNet-VAE, $\dim(\mathcal{Z}) \approx 0.66 \dim(\mathcal{X})$. We did not observe improved results by simply increasing the number of latent channels/dimensions in the former model. We follow the same model training practice as reported by Minnen & Singh (2020), training both models to around 5 million gradient steps.

In our problem setup, we consider the reproduction alphabet to be the discrete set of pixel values $\mathcal{X} = \{0, 1, ..., 255\}^n$, so the decoder network ω needs to discretize the output of the final convolutional layer (in `Tensorflow`, this can be implemented by a `saturate_cast` to `uint8`). However, since the discretization operation has no useful gradient, we skip it during training, and only apply it for evaluation.

The operational quality-rate curves of various compression methods are taken from the `tensorflow-compression`[6] and `compressAI`[7] libraries. The results of Minnen & Singh (2020) represent the current state-of-the-art of neural image compression, to the best of our knowledge.

Lower Bound Experiments When running our lower bound algorithm on 256×256 crops of natural images, we observed similar behavior as on high-dimension Gaussians, with the loss $\tilde{\ell}_k$

[5] https://github.com/tensorflow/compression/blob/f9edc949fa58381ffafa5aa8cb37dc8c3ce50e7f/models/ms2020.py

[6] https://github.com/tensorflow/compression/tree/8692f3c1938b18f123dbd6e302503a23ce75330c/results/image_compression

[7] https://github.com/InterDigitalInc/CompressAI/tree/baca8e9ff070c9f712bf4206b8f2da942a0e3dfe/results

quickly plateauing to a value around $\log k$. This issue persisted when we tried different u model architectures. See a discussion on this phenomenon in Section A.5.2.

A.6 Measures of Variability

As alluded to in Sec. 3, the proposed R-D bounds are estimated as empirical averages from samples, which only equal the true quantities in expectation. Before we characterize the variability of the bound estimates, we first define the exact form of the estimators used.

Upper Bound Estimator For the proposed upper bound, consider the variational problem solved by the Lagrangian Eq. 3, and define the random variables from the rate and distortion terms of the Lagrangian:

$$\mathcal{R}(Z, X) := \log q(Z|X) - \log q(Z),$$

and

$$\mathcal{D}(Z, X) := \rho(X, \omega(Z)).$$

Above, X and Z follow the joint distribution $P_X Q_{Z|X}$, ω is the decoder function, and $q(z|x)$ and $q(z)$ are the Lebesgue density functions of absolutely continuous $Q_{Z|X=x}$ and Q_Z variational distributions (we only worked with absolutely continuous variational distributions for simplicity). Then it is clear that the expectations of the two random variables equal the rate and distortion terms, i.e.,

$$\mathbb{E}[\mathcal{R}(Z, X)] = \mathbb{E}_{x\sim P_X}[KL(Q_{Z|X=x} \| Q_Z)],$$

and

$$\mathbb{E}[\mathcal{D}(Z, X)] = \mathbb{E}_{P_X Q_{Z|X}}[\rho(X, \omega(Z))].$$

Recall for any given decoder and variational distributions, the above rate and distortion terms define a point $(\mathbb{E}[\mathcal{D}], \mathbb{E}[\mathcal{R}])$ that lies on an upper bound $R_U(D)$ of the source $R(D)$ curve. Given m i.i.d. pairs of $(Z_1, X_1), (Z_2, X_2), ..., (Z_m, X_m) \sim P_X Q_{Z|X}$, we estimate such a point by $(\mu_{\mathcal{D}}, \mu_{\mathcal{R}})$, using the usual sample mean estimators $\mu_{\mathcal{R}} := \frac{1}{m}\sum_{j=1}^m \mathcal{R}(Z_j, X_j)$ and $\mu_{\mathcal{D}} := \frac{1}{m}\sum_{j=1}^m \mathcal{D}(Z_j, X_j)$.

Lower Bound Estimator For the proposed lower bound, given i.i.d. $X_1, ..., X_k \sim P_X$, define the random variable

$$\xi := -\frac{1}{k}\sum_i^k \log u(X_i) - \frac{C_k}{\alpha} - \log \alpha + 1,$$

where the various quantities are defined as in Sec. 4. Then it is clear that its expectation equals the lower bound objective Eq. 9, i.e., $\mathbb{E}[\xi] = \tilde{\ell}_k := \mathbb{E}[-\log u(X)] - \mathbb{E}[C_k]/\alpha - \log \alpha + 1$. Given m i.i.d. samples of ξ, we form the usual sample mean estimator $\mu_\xi := \frac{1}{m}\sum_j \xi_j$, and form an R-D lower bound estimate by

$$\hat{R}_L(D) := -\lambda D + \mu_\xi.$$

In computing μ_ξ, we fix the constant α to a sample mean estimate of $\mathbb{E}[C_k]$ computed separately (as in line 7 of Algorithm 1). As explained in Sec. A.4, we repeat this procedure with different λ, and our overall R-D lower bound is obtained by taking the point-wise maximum of such linear under-estimators over λ.

Results Below we give measures of variability of for the sandwich bound estimates obtained on GAN generated images (Section 6.2). We report detailed statistics in the following tables, and plot them in Figure 10. Our results on other experiments exhibit similar or comparatively less variability.

For the upper bound, given each trained β-VAE, we compute m samples of $(\mathcal{R}, \mathcal{D})$, essentially passing m samples of X through the autoencoder and collect the rate and distortion values; then we report the sample mean (μ), sample standard deviation (s), and 95% large-sample confidence interval for the population mean (the true expected value) for $\mathcal{R}$ and $\mathcal{D}$, respectively, in Tables 1 and 2.

For the lower bound, given each trained $\log u$ model, we compute m samples of ξ, then report the sample mean (μ_ξ), sample standard deviation (s_ξ), and 90% large-sample confidence lower bound for the population mean (the true expected value $\mathbb{E}[\xi] = \tilde{\ell}_k$) in Tables 3 and 4.

λ	$\mu_{\mathcal{R}}$	$s_{\mathcal{R}}$	$\mathbb{E}[\mathcal{R}]$ CI	$\mu_{\mathcal{D}}$	$s_{\mathcal{D}}$	$\mathbb{E}[\mathcal{D}]$ CI
300	0.278	0.8	(0.15, 0.41)	0.006	0.00287	(0.00553, 0.00647)
500	1.25	1.27	(1.04, 1.46)	0.00351	0.00207	(0.00317, 0.00385)
1e+03	2.69	1.21	(2.49, 2.89)	0.00157	0.00103	(0.0014, 0.00174)
4e+03	4.77	1.25	(4.56, 4.97)	0.000352	0.000285	(0.000305, 0.000399)
8e+03	5.65	1.31	(5.44, 5.87)	0.000196	0.000152	(0.000171, 0.000221)
1.6e+04	6.43	1.32	(6.21, 6.64)	0.000126	9.68e-05	(0.00011, 0.000142)

Table 1: Statistics for the estimated R-D upper bound on GAN-generated images with $d = 2$, computed with $m = 100$ samples of $(\mathcal{R}, \mathcal{D})$. "CI" denotes 95% large-sample confidence interval.

λ	$\mu_{\mathcal{R}}$	$s_{\mathcal{R}}$	$\mathbb{E}[\mathcal{R}]$ CI	$\mu_{\mathcal{D}}$	$s_{\mathcal{D}}$	$\mathbb{E}[\mathcal{D}]$ CI
300	0.265	0.689	(0.15, 0.38)	0.00978	0.00376	(0.00916, 0.0104)
500	1.44	1.24	(1.23, 1.64)	0.00693	0.00284	(0.00646, 0.00739)
1e+03	4.08	1.73	(3.80, 4.36)	0.00332	0.00152	(0.00307, 0.00357)
2e+03	6.59	1.88	(6.28, 6.90)	0.00151	0.000752	(0.00139, 0.00164)
4e+03	8.66	1.89	(8.35, 8.97)	0.000774	0.000389	(0.00071, 0.000838)
8e+03	10.4	1.89	(10.06, 10.68)	0.000476	0.000229	(0.000438, 0.000514)
1.6e+04	12	1.83	(11.69, 12.29)	0.000323	0.000147	(0.000299, 0.000347)

Table 2: Statistics for the estimated R-D upper bound on GAN-generated images with $d = 4$, computed with $m = 100$ samples of $(\mathcal{R}, \mathcal{D})$. "CI" denotes 95% large-sample confidence interval.

λ	μ_ξ	s_ξ	$\mathbb{E}[\xi]$ LCB
49.80	0.37	0.01	0.36
99.30	0.74	0.02	0.73
199.82	1.46	0.03	1.45
299.58	2.01	0.03	2.00
499.87	2.76	0.05	2.74
752.70	3.29	0.05	3.27
799.92	3.41	0.05	3.39
999.44	3.63	0.07	3.60
1514.20	4.04	0.06	4.01
1999.30	4.21	0.08	4.18
2275.80	4.33	0.07	4.30
3000.00	4.50	0.07	4.47

Table 3: Statistics for the estimated R-D lower bound on $d = 2$ GAN-generated images, computed with $m = 15$ samples of ξ. "LCB" denotes 90% large-sample lower confidence bound.

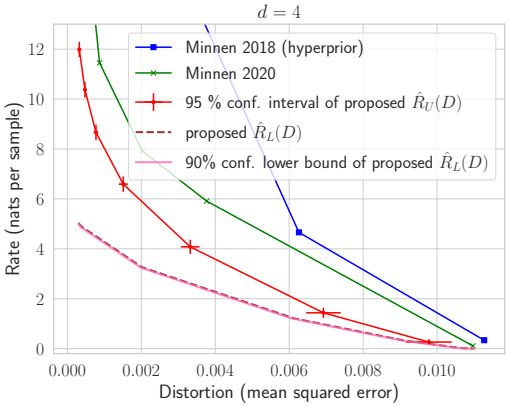

Figure 10: Zoomed-in version of Figure 3-Middle-Bottom (GAN-generated images with intrinsic dimension $d = 4$), showing the sample mean estimates of R-D upper bound points $(\mu_\mathcal{D}, \mu_\mathcal{R})$ bracketed by 95% confidence intervals (**red**), as well as lower bound estimates based on the sample mean μ_ξ (**maroon**, dashed) and its 90% confidence lower bound (**pink** line, beneath **maroon**).

λ	μ_ξ	s_ξ	$\mathbb{E}[\xi]$ LCB
99.63	1.09	0.02	1.08
199.90	2.15	0.04	2.13
299.15	3.05	0.05	3.03
499.69	4.27	0.08	4.24
999.01	5.28	0.11	5.23
1011.20	5.23	0.12	5.18
1999.70	5.64	0.08	5.60
2024.00	5.63	0.06	5.60
2999.30	5.76	0.09	5.72
3036.80	5.74	0.10	5.69
4000.00	5.81	0.08	5.77

Table 4: Statistics for the estimated R-D lower bound on $d = 4$ GAN-generated images, computed with $m = 15$ samples of ξ. "LCB" denotes 90% large-sample lower confidence bound.