LOWER BOUNDING RATE-DISTORTION FROM SAMPLES

Yibo Yang and Stephan Mandt
Department of Computer Science, UC Irvine
{yibo.yang,mandt}@uci.edu

ABSTRACT

The rate-distortion function $R(D)$ tells us the minimal number of bits on average to compress a random object within a given distortion tolerance. A lower bound on $R(D)$ therefore represents a fundamental limit on the best possible rate-distortion performance of any lossy compression algorithm, and can help us assess the potential room for improvement. We make a first attempt at an algorithm for estimating such a lower bound from data samples, applicable to general memoryless data sources. Based on a dual characterization of $R(D)$ (Csiszár, 1974), our method solves a constrained maximization problem over a family of functions parameterized by neural networks. On a 2D Gaussian source, we obtain a lower bound within 1 bit of the analytical $R(D)$. Our code can be found here.

1 INTRODUCTION AND BACKGROUND

Let $X \in \mathcal{X}$ be a random variable with distribution P_X [1] that represents a memoryless data source, let $\mathcal{Y}$ be the set of lossy representations, and let $\rho : \mathcal{X} \times \mathcal{Y} \to [0, \infty)$ be a distortion cost function. The (information) rate-distortion function $R(D)$ is defined as the solution to the optimization problem,

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

where we consider all stochastic transforms $P_{Y|X}$ whose expected distortion is within the given threshold D, and minimize the mutual information between the source X and its reproduced representation Y. Rate-distortion theory (Shannon, 1959; Cover, 1999) gives operational meaning to the above definition as the asymptotic fundamental limit of lossy data compression. In essence, $R(D)$ delimits the minimal average number of bits (or nats, depending the base of $\log$) needed to convey i.i.d. samples of X within average distortion D, for any block code that is allowed arbitrarily long blocks or high complexity, regardless of its implementation technology. Therefore, the rate-distortion function, or a lower bound on it, can help analyze how close lossy compression algorithms are to their theoretical performance limits. If the operational rate and distortion of a compression algorithm lies far above the $R(D)$ curve (or its lower bound), then we may expect room for further improvement; otherwise, the algorithm's rate-distortion performance is already close to theoretically optimal, so we may better focus our attention on improving other aspects of the algorithm.

Although $R(D)$ has no analytical form in general, it can be computed in principle by solving the convex optimization problem Eq. 1. The mutual information $I(X;Y)$, however, is typically challenging to compute, so a tractable upper bound $I(X;Y) \leq I(X;Y)+KL(P_Y\|\tilde{P}_Y) = \mathbb{E}_X[KL(P_{Y|X}\|\tilde{P}_Y)]$ is often used instead in the optimization problem. Solving the resulting minimization problem then leads to upper bounds on $R(D)$, and is the basis of the Blahut–Arimoto algorithm (Arimoto, 1972). The same mutual information upper bound has also made numerous appearances in machine learning (Alemi et al., 2017; Poole et al., 2019), notably as the aggregate KL-divergence regularizer in Variational-Autoencoders (VAEs) (Kingma & Welling, 2013). Traditionally, the set of lossy presentations $\mathcal{Y}$ is often a subset of the data space $\mathcal{X}$, in the form of quantization points, and distortion ρ is fixed. In VAEs and their recent application to learned data compression (Ballé et al., 2017), $\mathcal{Y}$ has the interpretation of a *latent space*, $\tilde{P}_Y$ is the *prior*, and $\rho(x, y) = \rho_0(x, f_\theta(y))$ corresponds

[1] Formally, $X : \Omega \to \mathcal{X}$ is a measurable function on an underlying probability space $(\Omega, \mathcal{A}, \mathbb{P})$, and P_X is the image measure of $\mathbb{P}$ under X.

to the negative log likelihood $\log p_\theta(x|y)$ parameterized by a *decoder* $f_\theta : \mathcal{Y} \to \mathcal{X}$ (e.g., $p_\theta(x|y)$ is a Gaussian density with mean $f_\theta(y)$, ρ_0 is a squared error), so the distortion ρ is learned. The Lagrangian $\mathbb{E}_X[KL(P_{Y|X}\|\tilde{P}_Y)] + \lambda\mathbb{E}[\rho(X,Y)]$ then corresponds to the negative ELBO (Kingma & Welling, 2013) for learning and inference. Using the analytical density of $P_{Y|X}$ that optimizes the Lagrangian (assumed to exist), Huang et al. (2020) sample from this optimal $P_{Y|X}$ to compute optimistic upper bounds on $R(D)$ of trained VAEs and other generative models, hinting at their *possible* compression performance. Our method is agnostic to the choice of representations $\mathcal{Y}$ or distortion ρ, and can also be extended to compute $R(D)$ lower bounds of trained decoder-based models. Unlike (Huang et al., 2020), however, we do not assume (or require) a prior distribution $\tilde{P}_Y$, and our lower bound establishes what kind of lossy compression performance is theoretically *impossible* to obtain.

Lower bounds of $R(D)$, by contrast, have received considerably less attention, potentially because of difficulty in obtaining them. A variational lower bound of $R(D)$ already appeared in Shannon's landmark paper (Shannon, 1948), which was later extended (Berger, 1971) and proved rigorously for X in a general abstract probability space by Csiszár (1974); we adopt the version in (Kostina, 2016). The special case of a discrete and known source P_X is considered by Chiang & Boyd (2004), who solve the corresponding (finite-dimensional) dual problem of $R(D)$ by geometric programming.

2 Method

2.1 Dual characterization of $R(D)$

We use the below general characterization of $R(D)$ in terms of a constrained maximization problem:

Theorem 2.1. *(Kostina, 2016) Under basic regularity conditions (e.g., satisfied by a bounded ρ; see Appendix Sec. A.1), for any distortion tolerance $D > D_{min} := \inf\{D : R(D) < \infty\}$, it holds that*

$$R(D) = \max_{g(x),\lambda} \{\mathbb{E}[-\log g(X)] - \lambda D\} \tag{2}$$

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

$$\mathbb{E}\left[\frac{\exp(-\lambda\rho(X,y))}{g(X)}\right] = \int \frac{\exp(-\lambda\rho(x,y))}{g(x)} dP_X(x) \leq 1, \forall y \in \mathcal{Y} \tag{3}$$

In other words, every pair of feasible (λ, g) satisfying Eq. 3 yields a lower bound of $R(D)$, i.e., $\mathbb{E}[-\log g(X)] - \lambda D \leq R(D)$, and we obtain $R(D)$ via the tightest such lower bound. This characterization of $R(D)$ can be derived from Lagrange duality (Chiang & Boyd, 2004) in the case of a finite alphabet $\mathcal{X}$; the extension to a general $\mathcal{X}$ requires a more involved argument (Csiszár, 1974).

2.2 Lower Bounding $R(D)$ via numerical optimization

We rarely know P_X explicitly; e.g., when P_X is induced by natural images, characterizing the support of P_X (a low dimensional manifold of Euclidean space) alone is non-trivial. However, if we can compute the expectations in Eqs. 2 and 3, then we can still obtain a lower bound of $R(D)$ by numerically solving a constrained optimization problem. The main idea is to replace maximizing over functions $g \geq 0$ by a subset of functions $g_\theta \geq 0$ parameterized by a vector θ. Any feasible (g_θ, λ) would then give us a (possibly loose) lower bound on $R(D)$. We can use a flexible family of functions for g_θ, such as neural networks with a non-negative output activation to ensure $g_\theta \geq 0$ (or, we can parameterize $\log g$ instead). Noting that the constraint Eq. 3 is equivalent to the constraint function $c(g,\lambda) := \sup_y \mathbb{E}\left[\exp(-\lambda\rho(X,y))/g(X)\right] - 1$ being non-positive, we can then consider solving the following constrained maximization problem to obtain a $R(D)$ lower bound, $R_L(D)$:

$$R(D) \geq R_L(D) = \max_{\theta,\lambda\geq 0} f(\theta,\lambda), \quad \text{where } f(\theta,\lambda) := \mathbb{E}[-\log g_\theta(X)] - \lambda D \tag{4}$$

$$\text{subject to} \quad c(\theta,\lambda) = \sup_y \mathbb{E}\left[\frac{\exp(-\lambda\rho(X,y))}{g_\theta(X)}\right] - 1 \leq 0 \tag{5}$$

Assuming the supremum in Eq. 5 can be computed exactly, then a variety of algorithms can be used to find a local optimum of this problem. We consider a simple formulation based on exact penalty

method (Nocedal & Wright, 2006): we solve a sequence of unconstrained problems

$$\max_{\theta,\lambda\geq 0} f(\theta,\lambda) - \gamma_t[c(\theta,\lambda)]^+ \tag{6}$$

where $[c(\theta,\lambda)]^+ = \max\{0, c(\theta,\lambda)\}$ measures the amount of constraint violation, and the sequence of penalty coefficients γ_t is gradually increased. Under fairly general conditions, a feasible solution $\{(\theta_t^*, \lambda_t^*)\}$ to Eq. 6 is also a local optimum of the original constrained optimization problem in Eqs. 4, 5 (Theorem 17.4 of Nocedal & Wright (2006)). Then we can estimate $R_L = f(\theta_t^*, \lambda_t^*)$.

2.3 An Approximate Algorithm using stochastic estimators

In most practical problems, the expectations in the optimization problem can only be approximated by Monte-Carlo averaging over samples of X. Let's focus on the constraint function c for now, and denote $\psi(x,y) := \frac{\exp(-\lambda\rho(x,y))}{g(x)} - 1$, suppressing its dependence on (g,λ) for brevity. Note, however, that $c = \sup_y \mathbb{E}[\psi(X,y)]$ does not have the form of an expectation due to the maximization over y. Still, we may try to estimate c by maximizing the sampled approximation of $\mathbb{E}[\psi(X,y)]$, i.e., compute the k-sample estimator $C_k := \sup_y \frac{1}{k}\sum_{i=1}^k \psi(X_i,y)$, with $X_i \sim P_X$. What's the relation between C_k and c? Given a candidate (g,λ), can we verify its feasibility using C_k, instead of the intractable c? We answer these questions in the following theorem (proved in Appendix Sec. A.2):

Theorem 2.2. *Let $X_1, X_2, ...$ be a sequence of i.i.d. random variables with the shared data distribution $X_k \sim P_X$. Define the sequence of random variables $C_k := \sup_y \frac{1}{k}\sum_i \psi(X_i,y)$. Then*

1. $\mathbb{E}[C_k] = \mathbb{E}_{X_1,...,X_k}[\sup_y \frac{1}{k}\sum_i \psi(X_i,y)] \geq \sup_y \mathbb{E}[\psi(X,y)] =: c$;
2. $\mathbb{E}[C_1] \geq \mathbb{E}[C_2] \geq ... \geq \mathbb{E}[C_k] \geq \mathbb{E}[C_{k+1}] \geq ... \sup_y \mathbb{E}[\psi(X,y)] = c$;
3. *If $\psi(x,y)$ is bounded and continuous in y, and if $\mathcal{Y}$ is compact, then 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 $\lim_{k\to\infty}\mathbb{E}[C_k] = c$.*

Theorem 2.2 tells us that C_k is on average an over-estimator of the value of the constraint 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. This means that we can form an empirical estimate of the constraint function c by computing $\hat{C}_k := \sup_y \frac{1}{k}\sum_i \psi(x_i,y)$ from a large batch of samples $\{x_1, x_2, ..., x_k\}$ (or, to be more precise, we average many such $\hat{C}_k$ computed from multiple batches to estimate $\mathbb{E}[C_k]$); if the empirical estimate of c is satisfied, then we can be confident so is the true constraint c satisfied.

In light of this, we can replace the original constraint $c \leq 0$ by the more conservative constraint $\mathbb{E}[C_k] \leq 0$. Using the unbiased estimator $\hat{C}_k$ for $\mathbb{E}[C_k]$, and similarly the sample-mean $\hat{f} := \frac{1}{m}\sum_{j=1}^m -\log g(x_j) - \lambda D$ for f, our algorithm solves the following sequence of penalty problems,

$$\max_{\theta,\lambda\geq 0} \hat{\ell}(\theta,\lambda), \quad \text{with } \hat{\ell}(\theta,\lambda) := \hat{f}(\theta,\lambda) - \gamma_t[\hat{C}_k(\theta,\lambda)]^+ \tag{7}$$

where given a penalty coefficient γ_t, we perform stochastic gradient ascent on the unconstrained objective $\hat{\ell}(\theta,\lambda)$ till convergence; we then increase the penalty e.g., $\gamma_{t+1} := \beta\gamma_t$ with $\beta > 1$, and solve the maximization problem w.r.t. (θ,λ) again, until we can verify feasibility of resulting (θ,λ) (e.g., $\hat{C}_k \leq 0$) with high confidence. We specify Algorithm 1 in full detail in Appendix Sec. A.3.

2.4 Inner optimization with respect to y

So far we have assumed that we can exactly compute the supremum $\hat{C}_k = \sup_y \frac{1}{k}\sum_i \psi(x_i,y)$. In practice, $\frac{1}{k}\sum_i \psi(x_i,y)$ is rarely concave in y, and we can only find a local optimum, e.g., with a gradient-based method. This is less problematic during training, as long as (g,λ) receive the appropriate training signal to eventually arrive at a feasible solution. However, once the training terminates with some candidate solution (g^*,λ^*), we need to compute $\hat{C}_k$ exactly (ideally with as large k as possible), in order to verify that the solution (g^*,λ^*) is feasible (with

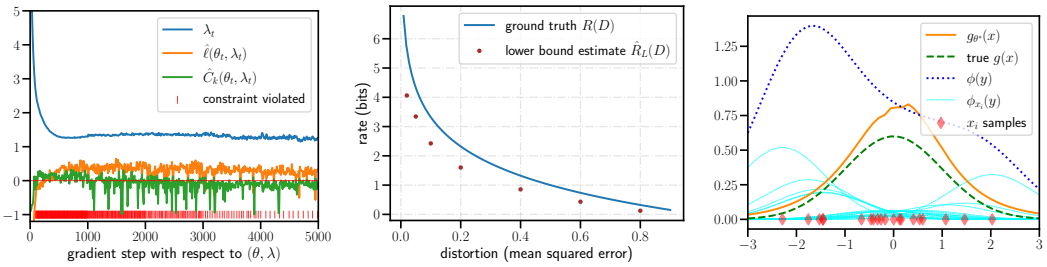

Figure 1: **Left**: trajectories of various quantities across training steps, for the $D = 0.6$ experiment; **Middle**: the final lower bound estimate $\hat{R}_L(D)$, plotted against the true $R(D)$; **Right**: visualizing (along the y-axis of $\mathbb{R}^2$) a converged neural network solution $g_{\theta^*}(x)$, the theoretically optimal $g(x)$, and the optimization objective of the constraint estimator $\hat{C}_k$, $\phi(y) := \sum_{i=1}^{k} \phi_{x_i}(y)$, with $\phi_{x_i}(y) := \exp\{-\lambda^* \|x_i - y\|^2\}/\{k g_{\theta^*}(x_i)\}$, computed from $k = 20$ samples, in the $D = 0.6$ experiment.

high probability) and that $\hat{f}(g^*, \lambda^*)$ gives a valid estimate of a $R(D)$ lower bound. This requirement makes it computationally expensive (if not infeasible) to verify a solution, and indeed poses an open challenge for our method; techniques from global optimization might offer useful tools. We note that in the simple case $\mathcal{X} = \mathcal{Y} = \mathbb{R}^n$ and ρ is the (mean) squared error, maximizing $\phi(y) := \frac{1}{k}\sum_i \psi(x_i, y) + 1 = \alpha \sum_i \alpha_i \exp(-\lambda \|x_i - y\|^2)$ is equivalent to finding the mode of a k-component Gaussian mixture density. Global optimization algorithms for the latter problem exist with (partial) optimality guarantees (Carreira-Perpinan, 2000); e.g., we can run k gradient ascent procedures separately from each of the component modes $x_1, ..., x_k$, and take the largest result.

3 Results

We apply our algorithm to a 2D standard Gaussian source, using mean-squared error as distortion ρ. We parameterize $\log g$ by a two-layer fully connected neural network. During training, we use $k = 100$ for the constraint estimator $\hat{C}_k$, and run a simple gradient ascent procedure in the inner optimization w.r.t. y for computational efficiency. We start with a penalty parameter $\gamma_0 = 1$, and double it every 1000 training steps (which seemed sufficient for convergence). We stop training after 4000 steps, at which point our simple estimate of constraint $\hat{C}_k$ is frequently satisfied across training steps; see Fig. 1 (Left) for example training curves. To certify the resulting solution (g^*, λ^*) as valid, we then compute $\hat{C}_k$ with the global optimization algorithm described in Section 2.4, using a large number of samples ($k = 10000$); in all our experiments, $\hat{C}_{10000}(g^*, \lambda^*) < 0$, so we conclude the solutions are feasible, and $\hat{f}(g^*, \lambda^*)$ gives a valid lower bound of $R(D)$ with very high probability.

We repeat the above with D ranging from 0.02 to 0.8, and plot the corresponding $\hat{R}_L(D) = \hat{f}(g^*, \lambda^*)$ to trace out an $R(D)$ lower bound. As shown in Fig. 1 (Middle), our lower bound shows good agreement with the true $R(D)$, with an average gap of 0.74 bit. Fig. 1 (Right) visualizes the optimization problem for $D = 0.6$, where we take a vertical slice of $\mathbb{R}^2$ and show the learned $g_{\theta^*}(x)$, the theoretically derived optimal $g(x)$ (a scaled Gaussian density), and the $\hat{C}_k$ optimization objective $\phi(y)$ and its component functions $\phi_{x_i}(y)$ centered on $k = 20$ random samples, plugging in the learned $(g_{\theta^*}, \lambda^*)$. Note that the component functions have the form of a Gaussian kernel, whose steepness is controlled by λ. When λ is large or k is small, the resulting $\hat{C}_k = \max_y \phi(y) - 1$ more severely overestimates the true constraint c. With a smaller λ, or by increasing k, the landscape of $\phi(y)$ is more smoothed out, so the overestimating effect is reduced. Indeed, in this figure we see $\max_y \phi(y) > 1.25$; by increasing the sample size k from 20 to 10000, $\max_y \phi(y)$ would decrease to around 0.93 instead (so that $\hat{C}_k \approx -0.07 < 0$), as predicted by Theorem 2.2. In the low distortion (small D) regime, λ is driven to large values, so that $\hat{C}_k$ exhibits higher bias, and constraint violations $\max\{0, \hat{C}_k\}$ are often penalized overly harshly (as $\hat{C}_k$ tends to overestimate c); this likely explains why our lower bound has a larger gap to the true $R(D)$ as we decrease D. We expect the bound can be improved by more accurately estimating the gradient of the penalty/constraint function (we elaborate this point in Appendix Sec. A.3), and/or training with a larger k especially in the lower distortion regime. We provide more experimental details and visualizations in Appendix Sec. A.4.

4 ACKNOWLEDGEMENTS

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0021. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA). Yibo Yang acknowledges funding from the Hasso Plattner Foundation. Furthermore, this work was supported by the National Science Foundation under Grants 1928718, 2003237 and 2007719, as well as Intel and Qualcomm.

REFERENCES

Alexander A Alemi, Ben Poole, Ian Fischer, Joshua V Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. *arXiv preprint arXiv:1711.00464*, 2017.

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

Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *International Conference on Learning Representations*, 2017.

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

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.

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

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

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

Thomas S Ferguson. *A course in large sample theory*. Routledge, 2017.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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.

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

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

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.

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

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

A APPENDIX

A.1 FULL VERSION OF THEOREM 2.1

Theorem A.1. (Theorem 2.3, Csiszár (1974); 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 $E \subset \mathcal{Y}$ such that*

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

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

$$R(D) = \max_{g(x), \lambda} \{\mathbb{E}[-\log g(X)] - \lambda D\} \tag{8}$$

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

$$\mathbb{E}\left[\frac{\exp(-\lambda \rho(X, y))}{g(X)}\right] = \int \frac{\exp(-\lambda \rho(x, y))}{g(x)} dP_X(x) \leq 1, \forall y \in \mathcal{Y} \tag{9}$$

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.

A.2 PROOF OF THEOREM 2.2

Theorem A.2. (Basic properties of the proposed estimator C_k.)

Let $X_1, X_2, ... \sim P_X$ be a sequence of i.i.d. random variables. Let $\psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ be a measurable function. For each k, define the random variable $C_k := \sup_y \frac{1}{k} \sum_i \psi(X_i, y)$. Then

1. *C_k is an overestimator of the sup-partition function c, i.e.,*
   $\mathbb{E}[C_k] = \mathbb{E}_{X_1,...,X_k}[\sup_y \frac{1}{k} \sum_i \psi(X_i, y)] \geq \sup_y \mathbb{E}[\psi(X, y)] =: c$;
2. *The bias of C_k decreases with increasing k, i.e.,*
   $\mathbb{E}[C_1] \geq \mathbb{E}[C_2] \geq ... \geq \mathbb{E}[C_k] \geq \mathbb{E}[C_{k+1}] \geq ... \sup_y \mathbb{E}[\psi(X, y)] = c$;
3. *If $\psi(x, y)$ is bounded and continuous in y, and if $\mathcal{Y}$ 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. $\mathbb{E}[C_k] = \mathbb{E}[\sup_y \frac{1}{k} \sum_i \psi(X_i, y)] \geq \sup_y \mathbb{E}[\frac{1}{k} \sum_i \psi(X_i, y)] = \sup_y \mathbb{E}[\psi(X, y)] = c$
2. First, note that $\mathbb{E}[C_1] \geq \mathbb{E}[C_k]$ since

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

We therefore have

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

3. Define the shorthand $M_k(y) := \frac{1}{k}\sum_{i=1}^{k}\psi(X_i,y), \Psi(y) = \mathbb{E}[\psi(X,y)]$. Denote the sup norm on the set of continuous bounded functions on $\mathcal{Y}$ by $\|f(\cdot)\|_\infty := \sup_{y\in\mathcal{Y}}|f(y)|$. Then it holds that

$$\begin{aligned}
|\sup_y M_k(y) - \sup_y \Psi(y)| &= |\sup_y |M_k(y)| - \sup_y |\Psi(y)|| \\
&= |\|M_k(\cdot)\|_\infty - \|\Psi(\cdot)\|_\infty| \\
&\leq \|M_k(\cdot)-\Psi(\cdot)\|_\infty \\
&= \sup_y |M_k(y)-\Psi(y)|,
\end{aligned}$$

where we made use of the fact that $\psi \geq 0$ in the first equality, and the reverse triangle inequality in the second-to-last step. By the uniform strong law of large numbers (Ferguson, 2017),

$$\lim_{k\to\infty}\sup_y |M_k(y)-\Psi(y)| = 0$$

almost surely. Therefore, by the inequality we just showed, we also have

$$\lim_{k\to\infty}|\sup_y M_k(y) - \sup_y \Psi(y)| \leq \lim_{k\to\infty}\sup_y |M_k(y)-\Psi(y)| = 0$$

almost surely, i.e.,

$$\lim_{k\to\infty} C_k = \lim_{k\to\infty}\sup_y M_k(y) = \sup_y \mathbb{E}[\psi(X,y)] := c$$

almost surely. Then it trivially follows that C_k also converges to c in probability, and that $\lim_{k\to\infty}\mathbb{E}[C_k] = c$ by the bounded convergence theorem.

□

Corollary A.2.1. *Under the conditions of Theorem 2.1, if any pair of non-negative (g,λ) satisfies $\mathbb{E}[C_k] \leq 0$, where $C_k := \sup_y \frac{1}{k}\sum_i \psi(X_i,y) = \sup_y \frac{1}{k}\sum_i \psi(X_i,y)$ and $\psi(x,y) := \frac{\exp(-\lambda\rho(x,y))}{g(x)} - 1$ as before, then for every $D > D_{min}$, it holds that $\mathbb{E}[-\log g_\theta(X)] - \lambda D \leq R(D)$.*

Proof. The feasibility of such (g,λ) trivially follows from the fact that $c \leq \mathbb{E}[C_k] \leq 0$ by Theorem 2.2. The fact that the resulting function $\mathbb{E}[-\log g_\theta(X)] - \lambda D$ underestimates $R(D)$ for *every* $D > D_{\text{min}}$ follows from the variational characterization of $R(D)$ as the upper envelope of tangent linear underestimators (see, e.g., lemma 1.2 of Csiszár (1974)). □

A.3 Proposed Stochastic Optimization Algorithm

Our algorithm aims to find a solution to the following constrained optimization problem:

$$\max_{\theta,\lambda\geq 0} f(\theta,\lambda), \quad \text{where } f(\theta,\lambda) := \mathbb{E}[-\log g_\theta(X)] - \lambda D \tag{10}$$

$$\text{subject to} \quad \mathbb{E}[C_k(\theta,\lambda)] = \mathbb{E}\left[\sup_y \frac{1}{k}\sum_i \frac{\exp(-\lambda\rho(X_i,y))}{g_\theta(X_i)}\right] - 1 \leq 0 \tag{11}$$

Algorithm 1: Proposed algorithm for estimating rate-distortion lower bound $R_L(D)$.

Requires: Model g_θ (e.g., a neural network), initial parameters $(g_{\theta_0}, \lambda_0)$, initial penalty parameter $\gamma_0 = 1$, batch sizes k, m, gradient ascent step size η, step counter $t = 0$
while *True* **do**
  `// Find a local maximum of` $\hat{\ell}(\theta, \lambda)$ `given the current` γ_t
  while (θ_t, λ_t) *not converged* **do**
    Draw two batches of data samples $\{x_1, ..., x_k\}$ and $\{x_1, ..., x_m\}$
    $y^*, \hat{C}_k =$ `estimate_constraint` $(\theta_t, \lambda_t, \{x_1, ..., x_k\})$
    Update parameters by $(\theta_{t+1}, \lambda_{t+1}) = (\theta_t, \lambda_t) +$
    $\eta \nabla_{(\theta,\lambda)} \left(\frac{1}{m} \sum_{j=1}^m - \log g_{\theta_t}(x_j) - \gamma_t \max\{0, \frac{1}{k} \sum_{i=1}^k \frac{\exp\{-\lambda_t \rho(x_i, y^*)\}}{g_{\theta_t}(x_i)} - 1\} \right)$
    Update training step counter $t := t + 1$
  end
  $y^*, \hat{C}_k =$ `estimate_constraint_global_opt` $(\theta_t, \lambda_t, \{x_1, ..., x_k\})$
  if $\hat{C}_k \leq 0$ **then**
    `// True constraint` c `is likely satisfied`
    Return solution (θ_t, λ_t), and lower bound estimate $\hat{R}_L = \sum_{j=1}^m - \log g_{\theta_t}(x_j)$
  else
    Increase penalty parameter, $\gamma_t := \beta \gamma_t$, with e.g., $\beta = 2$ or 10
  end
end
Subroutine `estimate_constraint` $(\theta, \lambda, \{x_1, ..., x_k\})$:
  Run gradient ascent on $\phi(y) := \sum_{i=1}^k \exp\{-\lambda \rho(x_i, y)\} / g_\theta(x_i)$, until converging to y^*
  Compute $\hat{C}_k = \phi(y^*) - 1$
  Return $(y^*, \hat{C}_k)$
Subroutine `estimate_constraint_global_opt` $(\theta, \lambda, \{x_1, ..., x_k\})$:
  Find global maximizer $y^* = \arg\max_y \phi(y)$
  Compute $\hat{C}_k = \phi(y^*) - 1$
  Return $(y^*, \hat{C}_k)$

which, by Theorem 2.2 and Corollary A.2.1, would guarantee a feasible solution to the original problem and yield a valid lower bound on $R(D)$.

We apply the penalty method and solve the following sequence of unconstrained problems:

$$\max_{\theta, \lambda \geq 0} f(\theta, \lambda) - \gamma_t [\mathbb{E}[C_k(\theta, \lambda)]]^+$$

where we have essentially replaced $c(\theta, \lambda)$ in Eq. 6 by its overestimator $\mathbb{E}[C_k(\theta, \lambda)]$.

We estimate $\mathbb{E}[C_k(\theta, \lambda)]]^+$ by the plug-in estimator:

$$[\mathbb{E}[C_k]]^+ \approx \left[\frac{1}{s} \sum_{j=1}^s \hat{C}_k^j \right]^+ = \left[\frac{1}{s} \sum_{j=1}^s \sup_y \frac{1}{k} \sum_{i=1}^k \psi(x_{s[i]}, y) \right]^+$$

where we draw s mini-batches of data samples, each mini-batch containing k samples. Since computing each $\hat{C}_k$ requires solving a global optimization problem with respect to y, we simply use $s = 1$ for computational efficiency, i.e.,

$$[\mathbb{E}[C_k]]^+ \approx \left[\hat{C}_k \right]^+ = \left[\sup_y \frac{1}{k} \sum_{i=1}^k \psi(x_i, y) \right]^+$$

The sampled loss function of the unconstrained penalty (sub)problem is then

$$\max_{\theta, \lambda \geq 0} \hat{f}_m(\theta, \lambda) - \gamma_t [\hat{C}_k(\theta, \lambda)]^+$$

To apply stochastic gradient ascent, we derive the gradient for each of the two terms with respect to θ (the gradient with respect to λ is similar). The gradient of $\hat{f}$ is simply a sample average:

$$\nabla_\theta \hat{f}_m = \nabla_\theta \left(\frac{1}{m} \sum_{j=1}^{m} -\log g_\theta(x_j) - \lambda D \right) = \frac{1}{m} \sum_{j=1}^{m} -\nabla_\theta \log g_\theta(x_j) - \lambda D$$

The gradient with respect to the penalty term requires more work. First we need to compute $\hat{C}_k$ by solving a global optimization problem, finding a y^* such that

$$\hat{C}_k = \sup_y \frac{1}{k} \sum_{i=1}^{k} \psi_\theta(x_i, y) = \frac{1}{k} \sum_{i=1}^{k} \psi_\theta(x_i, y^*),$$

and then we have

$$\nabla_\theta [\hat{C}_k]^+ = \begin{cases} 0, & \text{if } \hat{C}_k \leq 0 \\ \nabla_\theta \hat{C}_k = \nabla_\theta \frac{1}{k} \sum_{i=1}^{k} \psi_\theta(x_i, y^*) = \frac{1}{k} \sum_{i=1}^{k} \nabla_\theta \psi_\theta(x_i, y^*), & \text{if } \hat{C}_k > 0 \end{cases}$$

where in the case $\hat{C}_k > 0$, we plug in y^* and remove the dependence of the gradient on y by appealing to an envelope theorem (Milgrom & Segal, 2002).

We give a pseudocode implementation of our proposed stochastic optimization procedure in Algorithm 1. We follow a simple exact penalty method (Nocedal & Wright, 2006), but other constrained optimization methods like augmented Lagrangian can also be used instead.

The subroutine `estimate_constraint` in the inner loop is intended as a computationally cheaper alternative to the exact (but expensive) `estimate_constraint_global_opt` procedure, which we reserve for verifying the feasibility of a candidate solution. We give an example implementation of `estimate_constraint` that simply performs a single gradient ascent run on $\phi(y)$ to estimate $\hat{C}_k$ (and its gradient), which proved sufficient for training in our experiments; however, since gradient ascent only gives us a local maximum of $\phi(y)$ in general, this naive implementation of the algorithm may run into convergence issues.

A.4 MORE EXPERIMENTAL DETAILS AND RESULTS

We used a simple 2 layer fully connected network with a scalar output for $\log g$, with 8 hidden units in each layer. We parameterized λ as $\lambda = \exp\tilde{\lambda}$ using an unconstrained variable $\tilde{\lambda}$. We used the Adam optimizer for all gradient based optimization. During training, we set batch size $k = m = 100$, and implement `estimate_constraint` by a single gradient ascent run, initializing y to the data sample in the batch that achieves the highest value of $\phi(\cdot)$ (we find that initializing y to the batch mean $\frac{1}{k}\sum_{i=1}^{k} x_i$ also works well).

We used the (g^*, λ^*) trained after 3500 steps for estimating the final $R(D)$ lower bound. We first verified their feasibility by running the global optimization procedure of Carreira-Perpinan (2000) with $k = 10000$, as assured by the negative values of the resulting $\hat{C}_{10000}$, which were $-0.199, -0.106, -0.254, -0.202, -0.115, -0.068, -0.052$, for $D = 0.02, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8$. Then we compute $\hat{f}_m$ from $m = 10000$ samples, repeat for 10 different batches, and plot the mean $\hat{f}_m$ as the lower bound $\hat{R}_L(D)$ in Figure 1 (Middle), corresponding to the (D, R) points $(0.02, 4.062), (0.05, 3.343), (0.1, 2.424), (0.2, 1.600), (0.4, 0.853), (0.6, 0.432), (0.8, 0.125)$, in terms of (MSE, bits). The standard deviations of $\hat{f}$ from the 10 different batches are between 0.005 and 0.007, so the error bars do not appear noticeable in the figure.

In our problem setting, the optimal $g(x)$ can be found analytically as $g(x) = f_X(x)(2\pi D)^{n/2}$, where $n = 2$ and f_X is the density of the source X (2D standard Gaussian). The Shannon lower bound coincides with the true $R(D)$ (Cover, 1999), giving $R(D) = h(X) - h(D)$, where $h(X), h(D)$ are the differential entropy of X, and an isotropic n-dimensional Gaussian distribution with diagonal covariances equal to D, respectively.

In the next figure, we visualize the evolution of g_θ during training for a few experiments.

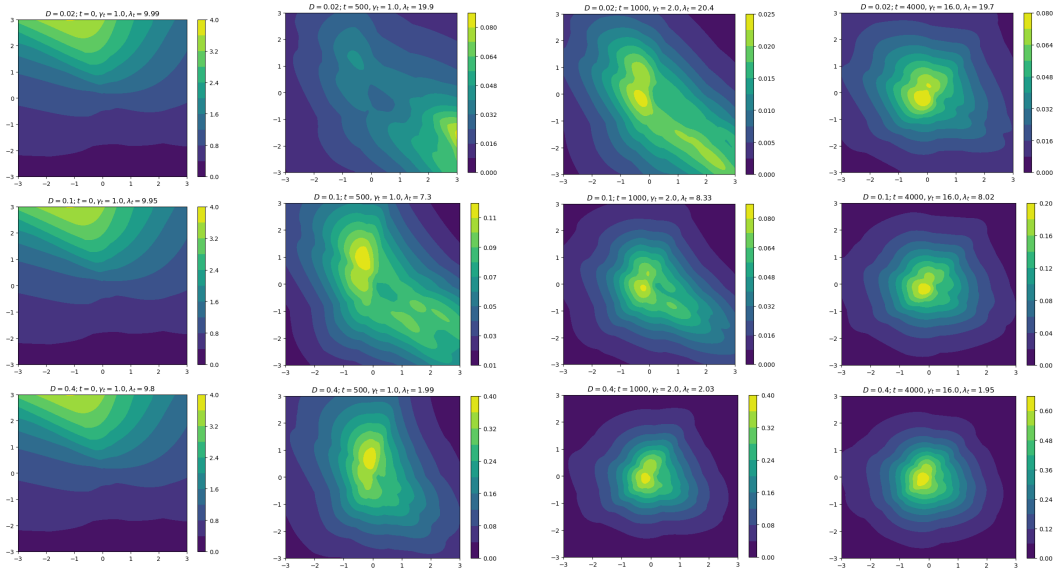

Figure 2: Example contour plots of g_θ during training. The columns correspond to the situation after step 0, 500, 1000, and 4000. Each row corresponds to a different setting of D, with $D = 0.02, 0.1, 0.4$; the training configuration (including initialization) is otherwise kept identical. In each case, the ground truth optimal g is a scaled Gaussian density with spherical contour lines, which attains its maximum value equal to D at the origin. The shape of the learned g_θ generally resembles the ground truth g, but in the lower distortion regime (e.g., when $D = 0.02$), the agreement is worse and g_θ more severely overestimates g.