Estimating the Rate-Distortion Function by Wasserstein Gradient Descent

Yibo Yang [1] **Stephan Eckstein** [2] **Marcel Nutz** [3] **Stephan Mandt** [1]

Abstract

In the theory of lossy compression, the rate-distortion function $R(D)$ of a given data source characterizes the fundamental limit of compression performance by any algorithm. We propose a method to estimate $R(D)$ in the continuous setting based on Wasserstein gradient descent. While the classic Blahut–Arimoto algorithm only optimizes probability weights over the support points of its initialization, our method leverages optimal transport theory and learns the support of the optimal reproduction distribution by moving particles. This makes it more suitable for high dimensional continuous problems. Our method complements state-of-the-art neural network-based methods in rate-distortion estimation, achieving comparable or improved results with less tuning and computation effort. In addition, we can derive its convergence and finite-sample properties analytically.

Our study also applies to maximum likelihood deconvolution and regularized Kantorovich estimation, as those tasks boil down to mathematically equivalent minimization problems.

1. Introduction

Given source and reproduction alphabets $\mathcal{X}, \mathcal{Y}$ and a distortion function $\rho : (\mathcal{X}, \mathcal{Y}) \to [0, \infty)$, the rate-distortion (R-D) function of a source $X \sim P_X$ is defined by

$$R(D) = \inf_{Q_{Y|X} : \mathbb{E}_{P_X Q_{Y|X}}[\rho(X,Y)] \le D} I(X;Y), \qquad (1)$$

where $Q_{Y|X}$ is any transition kernel from $\mathcal{X}$ to $\mathcal{Y}$, which conceptually corresponds to a (possibly) stochastic compression algorithm. For a given source, $R(D)$ describes the best achievable compression cost by *any* algorithm subject to a distortion constraint (Shannon, 1959). Thus establishing $R(D)$ helps evaluate the (sub)optimality of compression algorithms on a given source and guide their development.

Unfortunately, $R(D)$ is defined by an optimization problem requiring perfect knowledge of the source. In practice, we only have indirect access to the source via samples. This has prompted research that aims to estimate $R(D)$ from data samples (Harrison and Kontoyiannis, 2008; Gibson, 2017), with recent methods (Yang and Mandt, 2022; Lei et al., 2023) inspired by deep learning. However, a drawback with deep learning-based methods is that they involve customizing neural network architectures to the data source of interest; otherwise the resulting bounds can be quite loose (Yang and Mandt, 2022). They can also require extensive hyperparameter tuning and computation resources.

In this work, we focus on upper-bounding $R(D)$ in the continuous-alphabet setting and take a different approach. Our algorithm minimizes a suitable functional over the Wasserstein space of probability measures, implemented via moving particles. Leveraging a connection between the R-D problem and optimal transport, we also develop bounds on the quality of our estimator in terms of the number of source samples and particles chosen. In practice we find our algorithm to converge quickly to a near optimum, and obtain comparable or improved results against neural network-based methods with hand-tuned architectures. Our study also applies to maximum likelihood deconvolution and regularized Kantorovich estimation, as those tasks boil down to mathematically equivalent minimization problems.

2. Lossy compression, entropic optimal transport, and MLE

2.1. Setup

For ease of presentation, we now switch to a more abstract notation without reference to random variables. We provide the precise definitions in the Supplementary Material. Let $\mathcal{X}$ and $\mathcal{Y}$ be standard Borel spaces; let $\mu \in \mathcal{P}(\mathcal{X})$ be a probability measure on $\mathcal{X}$, which should be thought of as the source distribution P_X. For a measure π on the product space $\mathcal{X} \times \mathcal{Y}$, the notation π_1 (or π_2) denotes the first (or second) marginal of π. For any $\nu \in \mathcal{P}(\mathcal{Y})$, we denote by $\Pi(\mu, \nu)$ the set of couplings between μ and ν (i.e., $\pi_1 = \mu$ and $\pi_2 = \nu$). Similarly, $\Pi(\mu, \cdot)$ denotes the set of measures

[1]Department of Computer Science, University of California, Irvine [2]Department of Mathematics, ETH Zurich [3]Departments of Statistics and Mathematics, Columbia University. Correspondence to: Yibo Yang <yibo.yang@uci.edu>.

Proceedings of the ICML 2023 Neural Compression Workshop, Honolulu, Hawaii, USA.

π with $\pi_1 = \mu$. Throughout the paper, K denotes a transition kernel (conditional distribution) from $\mathcal{X}$ to $\mathcal{Y}$, and $\mu \otimes K$ denotes the product measure formed by μ and K. Then $R(D)$ is equivalent to

$$R(D) = \inf_{K:\ \int \rho d(\mu\otimes K) \le D} H(\mu \otimes K | \mu \otimes (\mu \otimes K)_2) \quad (2)$$

where H denotes relative entropy, i.e., for two measures α, β defined on a common measurable space, $H(\alpha|\beta) := \int \log(\frac{d\alpha}{d\beta}) d\alpha$ when $\alpha \ll \beta$ and infinite otherwise.

To make the problem more tractable, we follow the approach of the classic Blahut–Arimoto algorithm (Blahut, 1972; Arimoto, 1972) and work with an equivalent unconstrained Lagrangian problem as follows. For a fixed $\lambda \ge 0$, we aim to solve the following optimization problem,

$$F_\lambda(\mu) := \inf_{\nu \in \mathcal{P}(\mathcal{Y})} \inf_{\pi \in \Pi(\mu,\cdot)} \lambda \int \rho d\pi + H(\pi|\mu\otimes\nu). \quad (3)$$

Geometrically, $F_\lambda(\mu) \in \mathbb{R}$ is the y-axis intercept of a tangent line to the $R(D)$ with slope $-\lambda$, and $R(D)$ is determined by the convex envelope of all such tangent lines (Gray, 2011). To simplify notation, we often drop the dependence on λ (e.g., we write $F(\mu) = F_\lambda(\mu)$) whenever it is harmless.

To prepare for later discussions, we write the unconstrained R-D problem as

$$F_\lambda(\mu) = \inf_{\nu \in \mathcal{P}(\mathcal{Y})} \mathcal{L}_{BA}(\mu, \nu), \quad (4)$$

$$\mathcal{L}_{BA}(\mu,\nu) := \inf_{\pi \in \Pi(\mu,\cdot)} \lambda \int \rho d\pi + H(\pi|\mu\otimes\nu) \quad (5)$$

$$= \inf_K \lambda \int \rho d(\mu\otimes K) + H(\mu\otimes K|\mu\otimes\nu), \quad (6)$$

where we refer to $\mathcal{L}_{BA}$ as the *rate function* (Harrison and Kontoyiannis, 2008). We abuse the notation to write $\mathcal{L}_{BA}(\mu,\nu) = \mathcal{L}_{BA}(\nu)$ when it is viewed as a function of ν only, and refer to it as the *rate functional*. The rate function characterizes a generalized Asymptotic Equipartition Property, where $\mathcal{L}_{BA}(\mu,\nu)$ is the asymptotically optimal cost of lossy compression of data $X \sim \mu$ using a random codebook constructed from samples of ν (Dembo and Kontoyiannis, 2002). Notably, $\mathcal{L}_{BA}$ can be simplified analytically as (Csiszár, 1974)

$$\mathcal{L}_{BA}(\mu,\nu) = \int_{\mathcal{X}} -\log \int_{\mathcal{Y}} e^{-\lambda\rho(x,y)} \nu(dy) \ \mu(dx). \quad (7)$$

In practice, the source μ is only accessible via independent samples, on the basis of which we propose to estimate its $R(D)$, or equivalently $F(\mu)$. Let μ^m denote an m-sample empirical measure of μ, i.e., $\mu^m = \frac{1}{m}\sum_{i=1}^m \delta_{x_i}$ with $x_{1,\dots,n}$ being independent samples from μ, which should be thought of as the "training data". Following Harrison and Kontoyiannis (2008), we consider two kinds of (plug-in) estimators for $F(\mu)$: (1) the non-parametric estimator $F(\mu^m)$, and (2) the parametric estimator $F^{\mathcal{H}}(\mu^m) := \inf_{\nu\in\mathcal{H}} \mathcal{L}_{BA}(\mu^m, \nu)$, where $\mathcal{H}$ is a family of probability measures on $\mathcal{Y}$. Harrison and Kontoyiannis (2008) showed that under rather broad conditions, both kinds of estimators are strongly consistent, i.e., $F(\mu^m)$ converges to $F(\mu)$ (and respectively, $F^{\mathcal{H}}(\mu^m)$ to $F^{\mathcal{H}}(\mu)$) with probability one as $m \to \infty$.

2.2. Optimal transport perspective

The R-D problem turns out to be closely related to entropic optimal transport, which we will exploit in Sec. 4.2 to obtain sample complexity results under our approach. For $\epsilon > 0$, the entropic optimal transport problem is defined by (Peyré and Cuturi, 2019)

$$\mathcal{L}_{EOT}(\mu,\nu) := \inf_{\pi\in\Pi(\mu,\nu)} \int \rho d\pi + \epsilon H(\pi|\mu\otimes\nu). \quad (8)$$

Interpreting $\mathcal{L}_{EOT}$ loosely as a distance between probability measures, we consider the "projection" of μ onto $\mathcal{P}(\mathcal{Y})$:

$$\inf_{\nu\in\mathcal{P}(\mathcal{Y})} \mathcal{L}_{EOT}(\nu). \quad (9)$$

In the OT literature this is known as the (regularized) Kantorovich estimator (Bassetti et al., 2006) for μ, and can also be viewed as a Wasserstein barycenter problem (Agueh and Carlier, 2011).

With the identification $\epsilon = \lambda^{-1}$, the above problem turns out equivalent to the R-D problem (4): compared to $\mathcal{L}_{BA}$ (5), the extra constraint on the second marginal of π in $\mathcal{L}_{EOT}$ (8) is redundant at the optimal ν. More precisely, we have

$$\arg\min_\nu \mathcal{L}_{EOT}(\nu) = \arg\min_\nu \mathcal{L}_{BA}(\nu) \qquad \text{and}$$
$$\inf_\nu \mathcal{L}_{EOT}(\nu) = \inf_\nu \lambda^{-1}\mathcal{L}_{BA}(\nu).$$

The proof follows from a basic property of relative entropy (Csiszár, 1974, Lemma 1.3); see Appendix.

2.3. Statistical interpretations

The R-D problem (4), and its equivalent EOT "projection" problem (9), also admit a statistical interpretation as maximum likelihood estimation (MLE). The connection between R-D and model estimation has been observed in the information theory and compression literature (Harrison and Kontoyiannis, 2008; Ballé et al., 2017; Theis et al., 2017; Yang and Mandt, 2022), and Rigollet and Weed (2018) noted the connection between the EOT problem (9) and maximum-likelihood deconvolution (Carroll and Hall, 1988). We give a unifying account in Sec. 8 of the Supplementary Material.

3. Related work

The BA algorithm (Blahut, 1972; Arimoto, 1972) is the default method for computing $R(D)$ in the finite-alphabet case. It solves the optimization problem (8) by coordinate ascent w.r.t. K and β, which can be done in matrix/vector operations. When the alphabets are not finite, the algorithm no longer applies, as it is unclear how to tractably represent the measure and kernel K and to perform the required integrals. The common workaround is to do a discretization step and then apply BA on the resulting discrete problem (Gray and Neuhoff, 1998). Grid-based discretization quickly becomes infeasible in higher dimensions (Yang and Mandt, 2022; Lei et al., 2023), we therefore consider randomly discretize the alphabets to consist of samples (Harrison and Kontoyiannis, 2008; Lei et al., 2023) in our experiments.

To overcome the limitations of the BA algorithm, Yang and Mandt (2022) proposed to parameterize the transition kernel K and reproduction distribution of the BA algorithm by neural networks, and optimize the same objective (8) by (stochastic) gradient descent. The resulting method essentially trains a VAE (Kingma and Welling, 2013), which we dub the *RD-VAE*. Closely related, Lei et al. (Lei et al., 2023) proposed Neural Estimator of the R-D function (NERD), which instead optimizes the form of the rate functional in (7), via gradient descent on the parameters of parameterized by a neural network. The inner integral (7) of w.r.t. is non-trivial to compute exactly, and is estimated in practice with a plugin estimator using samples from .

Concurrent work by Yan et al. (2023) proposes to estimate Gaussian mixtures by gradient descent in the Fisher-Rao-Wasserstein (FRW) geometry (Chizat et al., 2018). Their problem is equivalent to an R-D estimation problem (see Sec. 2.3), and our algorithm is closely related to theirs. Essentially, the BA algorithm is equivalent to gradient descent in the Fisher-Rao geometry with a unit step size, and our hybrid algorithm (Sec. 4.3) corresponds to gradient descent in the FRW geometry with a different interpolation factor (Chizat et al., 2018). Yan et al. (2023) prove that, in an idealized setting with in nite particles, FRW gradient descent does not get stuck at local minima, whereas our convergence and sample-complexity results (Prop. 10.2, 10.4) hold for any nite number of particles. We additionally consider larger-scale problems and the stochastic optimization setting.

Lei et al. (2022) also noted the connection between the R-D problem (4) and EOT projection (9), and optimized the latter similarly to Genevay et al. (2018). Wu et al. (2022) proposed to solve the dual of the R-D problem (4) in the nite-alphabet case using Sinkhorn's algorithm

4. Proposed method

Let $X = Y = \mathbb{R}^d$ and be continuously differentiable. In this section, we introduce the gradient descent algorithm in Wasserstein space to solve the problems (4) and (9).

4.1. Wasserstein gradient descent (WGD)

Abstractly, Wasserstein gradient descent updates the variational measure to its pushforward $\tilde{}$ under the map $(\mathrm{id} \quad)$, for a function $: \mathbb{R}^d \to \mathbb{R}^d$ called the *Wasserstein gradient* of L at (see below) and a step size To implement this scheme, we represent as a convex combination of Dirac measures, $= \sum_{i=1}^n w_i \, _{x_i}$ with locations $x_{1;\dots;n}$ and weights $w_{1;\dots;n}$. The algorithm moves each particle x_i in the direction of (x_i), more precisely, $\tilde{} = \sum_{i=1}^n w_i \, _{x_i \quad (x_i)}$.

Since the objectives (4) and (9) appear as integrals w.r.t. the data distribution , we can also apply stochastic optimization and perform stochastic gradient descent on mini-batches with size m. This allows us to handle a very large or in nite amount of data samples, or when the source is continuous. We formalize the procedure in Algorithm 2.

Algorithm 1 Wasserstein gradient descent

Inputs: Loss function $L \in \{L_{BA}; L_{EOT}\}$; data distribution $\in \mathcal{P}(\mathbb{R}^d)$; initial measure $^{(0)} \in \mathcal{P}(\mathbb{R}^d)$; total number of iterations N; step sizes $_1; \dots; _N$; batch size $m \in \mathbb{N}$.
for $t = 1; \dots; N$ do
  if support of P_m contains more than m points then
    $_m \quad \frac{1}{m} \sum_{i=1}^m _{x_i}$ for $x_1; \dots; x_m$ independent samples from
    $_t$ Wasserstein gradient of $L(_m;)$ at $^{(t-1)}$ {see De nition 10.1}
  else
    $_t$ Wasserstein gradient of $L(;)$ at $^{(t-1)}$ {see De nition 10.1}
  end if
  $^{(t)} \quad (\mathrm{id} \quad _t \; _t)_{\#} \; ^{(t-1)}$ {"$\#$" denotes pushforward}
end for
Return: $^{(N)}$

In essence, our algorithm simulates the gradient ow of the BA functional L_{BA} (alternatively, L_{EOT}) in the Wasserstein space over Y (Ambrosio et al., 2008). The key step is computing the Wasserstein gradient, de ned below.

De nition 4.1. For a functional $L : \mathcal{P}(Y) \to \mathbb{R}$ and $\in \mathcal{P}(Y)$, we say that $V_L( ) : \mathbb{R}^d \to \mathbb{R}$ is a rst variation of L at if

$$\lim_{" \to 0} \frac{L((1 \quad ") \quad + " \tilde{}) \quad L( )}{"} = \int V_L( ) \, d(\tilde{} \quad);$$

for all $\sim \in \mathcal{P}(Y)$. We call its (Euclidean) gradient $\nabla V_L(\ ) : \mathbb{R}^d \to \mathbb{R}^d$, if it exists, the Wasserstein gradient of L at $\ $.

As we discuss in the Appendix, for $L = L_{BA}$ the first variation can be evaluated in closed form, whereas Sinkhorn's algorithm is required for $L = L_{EOT}$. An auto-differentiation package can then be used to evaluate the gradient of the first variation, and thus the Wasserstein gradient. Further, we prove that WGD on $L \in \{L_{BA}, L_{EOT}\}$ converges to at least a local optimum under mild conditions, in Prop. 10.2.

4.2. Finite-sample properties

In the case $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ and $\ell(x, y) = \|x - y\|^2$, we also leverage the equivalence between the R-D and EOT projection problem (see Sec. 2.2) and the result from (Mena and Niles-Weed, 2019) to derive finite-sample properties of our estimator, detailed in Appendix Prop. 10.4. Informally, for loss functional $L \in \{L_{BA}, L_{EOT}\}$ and a sub-Gaussian source, we show the optimized population loss converges to the global optimum (4) at a rate of $\mathcal{O}(\frac{1}{\sqrt{n}})$ where n is the number of particles, and the optimized empirical loss converges at a rate of $\mathcal{O}(\frac{1}{\sqrt{m}})$ where m is the number of source samples. This strengthens existing asymptotic results for the empirical R-D estimators of Harrison and Kontoyiannis (2008).

4.3. Hybrid algorithm

In the BA algorithm, the support of the sequence of $^{(t)}$ is restricted to that of the (possibly bad) initialization $^{(0)}$. On the other hand, Wasserstein gradient descent (Algorithm 2) only evolves the particle locations of , but not its weights. We therefore consider a hybrid algorithm where we alternate between Wasserstein gradient descent and the BA update steps, allowing us to optimize the particle weights as well. In Sec. 5.1, we observe accelerated convergence compared to the plain WGD algorithm, but note that the hybrid algorithm (like the BA algorithm) does not directly apply in the stochastic optimization setting, as performing BA updates on random mini-batches can lead to divergence.

5. Experiments

We study the empirical performance of the proposed Wasserstein gradient descent algorithm (WGD) and its hybrid variant, and compare with BA (Blahut, 1972; Arimoto, 1972), and neural network-based methods RD-VAE (Yang and Mandt, 2022) and NERD (Lei et al., 2023). We experimented with WGD for both L_{BA} and L_{EOT}. Empirically we found them to give similar results, while the former to be 10 to 100 times faster computationally; we therefore focus on WGD for L_{BA} in the discussions below. More details are given in the Appendix.

Figure 1. Top: Training losses shaded by one standard deviation over random seeds. The proposed WGD algorithms converge quickly to the theoretically optimal value OPT. Bottom: The final returned by the algorithms. The WGD algorithms recover the true (cyan) whereas the alternative methods fail to.

5.1. Deconvolution

We experiment with various methods on the maximum-likelihood deconvolution problem (Sec. 2.3), where the true is the uniform measure on the unit circle. The problem admits an analytical solution, and we numerically compute the optimal objective value $OPT = F(\ )$. To ensure roughly comparable computation complexity per iteration, we use the same for BA, NERD, and the proposed WGD method. We set a relatively small $= 20$ to mimic the high-dimensional scenario. We run the various methods and plot their training losses averaged over 5 random seeds in Fig. 1; the test losses are similar and given in the Appendix. We observe that the proposed WGD algorithms converge significantly faster than the alternative methods, and to the optimal value OPT. Furthermore, the hybrid algorithm converges even faster than the plain WGD algorithm. The other algorithms reach sub-optimal solutions, and we visualize their various failure modes in Fig. 1.

5.2. Higher-dimensional data

To demonstrate the scalability of our method, we also experiment on the physics and speech data from (Yang and Mandt, 2022). As the source distribution in each problem contains too many data points to be computed on directly, we focus on WGD and NERD (Lei et al., 2023) using mini-batch

SGD. As shown in Appendix Fig. 6, we obtain comparable or tighter R-D upper bounds than NERD, while using a significantly smaller number of particles.

References

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

Matthew T. Harrison and Ioannis Kontoyiannis. Estimation of the rate–distortion function. IEEE Transactions on Information Theory, 54(8):3757–3762, 2008. doi: 10.1109/tit.2008.926387.

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

Yibo Yang and Stephan Mandt. Towards empirical sandwich bounds on the rate-distortion function. In International Conference on Learnning Representations, 2022.

Eric Lei, Hamed Hassani, and Shirin Saeedi Bidokhti. Neural estimation of the rate-distortion function with applications to operational source coding. IEEE Journal on Selected Areas in Information Theory, 2023.

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.

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

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

Amir Dembo and L Kontoyiannis. Source coding, large deviations, and approximate pattern matching. IEEE Transactions on Information Theory, 48(6):1590–1615, 2002.

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

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. Foundations and Trends in Machine Learning, 11(5-6):355–607, 2019.

Federico Bassetti, Antonella Bodini, and Eugenio Regazzini. On minimum kantorovich distance estimators. Statistics & probability letters, 76(12):1298–1302, 2006.

Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. SIAM Journal on Mathematical Analysis, 43(2):904–924, 2011.

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

Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. International Conference on Learning Representations, 2017.

Philippe Rigollet and Jonathan Weed. Entropic optimal transport is maximum-likelihood deconvolution. Comptes Rendus Mathematique, 356(11-12):1228–1235, 2018.

Raymond J Carroll and Peter Hall. Optimal rates of convergence for deconvolving a density. Journal of the American Statistical Association, 83(404):1184–1186, 1988.

Robert M. Gray and David L. Neuhoff. Quantization. IEEE transactions on information theory, 44(6):2325–2383, 1998.

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

Yuling Yan, Kaizheng Wang, and Philippe Rigollet. Learning gaussian mixtures using the wasserstein- fisher-rao gradient flow. arXiv preprint arXiv:2301.01766, 2023.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An interpolating distance between optimal transport and fisher–rao metrics. Foundations of Computational Mathematics, 18:1–44, 2018.

Eric Lei, Hamed Hassani, and Shirin Saeedi Bidokhti. Neural estimation of the rate-distortion function for massive datasets. In 2022 IEEE International Symposium on Information Theory (ISIT), pages 608–613. IEEE, 2022.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. International Conference on Artificial Intelligence and Statistics, pages 1608–1617. PMLR, 2018.

Shitong Wu, Wenhao Ye, Hao Wu, Huihui Wu, Wenyi Zhang, and Bo Bai. A communication optimal transport approach to the computation of rate distortion functions. arXiv preprint arXiv:2212.10098, 2022.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. Gradient flows in metric spaces and in the space of probability measures Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, second edition, 2008.

Gonzalo Mena and Jonathan Niles-Weed. Statistical bounds for entropic optimal transport: sample complexity and the central limit theorem. Advances in Neural Information Processing Systems, 32, 2019.

Erhan Çinlar. Probability and stochastics, volume 261. Springer, 2011.

Gerald B Folland. Real analysis: modern techniques and their applications, volume 40. John Wiley & Sons, 1999.

Yury Polyanskiy and Yihong Wu. Information theory: From coding to learning. Book draft, 2022.

Jack Kiefer and Jacob Wolfowitz. Consistency of the maximum likelihood estimator in the presence of in nitely many incidental parameters. The Annals of Mathematical Statistics, pages 887–906, 1956.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In International Conference on Machine Learning, pages 2256–2265. PMLR, 2015.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. Journal of the American statistical Association, 112(518):859–877, 2017.

MJ Beal and Z Ghahramani. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. Bayesian statistics, 7(453-464):210, 2003.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society: series B (methodological), 39(1):1–22, 1977.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. Machine learning, 37: 183–233, 1999.

Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning, 1(1–2): 1–305, 2008.

Michael Irwin Jordan. Learning in graphical models. MIT press, 1999.

Marcel Nutz. Introduction to entropic optimal transport. Lecture notes, Columbia University, 2021. https://www.math.columbia.edu/~mnutz/docs/EOT_lecture_notes.pdf .

Guillaume Carlier, Lénaïc Chizat, and Maxime Laborde. Lipschitz continuity of the Schrödinger map in entropic optimal transport. arXiv preprint arXiv:2210.00225, 2022.

Toby Berger. Rate distortion theory, a mathematical basis for data compression. Prentice Hall, 1971.

Stephan Eckstein and Marcel Nutz. Convergence rates for regularized optimal transport via quantization. arXiv preprint arXiv:2208.14391, 2022.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax .

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

Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In International Conference on Learning Representations, 2015.

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.

Appendix

We review probability theory background and explain our notation from the main text in Section 6, elaborate on the connections between the R-D estimation problem and variational inference/learning in Section 9, give proofs of formal results for Wasserstein gradient descent in Section 10, provide an example implementation in Section 11, and finally provide additional experimental results and details in Section 12.

6. Notions from probability theory

In this section we collect notions of probability theory used in the main text. See, e.g., (Çinlar, 2011) or (Folland, 1999) for more background.

Marginal and conditional distributions. The source and reproduction spaces $\mathcal{X}, \mathcal{Y}$ are equipped with sigma-algebras $\mathcal{A}_{\mathcal{X}}$ and $\mathcal{A}_{\mathcal{Y}}$, respectively. Let $\mathcal{X} \times \mathcal{Y}$ denote the product space equipped with the product sigma algebra $\mathcal{A}_{\mathcal{X}} \otimes \mathcal{A}_{\mathcal{Y}}$. For any probability measure π on $\mathcal{X} \times \mathcal{Y}$, its first marginal is

$$\pi_1(A) := \pi(A \times \mathcal{Y}), \quad A \in \mathcal{A}_{\mathcal{X}},$$

which is a probability measure on $\mathcal{X}$. When π is the distribution of a random vector (X, Y), then π_1 is the distribution of X. The second marginal of π is defined analogously as

$$\pi_2(B) := \pi(\mathcal{X} \times B), \quad B \in \mathcal{A}_{\mathcal{Y}}.$$

For two measures μ, ν defined on a common measurable space, the notation $\mu \ll \nu$ denotes that μ is absolutely continuous with respect to ν, i.e., $\nu(A) = 0 \implies \mu(A) = 0$ for every measurable set A.

A Markov kernel or conditional distribution $K(x, dy)$ is a map $\mathcal{X} \times \mathcal{A}_{\mathcal{Y}} \to [0, 1]$ such that

1. $K(x, \cdot)$ is a probability measure on $\mathcal{Y}$ for each $x \in \mathcal{X}$;
2. the function $x \mapsto K(x, B)$ is measurable for each set $B \in \mathcal{A}_{\mathcal{Y}}$.

When speaking of the conditional distribution of a random variable Y given another random variable X, we occasionally also use the notation $Q_{Y|X}$ from information theory (Polyanskiy and Wu, 2022). Then $Q_{Y|X=x}(B) = K(x, B)$ is the conditional probability of the event $\{Y \in B\}$ given $X = x$.

Suppose that a probability measure μ on $\mathcal{X}$ is given, in addition to a kernel $K(x, dy)$. Together they define a unique measure $\pi = \mu K$ on the product space $\mathcal{X} \times \mathcal{Y}$. For a rectangle set $A \times B \in \mathcal{A}_{\mathcal{X}} \otimes \mathcal{A}_{\mathcal{Y}}$,

$$\pi(A \times B) = \int_A \mu(dx) K(x, B), \quad A \in \mathcal{A}_{\mathcal{X}}, B \in \mathcal{A}_{\mathcal{Y}}.$$

The measure $\pi := \mu K$ has first marginal $\pi_1 = \mu$.

The classic product measure is a special case of this construction. Namely, when a measure ν on $\mathcal{Y}$ is given, using the constant kernel $K(x, dy) := \nu(dy)$ (which does not depend on x) gives rise to the product measure $\mu \otimes \nu$,

$$(\mu \otimes \nu)(A \times B) = \mu(A)\, \nu(B), \quad A \in \mathcal{A}_{\mathcal{X}}, B \in \mathcal{A}_{\mathcal{Y}}.$$

Under mild conditions (for instance when $\mathcal{X}, \mathcal{Y}$ are Polish spaces equipped with their Borel sigma algebras, as in the main text), any probability measure π on $\mathcal{X} \times \mathcal{Y}$ is of the above form. Namely, the disintegration theorem asserts that π can be written as $\pi = \pi_1 K$ for some kernel K. When π is the joint distribution of a random vector (X, Y), this says that there is a measurable version of the conditional distribution $Q_{Y|X}$.

Table 1. Guide to notation and their interpretations in various problem domains. "LVM" stands for latent variable modeling, "NPMLE" stands for non-parametric MLE. The R-D problem (3) is equivalent to a "projection" problem in entropic optimal transport (discussed in Sec. 2.2) and statistical problems involving maximum-likelihood estimation (see discussion in Sec. 2.3 and below).

Context	$= P_X$	(x, y)	$K = Q_{Y\|X}$	$= Q_Y$
OT	source distribution	transport cost	"transport plan"	target distribution
R-D	data distribution	distortion criterion	compression algorithm	codebook distribution
LVM/NPMLE	data distribution	"$-\log p(x\|y)$"	variational posterior	prior distribution
deconvolution	noisy measurements	"noise kernel"	—	noiseless model

Optimal transport. Given a measure μ on $\mathcal{X}$ and a measurable function $T : \mathcal{X} \to \mathcal{Y}$, the pushforward (or image measure) of μ under T is a measure on $\mathcal{Y}$, given by

$$T_{\#}\mu(B) = \mu(T^{-1}(B)), \quad B \in \mathcal{A}_{\mathcal{Y}}.$$

If T is seen as a random variable and μ as the baseline probability measure, then $T_{\#}\mu$ is simply the distribution of T.

Suppose that μ and ν are probability measures on $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ with finite second moment. As introduced in the main text, $\Pi(\mu, \nu)$ denotes the set of couplings, i.e., measures on $\mathcal{X} \times \mathcal{Y}$ with $\pi_1 = \mu$ and $\pi_2 = \nu$. The 2-Wasserstein distance $W_2(\mu, \nu)$ between μ and ν is defined as

$$W_2(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \left(\int \|y - x\|^2 \, \pi(dx, dy) \right)^{1/2}.$$

This indeed defines a metric on the space of probability measures with finite second moment.

7. Relation between the R-D estimation and EOT "projection" problems

Here we show that the R-D problem and EOT "projection" problems share the same optimizer and optimal objective values up to rescaling. Recall the definitions of the BA and EOT functionals:

$$L_{EOT}(\pi, \alpha) := \inf_{\pi \in \Pi(\mu, \alpha)} \int d\pi + H(\pi \,|\, \mu \otimes \alpha). \tag{10}$$

$$L_{BA}(\pi, \alpha) := \inf_{\pi \in \Pi(\mu, \alpha)} \int d\pi + H(\pi \,|\, \mu \otimes \alpha) \tag{11}$$

$$= \inf_{K} \int d(\mu K) + H(\mu K \,|\, \mu \otimes \alpha). \tag{12}$$

We first relate the optimal values of the two functionals, with $\beta = \frac{1}{\gamma}$:

$$\inf_{\alpha \in \mathcal{P}(\mathcal{Y})} L_{EOT}(\alpha) = \inf_{\alpha \in \mathcal{P}(\mathcal{Y})} \inf_{\pi \in \Pi(\mu, \alpha)} \int d\pi + H(\pi \,|\, \mu \otimes \alpha) \tag{13}$$

$$= \inf_{\pi \in \Pi(\mu, \pi_2)} \int d\pi + H(\pi \,|\, \mu \otimes \pi_2) \tag{14}$$

$$= \inf_{\pi \in \Pi(\mu, \cdot)} \int d\pi + \inf_{\tilde{\alpha} \in \mathcal{P}(\mathcal{Y})} H(\pi \,|\, \mu \otimes \tilde{\alpha}) \tag{15}$$

$$= \inf_{\tilde{\alpha} \in \mathcal{P}(\mathcal{Y})} \inf_{\pi \in \Pi(\mu, \cdot)} \int d\pi + H(\pi \,|\, \mu \otimes \tilde{\alpha}) \tag{16}$$

$$= \frac{1}{\gamma} \inf_{\tilde{\alpha} \in \mathcal{P}(\mathcal{Y})} L_{BA}(\tilde{\alpha}); \tag{17}$$

where the third line makes use of a well-known upper bound on mutual information (Polyanskiy and Wu, 2022, Theorem 4.1, "golden formula").

For either problem, the uniqueness of a minimizer follows by strict convexity of relative entropy. Existence of a minimizer holds under mild conditions, for instance if $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ and (x, y) is a coercive lower semicontinuous function of x (Csiszár, 1974, p. 66).

Finally, the minimizers for both problems clearly coincide when they exist: if $= \arg\min_{\in\mathcal{P}(\mathcal{Y})} L_{BA}(\ )$, then $L_{EOT}(\ ) = \frac{1}{}L_{BA}(\ ) = \frac{1}{}\min_{\in\mathcal{P}(\mathcal{Y})} L_{BA}(\ ) = \min_{\in\mathcal{P}(\mathcal{Y})} L_{EOT}(\ )$; the same argument applies to $= \arg\min_{\in\mathcal{P}(\mathcal{Y})} L_{EOT}(\ )$.

8. Statistical interpretations of the R-D problem

The R-D problem (4), and its equivalent EOT "projection" problem (9), admit a statistical interpretation as solving a particular maximum likelihood estimation (MLE) problem. The connection between R-D and model estimation has been observed in the information theory and compression literature (Harrison and Kontoyiannis, 2008; Ballé et al., 2017; Theis et al., 2017; Yang and Mandt, 2022), and Rigollet and Weed (2018) noted the connection between the EOT problem (9) and maximum-likelihood deconvolution (Carroll and Hall, 1988). Here we provide a unifying account.

Putting on our hats as statisticians, the goal is to fit a density model to the true distribution based on samples. To specify the model, we start with a "prior" distribution belonging to some family $\in \mathcal{P}(\mathcal{Y})$ and a conditional density $p(x|y)$ on $\mathcal{X}$, and define the model by a marginal likelihood,

$$p(x) := \int_{\mathcal{Y}} p(x|y)\ (dy). \tag{18}$$

To fit p to the data distribution, we ideally maximize the population log likelihood,

$$\max_{\in\mathcal{H}} \int \log p(x)\ (dx); \tag{19}$$

or in practice, maximize the sample log-likelihood by replacing with its empirical measure ^{m}.

To connect the MLE problem to R-D estimation, suppose $p(x|y)$ arises from a distortion such that $p(x|y) \propto e^{-(x,y)}$, where the normalization constant does not depend on y. A common example is a Gaussian density with a fixed variance $^2 = \frac{1}{}$, corresponding to a squared error distortion $(x, y) = \frac{1}{2}\|x - y\|^2$. Then the negative of the population log likelihood equals the rate function (7), up to a constant. The setting where $\mathcal{H} = \mathcal{P}(\mathcal{Y})$, also known as non-parametric MLE (Kiefer and Wolfowitz, 1956), is equivalent to the R-D problem (4), and the best achievable population log-likelihood corresponds to an intercept to the R-D curve of. More generally, given any parametric family $\mathcal{H}$, the best achievable population (or, sample) log-likelihood (19) corresponds to the parametric R-D estimator $E^{\mathcal{H}}(\ )$ (or, $F^{\mathcal{H}}(\ ^m)$) introduced in Sec. 2.1.

Sometimes the data is known to originate from a "clean" distribution $\mathcal{Y}$, and the conditional density $p(x|y)$ corresponds to observation or measurement noise with known characteristics. In this case the estimation problem aims to recover $\mathcal{Y}$ from the noisy measurements, and is known as maximum-likelihood deconvolution (Carroll and Hall, 1988). Alternatively, (18) may be seen as a modeling choice corresponding to a latent variable model, i.e. a prior distribution over an unobserved latent variable. As a simple example, a Gaussian mixture model with weights $w_{1:k}$ and component locations $_{1;\dots;k}$ can be specified by a discrete latent variable with distribution $= \sum_{k=1}^{K} w_k\ _k$, and a conditional Gaussian density $p(x|y) = \mathcal{N}(y;\ ^2)$. Latent variable models have gained prominence in deep generative modeling with examples including VAEs (Kingma and Welling, 2013) and diffusion probablistic models (Sohl-Dickstein et al., 2015). These models are often trained by maximizing the evidence lower bound (Blei et al., 2017), which shares the variational formulation of the rate function (6) in R-D estimation, and even the BA algorithm itself has an interesting correspondence to the EM algorithm, which we explain below.

9. R-D estimation and variational inference/learning

In this section, we give a more detailed explanation of how the the R-D problem (3) relates to variational inference and learning in latent variable models.

To facilitate the discussion and make clearer the connections, we adopt notation more common in statistics and information theory. Table 1 summarizes the notation and the correspondence to the measure-theoretic notation used in the main text.

In statistical modeling, the goal is to fit a density $\hat{p}(x)$ to the true (unknown) data distribution P_X. Consider specifying $\hat{p}(x)$ as a latent variable model, where Y takes on the role of a latent space, and $Q_Y = \Theta$ is the distribution of a latent variable Y (which may encapsulate the model parameters). As we shall see, the optimization objective defining the rate functional (6) corresponds to an aggregate Evidence LOwer Bound (ELBO) (Blei et al., 2017). Thus, computing the rate functional corresponds to variational inference (Blei et al., 2017) in a given model (see Sec. 9.2), and the parametric R-D estimation problem, i.e.,

$$\inf_{\Theta \in \mathcal{H}} L_{BA}(\Theta);$$

is equivalent to estimating a model using the variational EM algorithm (Beal and Ghahramani, 2003) (see Sec. 9.3). The variational EM algorithm can be seen as a restricted version of the BA algorithm (see Sec. 9.3), whereas the EM algorithm (Dempster et al., 1977) shares its E-step with the BA algorithm but can differ in its M-step (see Sec. 9.4).

9.1. Setup

For concreteness, fix a reference measure ν on Y, and suppose Q_Y has density $q(y)$ w.r.t. ν. Often the latent space Y is a Euclidean space, and $\nu(dy)$ is the usual probability density function w.r.t. the Lebesgue measure, or when the latent space is discrete/countable, ν is the counting measure and $q(y)$ is the usual probability mass function. We will consider the typical parametric estimation problem and choose a particular parametric form for Q_Y indexed by a parameter vector θ. This defines a parametric family $\mathcal{H} = \{Q_Y : \theta \in \Theta\}$ for some parameter space Θ. Finally, suppose the distortion function induces a conditional likelihood density $p_\theta(x|y) \propto e^{-\beta(x;y)}$, with a normalization constant that has no y-dependence.

A latent variable model is then specified by the joint density $q(y)p(x|y)$. We use it to posit a density for the data by

$$\hat{p}(x) = \int_Y p(x|y)\,dQ_Y(y) = \int_Y p(x|y)q(y)\,\nu(dy). \tag{20}$$

As a simple example, a Gaussian mixture model with isotropic component variances can be specified as follows. Let Q_Y be a mixing distribution on $X = Y = \mathbb{R}^d$ parameterized by component weights $w_{1,\dots,k}$ and locations $\theta_{1,\dots,k}$, such that $Q_Y = \sum_{k=1}^K w_k \delta_{\theta_k}$. Let $p(x|y) = \mathcal{N}(y, \sigma^2)$ be a conditional Gaussian density with mean y and variance σ^2. Now formula (20) gives the usual Gaussian mixture density on $\mathbb{R}^d$.

Maximum-likelihood estimation then ideally maximizes the population log (marginal) likelihood,

$$E_{x \sim P_X}[\log \hat{p}(x)] = \int \log \hat{p}(x) P_X(dx) = \int \log \int_Y p(x|y)\,dQ_Y(y)\; P_X(dx). \tag{21}$$

To deal with the often intractable marginal likelihood in the inner integral, we turn to variational inference and learning (Jordan et al., 1999; Wainwright et al., 2008).

9.2. Connection to variational inference

Given a latent variable model and any data observation x, a central task in Bayesian statistics is to infer the Bayesian posterior (Jordan, 1999), which we formally view as a conditional distribution $Q_{Y|X=x}$. It is given by

$$\frac{dQ_{Y|X=x}(y)}{dQ_Y(y)} = \frac{p(x|y)}{\hat{p}(x)};$$

or, using the density $q(y)$ of Q_Y, given by the following conditional density via the familiar Bayes' rule,

$$q_\theta(y|x) = \frac{p(x|y)q(y)}{\hat{p}(x)} = \frac{p(x|y)q(y)}{\int_Y p(x|y)q(y)\,\nu(dy)}.$$

Unfortunately, the true Bayesian posterior is typically intractable, as the (marginal) data likelihood in the denominator involves an often high-dimensional integral. Variational inference (Jordan et al., 1999; Wainwright et al., 2008) therefore aims to approximate the true posterior by a variational distribution $Q_{Y|X=x} \in \mathcal{P}(Y)$ by minimizing their relative divergence

$H(Q_{Y|X=x}|Q_{Y|X=x})$. The problem is equivalent to maximizing the following lower bound on the marginal log-likelihood, known as the Evidence Lower BOund (ELBO) (Blei et al., 2017):

$$\begin{aligned}\arg\min_{Q_{Y|X=x}} H(Q_{Y|X=x}|Q_{Y|X=x}) &= \arg\max_{Q_{Y|X=x}} \mathrm{ELBO}(Q_Y; x; Q_{Y|X=x}),\\ \mathrm{ELBO}(Q_Y; x; Q_{Y|X=x}) &= \mathbb{E}_{y\sim Q_{Y|X=x}}[\log p(x|y)] - H(Q_{Y|X=x}|Q_Y)\\ &= \log \hat{p}(x) - H(Q_{Y|X=x}|Q_{Y|X=x}). \end{aligned} \tag{22}$$

Translating the de nition of the rate functional (6) into the present scenario,

$$\begin{aligned} L_{BA}(Q_Y) &= \inf_{Q_{Y|X}} \mathbb{E}_{x\sim P_X, y\sim Q_{Y|X=x}}[-\log p(x|y)] + \mathbb{E}_{x\sim P_X}[H(Q_{Y|X=x}|Q_Y)] + \text{const}\\ &= \inf_{Q_{Y|X}} \mathbb{E}_{x\sim P_X}[-\mathrm{ELBO}(Q_Y; x; Q_{Y|X=x})] + \text{const}, \end{aligned} \tag{23}$$

we recognize that the rate functional optimizes the population ELBO, and this optimization problem decouples over x and can be solved by the variational inference problem (22) involving $Q_{Y|X=x}$. At optimality, $Q_{Y|X} = Q_{Y|X}$, the ELBO (22) is tight and recovers $\log \hat{p}(x)$, and the rate functional takes on the form of a (negated) population marginal log likelihood (21), as given earlier by (7) in Sec. 2.1.

9.3. Connection to variational EM

The discussion so far concerns probabilistic inference, where a latent variable model $(Q_Y, p(x|y))$ has been given and we saw that computing the rate functional amounts to variational inference. Suppose now we wish to learn a model from data. The R-D problem (4) then corresponds to model estimation using the variational EM algorithm (Beal and Ghahramani, 2003).

To estimate a latent variable model by (approximate) maximum-likelihood, the variational EM algorithm maximizes the population ELBO

$$\mathbb{E}_{x\sim P_X}[\mathrm{ELBO}(Q_Y; x; Q_{Y|X=x})] = \mathbb{E}_{x\sim P_X, y\sim Q_{Y|X=x}}[\log p(x|y)] - \mathbb{E}_{x\sim P_X}[H(Q_{Y|X=x}|Q_Y)], \tag{24}$$

w.r.t. Q_Y and $Q_{Y|X}$. This precisely corresponds to the R-D problem $\inf_{Q_Y \in \mathcal{H}} L_{BA}(Q_Y)$, using the form of $L_{BA}(Q_Y)$ from (23).

In popular implementations of variational EM such as the VAE (Kingma and Welling, 2013), Q_Y and $Q_{Y|X}$ are restricted to parametric families. When they are allowed to range over all of $\mathcal{P}(\mathcal{Y})$ and all conditional distributions, variational EM then becomes equivalent to the BA algorithm.

9.4. The Blahut–Arimoto and EM algorithms

The BA and EM algorithms share the same objective function, namely the negative of the population ELBO (24). Both also perform coordinate descent / alternating projection, but they de ne the coordinates slightly differently — the BA algorithm uses $(Q_{Y|X}, Q_Y)$ with $Q_Y \in \mathcal{P}(\mathcal{Y})$, whereas the EM algorithm uses $(Q_{Y|X}, \theta)$ with θ indexing a parametric family $\mathcal{H} = \{Q_Y : \theta \in \Theta\}$. Thus the coordinate update w.r.t. $Q_{Y|X}$ (the "E-step") is the same in both algorithms, but the subseuquent "M-step" potentially differs depending on the role of θ.

Given the optimization objective, which is simply the negative of (24),

$$\mathbb{E}_{x\sim P_X, y\sim Q_{Y|X=x}}[-\log p(x|y)] + H(P_X Q_{Y|X} | P_X \otimes Q_Y), \tag{25}$$

both the BA and EM algorithms optimize the transition kernel $Q_{Y|X}$ the same way in the E-step, as

$$\frac{dQ_{Y|X=x}}{dQ_Y}(y) = \frac{p(x|y)}{\hat{p}(x)}. \tag{26}$$

For the M-step, the BA algorithm only minimizes the relative entropy term of the objective (25),

$$\min_{Q_Y \in \mathcal{P}(\mathcal{Y})} H(P_X Q_{Y|X} | P_X \otimes Q_Y),$$

(with the optimal Q_Y given by the second marginal of $P_X Q_{Y|X}$) whereas the EM algorithm minimizes the full objective w.r.t. the parameters of Q_Y,

$$\min_{Q_Y} E_{(x,y) \sim P_X Q_{Y|X}}[-\log p(x|y)] + H(P_X Q_{Y|X}; P_X Q_Y). \tag{27}$$

The difference comes from the fact that when we parameterize Q_Y by θ in the parameter estimation problem, $Q_{Y|X}$ — and consequently both terms in the objective (27) — will have functional dependence on θ through the E-step optimality condition (26).

In the Gaussian mixture example, $Q_Y = \sum_{k=1}^K w_k \delta_{\theta_k}$, and its parameters consist of the components weights $(w_1, \dots, w_K) \in \Delta^{d-1}$ and location vectors $\{\theta_1, \dots, \theta_K\} \subset \mathbb{R}^d$. The E-step computes $Q_{Y|X=x} = \sum_k w_k \frac{p(x|\theta_k)}{p(x)} \delta_{\theta_k}$. For the M-step, if we regard the locations as known so that $\theta = (w_1, \dots, w_K)$ only consists of the weights, then the two algorithms perform the same update; however if θ also includes the locations, then the M-step of the EM algorithm will not only update the weights as in the BA algorithm, but also the locations, due to the distortion term $E_{(x,y) \sim P_X Q_{Y|X}}[-\log p(x|y)] = \sum_k w_k \frac{p(x|\theta_k)}{p(x)} \int -\log p(x|\theta_k) P_X(dx)$.

10. Wasserstein gradient descent

10.1. Proposed algorithm

Algorithm 2 Wasserstein gradient descent

Inputs: Loss function $L \in \{L_{BA}, L_{EOT}\}$; data distribution $\in \mathcal{P}(\mathbb{R}^d)$; initial measure $\mu_0 \in \mathcal{P}(\mathbb{R}^d)$; total number of iterations N; step sizes $\tau_1, \dots, \tau_N$; batch size $m \in \mathbb{N}$.
for $t = 1, \dots, N$ **do**
  if support of P contains more than m points **then**
    $\hat{P}_m \leftarrow \frac{1}{m} \sum_{i=1}^m \delta_{x_i}$ for $x_1, \dots, x_m$ independent samples from P
    Wasserstein gradient of $L(\hat{P}_m, \cdot)$ at μ_{t-1} {see Definition 10.1}
  else
    Wasserstein gradient of $L(P, \cdot)$ at μ_{t-1} {see Definition 10.1}
  end if
  $\mu_t \leftarrow (\mathrm{id} - \tau_t \nabla \Phi_t)_{\#} \mu_{t-1}$ {"$\#$" denotes pushforward}
end for
Return: μ_N

There are two equivalent ways to introduce the Wasserstein gradient (Ambrosio et al., 2008). We start with the constructive one, which forms the computational basis of our algorithm.

Definition 10.1. For a functional $L : \mathcal{P}(Y) \to \mathbb{R}$ and $\in \mathcal{P}(Y)$, we say that $\nabla_L(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is a first variation of L at $$ if

$$\lim_{\varepsilon \to 0} \frac{L((1-\varepsilon) + \varepsilon\sigma) - L()}{\varepsilon} = \int \nabla_L(\cdot)\, d(\sigma -) \quad \text{for all } \sigma \in \mathcal{P}(Y).$$

We call its (Euclidean) gradient $\nabla \nabla_L(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$, if it exists, the Wasserstein gradient of L at $$.

For $L = L_{EOT}$, the first variation is given by the Kantorovich potential, which is the solution of the convex dual of L_{EOT} and commonly computed by Sinkhorn's algorithm (Peyré and Cuturi, 2019; Nutz, 2021). Specifically, let $(\cdot)$ be potentials for $L_{EOT}(\cdot, \cdot)$. Then $\nabla_L(\cdot) = $ is the first variation w.r.t. $$, and hence $\nabla = $ is the Wasserstein gradient. This gradient is known to exist whenever $$ is differentiable and the marginals are sufficiently light-tailed. For $L = L_{BA}$, the first variation can be computed explicitly. As we show below, the first variation is

$$(y) = \int_{\mathbb{R}} \frac{\exp(\quad (x,y))}{\exp(\quad (x,y))\, (dy)} \, (dx)$$

and then the Wasserstein gradient is $\nabla_{BA}(\cdot) = r$. We observe that (y) is computationally cheap; it corresponds to running a single iteration of Sinkhorn's algorithm. By contract, finding the potential for L_{EOT} requires running Sinkhorn's algorithm to convergence.

The second, more abstract possibility is to postulate the linearization property of the gradient, which will allow us to prove convergence of our gradient descent scheme (Prop. 10.2). Following (Carlier et al., 2022), we state this as follows: for any $\sim 2 P(Y)$ and $2 (; \sim)$,

$$L(\sim) \quad L( ) = \int (y \quad x)^{>} r \; V_L( )(x) \quad (dx; dy) + o \int ky \quad xk^2 \quad (dx; dy) \quad ; \tag{28}$$
$$\int kr \; V_L( )k^2 \quad kr \quad V_L(\sim)k^2 \, d \quad CW_2(; \sim):$$

The first line of (28) is proved in (Carlier et al., 2022, Proposition 4.2) in case X and Y are compact and $$ is twice continuously differentiable. The second line of (28) follows using $a^2 \quad b^2 = (a + b)(a \quad b)$ and a combination of boundedness and Lipschitz continuity of V_L, see (Carlier et al., 2022, Proposition 2.2 and Corollary 2.4).

Under suitable regularity conditions, (28) is in fact equivalent to the Wasserstein gradient de nition 10.1. Moreover, it allows us to show that Wasserstein gradient descent for L_{EOT} and L_{BA} converges to a stationary point in Proposition 10.2.

10.2. Wasserstein gradient for the rate functional

Below we calculate the Wasserstein gradient of $L_{BA}( ) = \int \log \int \exp(\quad (x; y)) \; (dy) \; (dx)$. Under suf cient integrability on $$ and $$ to exchange the order of limit and integral, we can calculate the first variation as

$$\begin{aligned}
\lim_{" ! \; 0} \frac{L((1 \quad ") \; + "\sim) \quad L( )}{"} &= \int \lim_{" ! \; 0} \frac{1}{"} \log \frac{\int \exp(\quad (x; y))(\quad + "(\sim \quad))(dy)}{\int \exp(\quad (x; y)) \; (dy)} \quad (dx) \\
&= \int \lim_{" ! \; 0} \frac{1}{"} \log \; 1 + \frac{\int \exp(\quad (x; y))"(\sim \quad)(dy)}{\int \exp(\quad (x; y)) \; (dy)} \quad (dx) \\
&= \int\int \frac{\exp(\quad (x; y))}{\int \exp(\quad (x; y)) \; (dy)} \; (dx)(\sim \quad)(dy);
\end{aligned}$$

where the last equality uses $\lim_{" ! \; 0} \frac{1}{"} \log(1 + "x) = x$ and Fubini's theorem. Thus the first variation of L_{BA} at $$ is

$$(y) = \int \frac{\exp(\quad (x; y))}{\int \exp(\quad (x; y)) \; (dy)} \; (dx): \tag{29}$$

To find the desired Wasserstein gradient of L_{BA}, it remains to take the Euclidean gradient of, i.e., $rL_{BA}( ) = r$.

10.3. Convergence of Wasserstein gradient descent

Here we show that Wasserstein gradient descent for L_{EOT} and L_{BA} converges to a stationary point under mild conditions.

Proposition 10.2 (Convergence of Wasserstein gradient descent)**.** Let $_1 \quad _2 \quad 0$ satisfy $\sum_{k=1}^{1} _k = 1$ and $\sum_{k=1}^{1} {}_k^2 < 1$. Let $L : P(\mathbb{R}^d) ! \; \mathbb{R}$ be Wasserstein differentiable in the sense that (28) holds. Denoting by ^{t} the steps in Algorithm 2, assume that $L(^0)$ is finite and $\int kr \; V_L(^t)k^2 \, d \; ^t$ is bounded. Then

$$\lim_{t!1} \int kr \; V_L(^t)k^2 \, d \; ^t = 0:$$

Before proving this proposition, we first provide an auxiliary result.

Lemma 10.3. Let $_1 \quad _2 \quad 0$ and $a_t \quad 0, t \; 2 \; \mathbb{N}, C > 0$ satisfy $\sum_{t=1}^{1} \; _t = 1$, $\sum_{t=1}^{1} \; {}_t^2 < 1$, $\sum_{t=1}^{1} a_t \; _t < 1$ and $ja_t \quad a_{t+1}j \quad C \; _t$ for all $t \; 2 \; \mathbb{N}$. Then $\lim_{t!1} \; a_t = 0$.

Proof. The conclusion remains unchanged when rescaling a_t by the constant C, and thus without loss of generality $C = 1$.

Clearly $_t \; ! \quad 0$ as $\sum_{t=1}^{1} {}_t^2 < 1$. Moreover, there exists a subsequence $(a_{t_k})_{t_k 2 \mathbb{N}}$ which converges to zero (otherwise there exists $> \; 0$ such that $a_t \quad > \; 0$ for all but finitely many t, contradicting $\sum_{t=1}^{1} \; _t a_t < 1$).

Arguing by contradiction, suppose that the conclusion fails, i.e., that there exists a subsequence $(a_t)_{t2\mathbb{N}}$ of which is uniformly bounded away from zero, say $a_t \quad > \; 0$ along that subsequence. Using this subsequence and the convergent subsequence

mentioned above, we can construct a subsequence $a_{i_1}, a_{i_2}, a_{i_3}, \ldots$ where $a_{i_n} \geq 0$ for n odd and $a_{i_n} \leq 0$ for n even. We will show that

$$\sum_{t=i_{2n-1}}^{i_{2n}} a_t \, \alpha_t \, \& \, \varepsilon^2/2 \quad \text{for all } n \geq 2 \in \mathbb{N};$$

contradicting the finiteness of $\sum_t \alpha_t a_t$. (The notation $\&$ indicates (in)equality up to additive terms converging to zero for $n \to \infty$.)

To ease notation, fix n and set $m = i_{2n-1}$ and $M = i_{2n}$. We show that $\sum_{t=m}^{M} a_t \, \alpha_t \, \& \, \varepsilon^2/2$. To this end, using $|a_t - a_{t+1}| \leq \alpha_t$ we find

$$a_t \geq a_M - \sum_{j=k}^{M-1} \alpha_j \geq -\sum_{j=k}^{M-1} \alpha_j.$$

Since $a_m \geq 0$, there exists a largest $n_0 \geq m$, such that $\sum_{j=n_0}^{M-1} \alpha_j \geq \varepsilon$ (and thus $\sum_{j=n_0}^{M-1} \alpha_j \geq \varepsilon$ as well). We conclude

$$\begin{aligned}
\sum_{t=m}^{M} \alpha_t a_t &\geq \sum_{t=n_0}^{M} \alpha_t a_t \geq \sum_{t=n_0}^{M} \alpha_t \left(-\sum_{j=k}^{M-1} \alpha_j \right) \geq -\varepsilon^2 - \sum_{t=n_0}^{M} \sum_{j=n_0}^{M} \alpha_t \alpha_j \mathbf{1}_{\{j \geq k\}} \\
&\geq -\varepsilon^2 - \left(\sum_{t=n_0}^{M} \alpha_t \right)^2 \geq -\varepsilon^2 - \frac{1}{2} \sum_{t=n_0}^{M} \alpha_t^2 \geq -\varepsilon^2/2,
\end{aligned}$$

where we used that $\sum_{t=n_0}^{M} \alpha_t^2 \leq 0$. This completes the proof. □

Proof of Proposition 10.2. Using the linear approximation property in (28), we calculate

$$\begin{aligned}
L(\theta^{(n)}) - L(\theta^{(0)}) &= \sum_{t=0}^{n-1} L(\theta^{(t+1)}) - L(\theta^{(t)}) \\
&= \sum_{t=0}^{n-1} \alpha_t \int \|\nabla V_L(\theta^{(t)})\|^2 \, d\mu^{(t)} + \alpha_t^2 o\left(\int \|\nabla V_L(\theta^{(t)})\|^2 \, d\mu^{(t)} \right).
\end{aligned}$$

As $L(\theta^{(0)})$ is finite and $L(\theta^{(n)})$ is bounded from below, it follows that

$$\sum_{t=0}^{\infty} \alpha_t \int \|\nabla V_L(\theta^{(t)})\|^2 \, d\mu^{(t)} < \infty.$$

The claim now follow by applying Lemma 10.3 with $a_t = \int \|\nabla V_L(\theta^{(t)})\|^2 \, d\mu^{(t)}$; note that the assumption in the lemma is satisfied due to the second inequality in (28). □

10.4. Sample complexity

Let $X = Y = \mathbb{R}^d$ and $\ell(x, y) = \|x - y\|^2$. Using the fact that the R-D problem (4) and EOT projection problem (9) share the same optimizers (see Sec. 2.2), we leverage a result from the OT literature (Mena and Niles-Weed, 2019) to prove finite-sample bounds on the optimal solution quality of WGD. This also sharpens known asymptotic consistency results on the empirical R-D estimators (Harrison and Kontoyiannis, 2008), and quantifies their finite-sample behavior.

Denote by $\mathcal{P}_n(\mathbb{R}^d)$ the set of probability measures on $\mathbb{R}^d$ which are supported on at most n points.

Proposition 10.4. Let ν be σ^2-subgaussian. Then every optimizer of (4) and (9) is also σ^2-subgaussian. Consider

$L := L_{EOT}$. For a constant C_d only depending on d, we have

$$\min_{2P(R^d)} L(\,;\,) \quad \min_{n} {}_{2P_n(R^d)} L(\,;\,{}_n) \quad C_d\left(1 + \frac{d5d{=}2e{+}6}{d5d{=}4e{+}3}\right)\frac{1}{\sqrt{n}};$$

$$E\left[\min_{2P(R^d)} L(\,;\,) \quad \min_{2P(R^d)} L({}^m;\,)\right] \quad C_d\left(1 + \frac{d5d{=}2e{+}6}{d5d{=}4e{+}3}\right)\frac{1}{\sqrt{m}};$$

$$E\left[\min_{2P(R^d)} L(\,;\,) \quad \min_{n} {}_{2P_n(R^d)} L({}^m;\,{}_n)\right] \quad C_d\left(1 + \frac{d5d{=}2e{+}6}{d5d{=}4e{+}3}\right)\left(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{n}}\right);$$

for all $n; m \in \mathbb{N}$, where ^{m} is the empirical measure of $$ with m independent samples and the expectation $E[\cdot]$ is over these samples. The same inequalities hold for $L = {}^{-1}L_{BA}$, with the identification $ = {}^{-1}$.

For the proof, we will need the following lemma which is of independent interest. We write $\le_c$ if $$ is dominated by $$ in convex order, i.e., $\int f\,d \le \int f\,d$ for all convex functions $f : R^d \to R$.

Lemma 10.5. Let $$ have finite second moment. Given $\in P(R^d)$, there exists $\tilde{} \in P(R^d)$ with $\tilde{} \le_c$ and

$$L_{EOT}(\,;\,\tilde{}) \le L_{EOT}(\,;\,):$$

This inequality is strict if $\neq_c$. In particular, any optimizer $$ of (9) satisfies $\le_c$.

Proof. Because this proof uses disintegration over Y, it is convenient to reverse the order of the spaces in the notation and write a generic point as $(x; y) \in Y \times X$. Consider $\in (\,;\,)$ and its disintegration $= (dx)\,K(x; dy)$ over $x \in Y$. Define $T : R^d \to R^d$ by

$$T(x) := \int y\,K(x; dy):$$

Define also $\tilde{} := (T; \mathrm{id})_{\#}$ and $\tilde{} := \tilde{}_1$. From the definition of T, we see that $$ is a martingale, thus $\tilde{} \le_c$. Moreover, $\tilde{} = (T; \mathrm{id})_{\#}$. The data-processing inequality now shows that

$$H(\tilde{}\,|\,\tilde{}) \le H(\,|\,):$$

On the other hand, $\int\int \|y - K(x; dy)\|\,(dx)$ $\int\int \|y\|^2 K(x; dy)\,(dx) - \int \|x\|^2\,(dx)$ $\int\int \|y\|^2 K(x; dy)$ since the barycenter minimizes the squared distance, and this inequality is strict whenever $\neq$ $y = K(x; dy)$. Thus

$$\int \|x\|^2\,\tilde{}(dx; dy) \quad \int \|y\|^2\,\tilde{}(dx; dy) = \int \|x\|^2\,(dx; dy) \quad \int \|y\|^2\,(dx; dy);$$

and the inequality is strict unless $T(x) = x$ for $$-a.e. x, which in turn is equivalent to $$ being a martingale. The claims follow. □

Proof of Proposition 10.4. Subgaussianity of the optimizer follows directly from Lemma 10.5.

Recalling that $\inf L_{EOT}(\,)$ and $\inf {}^{-1}L_{BA}(\,)$ have the same values and minimizers, it suffices to show the claim for $L = L_{EOT}$. Let $$ be an optimizer of (9) (i.e., an optimal reproduction distribution) and $\hat{}$ its empirical measure from n samples, then clearly

$$\min_{n} {}_{2P_n(R^d)} L_{EOT}(\,;\,{}_n) \quad \min_{2P(R^d)} L_{EOT}(\,;\,) = \min_{n} {}_{2P_n(R^d)} L_{EOT}(\,;\,{}_n) \quad \min_{2P(R^d)} L_{EOT}(\,;\,)$$
$$E[\,|L_{EOT}(\,;\,{}_n) \quad L_{EOT}(\,;\,)|]$$

where the expectation is taken over samples for $\hat{}$. The first inequality of Proposition 10.4 now follows from the sample complexity result for entropic optimal transport in (Mena and Niles-Weed, 2019, Theorem 2).

Denote by ${}_m$ the optimizer for the problem (9) with $$ replaced by ^{m}. Similarly to the above, we obtain

$$E\left[\min_{2P(R^d)} L_{EOT}(\,;\,) \quad \min_{2P(R^d)} L_{EOT}({}^m;\,)\right]$$
$$E\left[\max_{2\{\,;\,{}_m\}} |L_{EOT}(\,;\,) \quad L_{EOT}({}^m;\,)|\right];$$

where the expectation is taken over samples from ν^m. In this situation, we cannot directly apply (Mena and Niles-Weed, 2019, Theorem 2). However, the bound given by (Mena and Niles-Weed, 2019, Proposition 2) still applies, and the only dependence on $2 \{ 1, \ldots, m \}$ is through their subgaussianity constants. By Lemma 10.5, these constants are bounded by the corresponding constants of ν and ν^m. Thus, the arguments in the proof of (Mena and Niles-Weed, 2019, Theorem 2) can be applied, yielding the second inequality of Proposition 10.4.

The final inequality of Proposition 10.4 follows from the first two inequalities (the first one being applied with ν^m) and the triangle inequality, where we again use the arguments in the proof of (Mena and Niles-Weed, 2019, Theorem 2) to bound the expectation over the subgaussianity constants of ν^m. □

10.5. Estimation of rate and distortion

Here, we describe our estimator for an upper bound (R, D) of $R(D)$, after solving the Lagrangian problem (3).

For any given pair of ν and K, we always have that $D := \int d(x, y) \, K(dy|x) \, \nu(dx)$ and $R := H(\nu K | \nu) = \int \log \frac{dK(x, \cdot)}{d\nu}(y) \, K(dy|x) \, \nu(dx)$ yield an upper bound of $R(D)$ (Berger, 1971). The two quantities can be estimated by simple Monte Carlo, provided we can sample from ν and evaluate the density $\frac{d\nu K}{d\nu}(x, y) = \frac{dK(x, \cdot)}{d\nu}(y)$.

When only ν is given, e.g., obtained from optimizing (4), we estimate an R-D upper bound as follows. Taking a hint from the BA algorithm, we define a kernel K similarly as in an update step of the BA algorithm, $\frac{dK(x, \cdot)}{d\nu}(y) = \frac{e^{-\beta d(x, y)}}{\int e^{-\beta d(x, y')} \nu(dy')}$; then we estimate (R, D) using the pair (ν, K) as described earlier.

For NERD, which uses a continuous ν, we follow (Lei et al., 2023) and use an n-sample empirical measure of ν to estimate (R, D). A limitation of NERD and our particle method is that they tend to converge to a rate estimate of about $\log(n)$, where n is the support size of ν. This is because as the algorithms approach the n-point minimizer ν_n of the R-D problem, the rate estimate R approaches the mutual information of K_{ν_n}, which is upper-bounded by $\log(n)$ (Eckstein and Nutz, 2022).

11. Example implementation of WGD

We provide a self-contained minimal implementation of Wasserstein gradient descent on L_{BA}, using the Jax library (Bradbury et al., 2018). To compute the Wasserstein gradient, we evaluate the first variation of the rate functional in `compute_psi_sum` according to (29), yielding $\psi_\beta(x_i)_{i=1}^n$, then simply take its gradient w.r.t. the particle locations $x_{1:n}$ using Jax's autodiff tool on line 51.

The implementation of WGD on L_{EOT} is similar, except the first variation is computed using Sinkhorn's algorithm. Both versions can be easily just-in-time compiled and enjoy GPU acceleration.

```
1  # Wasserstein GD on the rate functional L_{BA}.
2  import jax.numpy as jnp
3  import jax
4  from jax.scipy.special import logsumexp
5
6  # Define the distortion function \rho.
7  squared_diff = lambda x, y: jnp.sum((x - y) ** 2)
8  pairwise_distortion_fn = jax.vmap(jax.vmap(squared_diff, (None, 0)), (0, None))
9
10
11 def wgrad(mu_x, mu_w, nu_x, nu_w, rd_lambda):
12     """
13     Compute the Wasserstein gradient of the rate functional, which we will use
14     to move the \nu particles.
15     :param mu_x: locations of \mu atoms.
16     :param mu_w: weights of \mu atoms.
17     :param nu_x: locations of \nu atoms.
18     :param nu_w: weights of \nu atoms.
```

```
    :param rd_lambda: R-D tradeoff hyperparameter.
    :return:
    """

    def compute_psi_sum(nu_x):
        """
        Here we compute a surrogate loss based on the first variation \psi, which
        allows jax autodiff to compute the desired Wasserstein gradient.
        :param nu_x:
        :return: psi_sum = \sum_i \psi(nu_x[i])
        """
        C = pairwise_distortion_fn(mu_x, nu_x)
        scaled_C = rd_lambda * C  # [m, n]
        log_nu_w = jnp.log(nu_w)  # [1, n]

        # Solve BA inner problem with a fixed nu.
        phi = - logsumexp(-scaled_C + log_nu_w, axis=1, keepdims=True)  # [m, 1]
        loss = jnp.sum(mu_w * phi)  # Evaluate the rate functional. Eq (6) in paper.

        # Let's also report rate and distortion estimates (discussed in Sec. 4.4 of the paper).
        # Find \pi^* via \phi
        pi = jnp.exp(phi - scaled_C) * jnp.outer(mu_w, nu_w)  # [m, n]
        distortion = jnp.sum(pi * C)
        rate = loss - rd_lambda * distortion

        # Now evaluate \psi on the atoms of \nu.
        phi = jax.lax.stop_gradient(phi)
        psi = - jnp.sum(jnp.exp(jax.lax.stop_gradient(phi) - scaled_C) * mu_w, axis=0)
        psi_sum = jnp.sum(psi)  # For computing gradient w.r.t. each nu_x atom.
        return psi_sum, (loss, rate, distortion)

    # Evaluate the Wasserstein gradient, i.e., \nabla \psi, on nu_x.
    psi_prime, loss = jax.grad(compute_psi_sum, has_aux=True)(nu_x)
    return psi_prime, loss


def wgd(X, n, rd_lambda, num_steps, lr, rng):
    """
    A basic demo of Wasserstein gradient descent on a discrete distribution.
    :param X: a 2D array [N, d] of data points defining the source \mu.
    :param n: the number of particles to use for \nu.
    :param rd_lambda: R-D tradeoff hyperparameter.
    :param num_steps: total number of gradient updates.
    :param lr: step size.
    :param rng: jax random key.
    :return: (nu_x, nu_w), the locations and weights of the final \nu.
    """
    # Set up the source measure \mu.
    m = jnp.size(X, 0)
    mu_x = X
    mu_w = 1 / m * jnp.ones((m, 1))
    # Initialize \nu atoms using random training samples.
    rand_idx = jax.random.permutation(rng, m)[:n]
    nu_x = X[rand_idx]  # Locations of \nu atoms.
```

```
    nu_w = 1 / n * jnp.ones((1, n))          # Uniform weights.
    for step in range(num_steps):
        psi_prime, (loss, rate, distortion) = wgrad(mu_x, mu_w, nu_x, nu_w, rd_lambda)
        nu_x -= lr * psi_prime
        print(f"step={step}, loss={loss:.4g}, rate={rate:.4g}, distortion={distortion:.4g}")

    return nu_x, nu_w


if __name__ == "__main__":
    # Run a toy example on 2D Gaussian samples.
    rng = jax.random.PRNGKey(0)
    X = jax.random.normal(rng, [10, 2])
    nu_x, nu_w = wgd(X, n=4, rd_lambda=2., num_steps=100, lr=0.1, rng=rng)
```

12. Further experimental results

12.1. Implementation

We implemented our algorithm and NERD in Jax, and borrowed the code for RD-VAE from (Yang and Mandt, 2022). We experimented with WGD for both L_{BA} and L_{EOT}. Empirically we found them to give similar results, while the former to be 10 to 100 times faster computationally; we therefore focus on WGD for L_{BA} in the discussions below.

For the RD-VAE, we used a similar architecture as the one used on the banana-shaped source in (Yang and Mandt, 2022), consisting of two-layer MLPs for the encoder and decoder networks, and Masked Autoregressive Flow (Papamakarios et al., 2017) for the variational prior. For NERD, we follow similar architecture settings as (Lei et al., 2023), using a two-layer MLP for the decoder network.

In most experiments, we use the Adam (Kingma and Ba, 2015) optimizer for updating the particle locations in WGD and for updating the variational parameters in other methods. For our hybrid WGD algorithm, which adjusts the particle weights in addition to their locations, we found that applying momentum to the particle locations can in fact slow down convergence, and therefore use plain gradient descent with a decaying step size.

Our deconvolution experiments were run on Intel(R) Xeon(R) CPUs, while the rest of the experiments were run on Titan RTX GPUs.

12.2. Deconvolution

Setup To understand the behavior and limitations of the various methods, we experiment with a setting where the sample size is very large. We consider the maximum-likelihood deconvolution problem described in Sec. 2.3. Here, the source distribution is the convolution of the uniform measure on the unit circle and a Gaussian noise $N(0; \sigma^2 I)$ with variance $\sigma^2 = 0.1$. Under a scaled squared distortion $d(x, y) = \frac{1}{2}\|x - y\|^2$, the R-D problem (3) becomes analytically tractable, and we find that $= S * N(0; \sigma^2 \frac{1}{-})$ whenever $\frac{1}{-} \geq \sigma^2$. We set $= \sigma^2 = 10$ for the MLE deconvolution problem, so $= S$ is the uniform measure on the circle. We use $m = 100000$ training samples, so that $\hat{m}$ and $F(\hat{m}) \approx F(\ ) =: OPT$, the latter of which we compute numerically. We can then assess the solution quality of an algorithm by how well its estimate of $F(\hat{m})$ agrees with the true OPT.

Loss curves and solutions from various methods. In Figure 2 we plot both the training and test losses for the various methods. The test losses are evaluated on freshly drawn samples from the source distribution, and provide estimates of the true population losses. As expected, the train losses appear similar to the test losses since we use a large sample size for training.

In Figure 3, we visualize the fitted measure after performing the optimization illustrated in Figure 2. We plot the location of the $n = 20$ particles from the BA, WGD, and hybrid algorithms, additionally coloring the particles from BA and the hybrid algorithm by their weights. To visualize the (continuous) learned by RD-VAE and NERD, we plot a scatter of 300 random samples drawn from each.

Figure 2. Left: The objective functions of the various methods across training iterations. Right: The same objective functions evaluated on random empirical measures of the source. The curve for each method is averaged over 5 reruns with different random seeds, with the shading corresponding to one standard deviation. The proposed WGD algorithms (orange, green) converge quickly to the theoretically optimal value OPT (cyan). BA (Blahut, 1972; Arimoto, 1972) (blue) converges quickly to a highly suboptimal solution, while the RD-VAE (Yang and Mandt, 2022) (red) converges more slowly, also to an inferior solution. NERD (Lei et al., 2023) (purple) fails to converge due to inaccuracy of its Monte-Carlo estimator when m is relatively small (see discussion in Sec. 3), leading to oscillating objective values.

Characterizing the optimal solution. In the deconvolution problem, $= S + N(0; \sigma^2 I)$, and whenever $\beta \leq \frac{1}{\sigma^2}$ the optimal solution to the R-D problem (3) is given by $= S + N(0; \sigma^2 - \frac{1}{\beta})$ and $K(x; dy) = N(x; \frac{1}{\beta})$. This follows from a basic property of the Gaussian distribution and an argument based on characteristic functions.

Knowing the optimal , we can therefore numerically compute the optimal loss,

$$OPT := L_{BA}( ) = \int_X \log \int_Y e^{(x;y)} (dy) (dx); \tag{30}$$

using the plug-in Monte Carlo estimator

$$\frac{1}{m}\sum_{i=1}^{m} \log\left(\frac{1}{n}\sum_{j=1}^{n} e^{(x_i;y_j)}\right); \tag{31}$$

where $x_{1;\dots;m}$ are drawn from and $y_{1;\dots;n}$ from . To reduce the bias of this estimator (also discussed in the context of NERD in Sec. 3), we use $m = 10000$ and the very large $n = 10^6$ in our Monte-Carlo estimation above.

Similarly, we can sample $\{(x_i; y_i)\}_{i=1}^{m}$ from K to compute the ground truth distortion and rate with high accuracy as follows,

$$D = \frac{1}{m}\sum_{i=1}^{m} (x_i; y_i);$$
$$R = OPT \quad D:$$

We can thus obtain the segment of the ground truth $R(D)$ where $\beta \leq \frac{1}{\sigma^2}$.

Scaling behavior of WGD. We rerun the various algorithms for $2 \{10; 30; 100\}$, and vary n for BA, WGD, and NERD, which determines their computational complexity. In Fig. 4 Left, we plot the relative difference between the estimated $F(m)$ and the true optimum in each case, and also include the R-D VAE result for reference. We observe that as increases (corresponding to lower entropic regularization), the problem becomes more dif cult and an increasingly accurate placement of the articles is required to achieve relatively low loss, demanding an increasingly large . Notably, the proposed WGD methods consistently dominate BA and NERD in its solution quality, translating to signi cantly fewer particles required for a given solution accuracy, especially in the more dif cult regime.

Figure 3. Visualizing the optimized measures from various algorithms. Left: $\varepsilon = 10 = \frac{1}{2}$, the same as in Fig. 2. Here ν is precisely the uniform distribution on the unit circle, colored in cyan. The proposed WGD algorithm places almost all atoms (green crosses) exactly on the circle. The proposed hybrid algorithm occasionally places atoms off the circle, and assigns them lower weights (orange stars) than the ones on the circle (red stars). This extra flexibility explains its faster convergence compared to the plain WGD algorithm seen in Fig. 2, while achieving the same optimized loss close to SOT. The BA algorithm is stuck with the randomly initialized set of atoms (blue) and can only manage to assign higher weights to atoms closer to the unit circle. RD-VAE and NERD have difficulty learning the true ν, as seen from the misplaced samples of ν from the two methods (faint red dots for RD-VAE and purple squares for NERD, respectively). Right: We repeat the experiment but with $\varepsilon = 2$. ν is now uniform on a circle with a smaller radius. The algorithms maintain their respective behavior from the $\varepsilon = 10$ case, with the BA, RD-VAE, and NERD algorithms failing to recover the support of ν. As $\varepsilon \to 0$, μ shrinks towards the mean of ν (the origin in this case), making it exceedingly difficult for the BA algorithm with a randomly discretized Y-space to locate the true support of ν.

Figure 4.Left: Relative error of the converged losses of various methods, compared to the ground truth (the lower the better; the curve for WGD coincides with that of the hybrid algorithm). The solution quality of the proposed WGD methods scales much more favorably than alternative methods in the number of particles, especially in the more difficult regime of large n. Right: Empirically validating the agreement between the scaling behavior of the optimality gap for WGD in the number of particles and our theoretical prediction.

Finally, we illustrate the scaling of the suboptimality gap for WGD as a function of n, as compared to our theoretical prediction of $O(\frac{1}{\sqrt{n}})$ in Proposition 10.4. We observe that the suboptimality gap appears to decrease at an even faster rate than $O(\frac{1}{\sqrt{n}})$, suggesting a conservative theoretical bound.

R-D upper bounds. We rerun the various algorithms with $\beta \in \{1, 3, 10, 30, 100, 300\}$ to produce upper bounds on $R(D)$, and plot the results in Figure 5-Right. We observe that WGD gives the tightest upper bound out of all the methods (the hybrid WGD algorithm produces overlapping curves and is omitted for clarity). As we increase n to 500 and 1000 (Figure 5-Middle, Left), the various methods increase linearly in their computational complexity (except for RD-VAE, which used a fixed architecture and didn't benefit noticeably from further increase in its neural network sizes), and eventually give qualitatively similar R-D upper bounds that generally agree with the true $R(D)$. Note that in large scale problems (e.g., those considered in Sec. 5.2), we are much more likely to operate in the "small n" regime due to computational constraints.

12.3. Higher-dimensional data

To demonstrate the scalability and accuracy of our proposed algorithm, we estimate the R-D functions of higher dimensional physics and speech sources considered in (Yang and Mandt, 2022).

For comparison, we obtained the results for the RD-VAE baseline, a neural compression method (Ballé et al., 2021), and a neural network-based R-D lower bound from (Yang and Mandt, 2022). As the datasets contain 10^5 to 10^6 data points, it becomes computationally impractical to work with the full data, so we focus on WGD and NERD (Lei et al., 2023) using mini-batch stochastic gradient descent. The BA and hybrid WGD algorithms do not directly apply in the stochastic setting, since performing the BA updates on randomly drawn samples tends to cause divergence (also discussed in Sec. 4.3).

We plot the resulting R-D bounds on the two datasets in Fig. 6, and observe that WGD yields similar or improved upper bounds compared to existing methods. For NERD, we set the default 40000 in the code provided by (Lei et al., 2023), on both datasets. On the physics data, we use only 4000 particles for WGD and outperform NERD on the physics dataset. The speech dataset appears more information dense than the physics dataset, and both NERD and WGD encounter the issue of a $\log(n)$ upper bound on the rate estimate as described in Sec. 10.5. Therefore, we increase to 20000 for WGD while this is no longer feasible for NERD, and as a result WGD provides better R-D estimates than NERD particularly in the low-distortion regime. The RD-VAE upper bound (Yang and Mandt, 2022) does not face this issue, and performs more favorably in this regime.

Finally, in the rightmost panel of Fig. 6 we plot the R-D bound estimates for WGD and NERD with increasing n on the physics dataset. We again observe a tighter bound from WGD across the R-D curve, and observe similar results on the speech dataset.

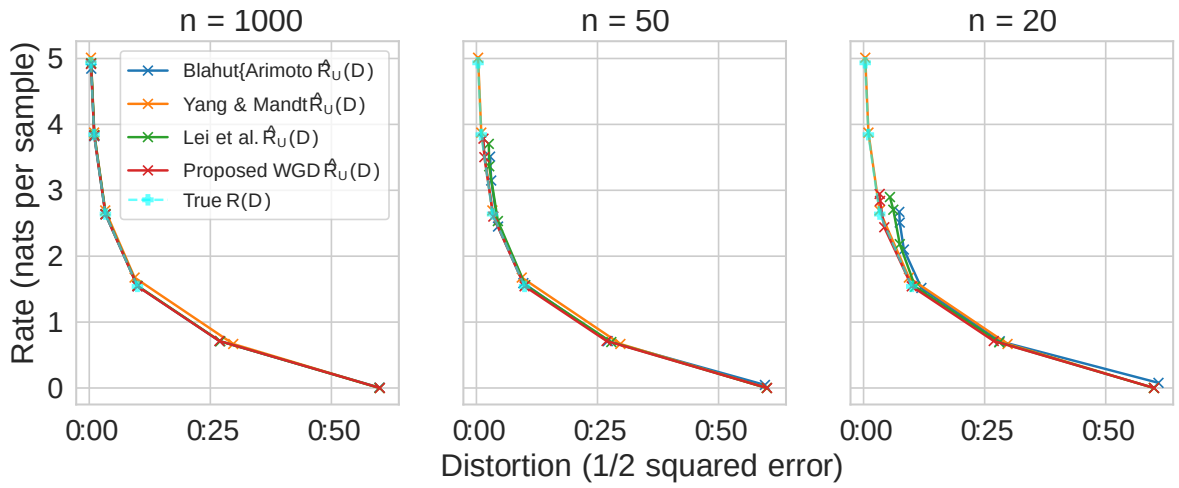

Figure 5. Final R-D upper bounds for the source $= S \quad N(0; 0{:}1I)$ in the maximum-likelihood deconvolution problem (Sec. 5.1), with different settings of n for BA (Blahut, 1972; Arimoto, 1972), NERD (Lei et al., 2023), and the WGD algorithm. The result using the hybrid WGD algorithm (Sec. 4.3) overlaps with that of WGD, hence is omitted for better readability. The ground truth $R(D)$ is known analytically for $\frac{1}{2}$ and computed numerically (see discussion in the text), and is drawn in cyan. The RD-VAE upper bound (orange; the same in each subplot) agrees fairly well with the true $R(D)$ except for some looseness when the distortion is between 0.1 and 0.25. **Left:** when the number of particles is large ($n = 1000$), BA, WGD, and NERD give similarly R-D upper bound estimates close to the true $R(D)$. **Middle:** as we allow ourselves to use fewer particles, e.g., $n = 50$, the bounds from BA, WGD, and NERD start to deviate from the true $R(D)$, with WGD appearing the least affected out of the three. **Right:** as we decrease n further to 20, WGD still mostly preserves the true $R(D)$, while BA and NERD shows much larger deviation.