LOWER BOUNDING RATE-DISTORTION FROM SAMPLES

Anonymous authors
Paper under double-blind review

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 computing such a lower bound, applicable to any data source that we have samples of. Based on a dual characterization of $R(D)$ in terms of constrained maximization, our method approximates the exact constraint function by an asymptotically unbiased sample-based estimator, allowing for stochastic optimization. On a 2D Gaussian source, we obtain a lower bound within 1 bit of the true $R(D)$ on average.

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, $R(D)$ gives 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. Therefore, the rate-distortion function, or a lower bound on the rate-distortion function, may help analyze how close lossy compression algorithms are to their theoretical performance limits, and is the motivation behind this work. 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 the theoretical optimum, 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 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

[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.

density with mean $f_\theta(y)$, ρ_0 is a squared error), so the distortion ρ is learned. The negative ELBO (Kingma & Welling, 2013) then corresponds to the Lagrangian $\mathbb{E}_X[KL(P_{Y|X}\|\tilde{P}_Y)]+\lambda\mathbb{E}[\rho(X,Y)]$, which provides the basis for learning. Huang et al. (2020) apply Annealed Importance Sampling to optimizing the Lagrangian, giving optimistic upper bounds on $R(D)$ for trained VAEs and other generative models. Like (Huang et al., 2020), our lower bound for $R(D)$ is agnostic to the choice of representations $\mathcal{Y}$ or distortion ρ, and can be directly extended to compute $R(D)$ lower bounds of trained generative models; however, unlike the upper bound of (Huang et al., 2020) which tells us what kind of rate-distortion performance is theoretically possible, our $R(D)$ lower bound establishes what kind of rate-distortion performance is so good that it 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 belonging to a general abstract probability space (Csiszar, 1974); the version appearning in (Kostina, 2016) is the basis for this work.

2 METHOD

2.1 DUAL FORMULATION OF $R(D)$

We employ an alternative characterization of $R(D)$ in terms of a constrained maximization problem:

Theorem 2.1. *(Kostina, 2016) Under basic assumptions that avoids degeneracy (see Appendix Section. A.1), for each distortion tolerance $D > D_{min} := \inf\{D : R(D) \leq \infty\}$, we have 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 delicate argument (Csiszar, 1974).

2.2 LOWER BOUNDING $R(D)$ VIA NUMERICAL OPTIMIZATION

Unfortunately, P_X is rarely known to us; e.g., when P_X is induced by natural images, characterizing the support of P_X (which is 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 a function $g \geq 0$ by a subset of functions $g_\theta \geq 0$ parameterized by a vector θ. A feasible (g_θ, λ) would still 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 function at the output layer to ensure $g_\theta \geq 0$ (or, we can parameterize $\log g$ instead for numerical stability). 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 optimization problem to obtain a $R(D)$ lower bound, $R_L(D)$:

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

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

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

$$\max_{\theta,\lambda\geq 0} f(\theta,\lambda) - \gamma_t \max\{0, c(\theta,\lambda)\} \tag{6}$$

where $\max\{0, c(\theta, \lambda)\}$ penalizes the objective by 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 compute $R_L = f(\theta_t^*, \lambda_t^*)$.

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 enough number of samples $\{x_1, x_2, ..., x_k\}$; if the empirical estimate of the constraint is satisfied, i.e., $\hat{C}_k \leq 0$, then we can claim that the true constraint c is satisfied with near certainty.

In light of this result, we propose to replace c by its sample-based estimator $\hat{C}_k$, and also use a Monte-Carlo estimate of the objective $\hat{f} := \frac{1}{m}\sum_{j=1}^m -\log g(x_j) - \lambda D$. Our proposed stochastic optimization algorithm then solves the following sequence of penalty problems,

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

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) (i.e., $\hat{C}_k \leq 0$) with high confidence. We specify Algorithm 1 in pseudocode in Appendix Sec. A.3.

2.4 Inner-loop 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 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 impossible) 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

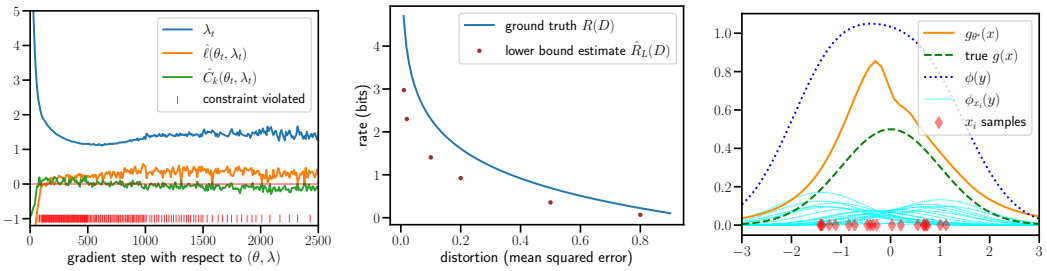

Figure 1: **Left**: trajectories of various quantities across training steps, for the $D = 0.5$ experiment; **Middle**: the final lower bound estimate $\hat{R}_L(D)$, compared to the true $R(D)$; **Right**: visualizing (along the y-axis of $\mathbb{R}^2$) a converged solution $g_{\theta^*}(x)$, the theoretically optimal $g(x)$, and the inner optimization objective of $\hat{C}_k$, $\phi(y) := \sum_{i=1}^{k} \phi_{x_i}(y)$, with $\phi_{x_i}(y) := \exp\{-\lambda^* \|x_i - y\|^2\}/\{k g_{\theta^*}(x_i)\}$, computed from $k = 20$ samples, (θ^*, λ^*) obtained with $D = 0.5$.

$\phi(y) := \frac{1}{k}\sum_i \psi(x_i, y) + 1 = \alpha \sum_k \alpha_k \exp(-\lambda \|x_i - y\|^2)$ 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 multiply it by 2 every 500 training steps (which seemed sufficient for convergence). We stop training after 2000 steps, at which point our simple estimate of constraint $\hat{C}_k$ is rarely violated 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 ($k = 1000$); in all our experiments, $\hat{C}_{1000}(g^*, \lambda^*) < 0$, so we conclude the solutions are feasible, and $\hat{f}(g^*, \lambda^*)$ provide valid lower bounds of $R(D)$ with very high probability.

We repeat the above procedure with D in $\{0.01, 0.02, 0.1, 0.2, 0.5, 0.8\}$, 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), the resulting lower bound shows reasonable agreement with the true $R(D)$, with an average gap of 0.88 bit; the gap decreases with increasing distortion tolerance, from 1.63 bit at $D = 0.01$, to 0.15 bit at $D = 0.8$. We attribute this behavior to the inaccuracy of our simple gradient ascent procedure for computing $\hat{C}_k$ during training: with a smaller D, there is less pressure to drive down λ, which makes the component functions ϕ_{x_i} more sharply peaked, and the overall inner objective $\phi(y)$ more multi-modal and harder to optimize. Fig. 1 (Right) visualizes the optimization problem for $D = 0.5$, 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 20 random samples, plugging in the solution $(g_{\theta^*}, \lambda^*)$. In this case, we see that $\hat{C}_k = \max_y \phi(y) - 1 > 0$, however this does not contradict the feasibility of $(g_{\theta^*}, \lambda^*)$; in fact, $\hat{C}_k$ decreases to become negative when we use more (e.g., $k \geq 50$) samples, as Theorem 2.2 predicts.

We provide more experimental details, and a visualization of the contours of g_θ being optimized during training, in Appendix Section. A.4. As shown in Appendix Figure. 2 the learned g_θ eventually converges to a uni-modal function with mode at the origin, roughly resembling the theoretically optimal g (which is equal to the source density scaled by a factor of $2\pi D$). Again, we see the agreement is worse for smaller D (e.g., $D = 0.01$), likely due to the inaccuracy of estimating $\hat{C}_k$ by a single gradient ascent run. This could potentially be addressed by fine-tuning the solution with a larger number of samples k (which has the effect of smoothing out the inner objective $\phi(y)$), and/or using a global optimization procedure for computing $\hat{C}_k$ instead.

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

Sicong Huang, Alireza Makhzani, Yanshuai Cao, and Roger Grosse. Evaluating lossy compression rates of deep generative models, 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.

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. *(Kostina, 2016) Suppose that the following basic assumptions are satisfied.*

1. *$R(D)$ is finite for some D, i.e., $D_{min} := \inf\{D : R(D) \leq \infty\} < \infty$;*
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 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. *(We restate the theorem for completeness). 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. *$\mathbb{E}[C_k] = \mathbb{E}[\sup_y \frac{1}{k} \sum_i \psi(X_i, y)] \geq \sup_y \mathbb{E}[\psi(X, y)] =: c$;*
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 is continuous in y, and if $\mathcal{Y}$ is compact , then C_k converges to $c = \sup_y \mathbb{E}[\psi(X, y)]$ almost surely (as well as in probability), and we have $\lim_{k\to\infty} \mathbb{E}[C_k] = c$.*

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. The proof for this resembles that of Theorem 1 in (Burda et al., 2015). Fix $y \in \mathcal{Y}$, and consider the random variable $M_k = \frac{1}{k}\sum_{i=1}^{k} \psi(X_i, y)$. If ψ is bounded, then it follows from the Strong Law of Large Numbers that M_k converges to $\mathbb{E}[M_1] = \mathbb{E}[\psi(X, y)]$ almost surely; in other words, for every ω belonging to some set with measure 1,

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

Then, for every such ω

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

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

□

A.3 Proposed Stochastic Optimization Algorithm

We give a pseudocode implementation of the penalty method, using our proposed estimators for the optimization objectives, in Algorithm 1. The subroutine `estimate_constraint` used during gradient ascent on the unconstrained penalty objective $\hat{\ell}(\theta, \lambda)$ is intended to be a computationally cheaper alternative to the exact (but expensive) `estimate_constraint_global_opt` procedure, which we reserve only 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 $\hat{C}_k$, which proved sufficient for training in our experiment; however, since gradient ascent only gives us a local maximum of $\phi(y)$, the unconstrained optimization of $\hat{\ell}(\theta, \lambda)$ with this naive implementation of $\hat{C}_k$ may run into convergence issues.

We appeal to Danskin's theorem in order to compute the gradient $\nabla_{(\theta,\lambda)}\hat{C}_k = \nabla_{(\theta,\lambda)} \sup_y \phi(y)$ by $\nabla_{(\theta,\lambda)}\phi(y^*)$, when doing gradient ascent on $\hat{\ell}(\theta, \lambda)$.

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 satisfied with high probability`
    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` $(\theta, \lambda, \{x_1, ..., x_k\})$:
  Initialize y, for example, using the mean of data samples, $y = \sum_{i=1}^{k} x_i$
  Update y by gradient ascent on $\phi(y) := \sum_{i=1}^{k} \exp\{-\lambda \rho(x_i, y)\}/g_\theta(x_i)$, until convergence
  $\hat{C}_k = \phi(y^*)$
  Return (converged $y^*, \hat{C}_k$)
Subroutine `estimate_constraint_global_opt` $(\theta, \lambda, \{x_1, ..., x_k\})$:
  Find global maximizer $y^* = \arg\max_y \sum_{i=1}^{k} \exp\{-\lambda \rho(x_i, y)\}/g_\theta(x_i)$
  $\hat{C}_k = \sum_{i=1}^{k} \exp\{-\lambda \rho(x_i, y^*)\}/g_\theta(x_i) - 1$
  Return $(y^*, \hat{C}_k)$

A.4 More experimental 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 Adam optimizer for all gradient based optimization. During training, we set $k = m = 100$, using the same batch of data samples for $\hat{f}$ and $\hat{C}_k$ for simplicity. We implement `estimate_constraint` by a single gradient ascent run, initializing y to the mean of k data samples.

We used $m = 10000$ samples when computing $\hat{f}$ for the final $R(D)$ lower bound; in Figure 1 (Middle), we plotted the mean of 10 different estimates of $\hat{f}$ (each with 10000 samples); the (mean, std) of the 10 estimates are (2.9730887, 0.011416249), (2.3030891, 0.0050365757), (1.4062412, 0.004873369), (0.92046356, 0.0046269516), (0.34891954, 0.007819665), (0.07717345, 0.0040917634), for $D = 0.01, 0.02, 0.1, 0.2, 0.5, 0.8$. Note these results are in terms of nats (unlike in Figure 1 (Middle) which we converted to bits). The values of $\hat{C}_{1000}$ for verifying feasibility of the solutions were -0.235, -0.532, -0.388, -0.335, -0.245, -0.145, and -0.04, for $D = 0.01, 0.02, 0.1, 0.2, 0.5, 0.8$, computed with the global optimization algorithm of (Carreira-Perpinan, 2000).

In our setup, 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, respectively.

In the next figure, we visualize the evolution of g_θ during training, when we run the algorithm with a few example settings of D.

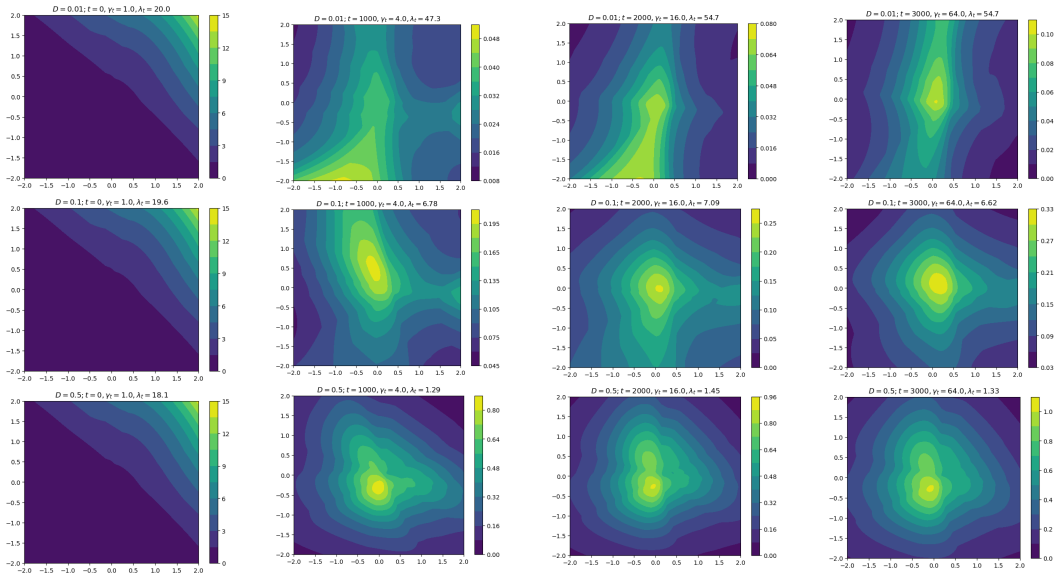

Figure 2: Contour plots of g_θ during training. The columns correspond to the situation after step 0, 1000, 2000, and 3000. Each row corresponds to a different setting of D, with $D = 0.01, 0.1, 0.5$. In each case, the ground truth optimal g is a scaled Gaussian density, such that its maximum value is equal to D at the mode (origin); we see that the learned g_θ generally overestimate the optimal g.