Telescoping Density-Ratio Estimation

Benjamin Rhodes
School of Informatics
University of Edinburgh
ben.rhodes@ed.ac.uk

Kai Xu
School of Informatics
University of Edinburgh
kai.xu@ed.ac.uk

Michael U. Gutmann
School of Informatics
University of Edinburgh
michael.gutmann@ed.ac.uk

Abstract

Density-ratio estimation via classification is a cornerstone of unsupervised learning. It has provided the foundation for state-of-the-art methods in representation learning and generative modelling, with the number of use-cases continuing to proliferate. However, it suffers from a critical limitation: it fails to accurately estimate ratios p/q for which the two densities differ significantly. Empirically, we find this occurs whenever the KL divergence between p and q exceeds tens of nats. To resolve this limitation, we introduce a new framework, telescoping density-ratio estimation (TRE), that enables the estimation of ratios between highly dissimilar densities in high-dimensional spaces. Our experiments demonstrate that TRE can yield substantial improvements over existing single-ratio methods for mutual information estimation, representation learning and energy-based modelling.

1 Introduction

Unsupervised learning via density-ratio estimation is a powerful paradigm in machine learning [67] that continues to be a source of major progress in the field. It consists of estimating the ratio p/q from their samples without separately estimating the numerator and denominator. A common way to achieve this is to train a neural network classifier to distinguish between the two sets of samples, since for many loss functions the ratio p/q can be extracted from the optimal classifier [67, 21, 45]. This discriminative approach has been leveraged in diverse areas such as covariate shift adaptation [66, 70], energy-based modelling [22, 4, 60, 71, 38, 19], generative adversarial networks [15, 53, 48], bias correction for generative models [20, 18], likelihood-free inference [57, 69, 8, 13], mutual-information estimation [2], representation learning [30, 31, 54, 25, 27], Bayesian experimental design [34, 35] and off-policy reward estimation in reinforcement learning [41]. Across this diverse set of applications, density-ratio based methods have consistently yielded state-of-the-art results.

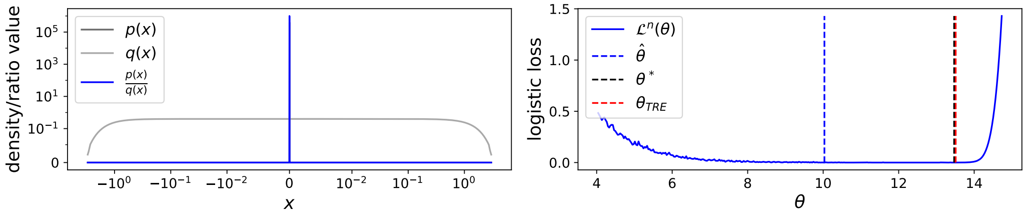
Despite the successes of discriminative density-ratio estimation, many existing loss functions share a severe limitation. Whenever the 'gap' between p and q is large, the classifier can obtain almost perfect accuracy with a relatively poor estimate of the density ratio. We refer to this failure mode as the *density-chasm problem*—see Figure 1a for an illustration. We observe empirically that the density-chasm problem manifests whenever the KL-divergence $D_{KL}(p \parallel q)$ exceeds ~ 20 nats[1]. This observation accords with recent findings in the mutual information literature regarding the limitations of density-ratio based estimators of the KL [44, 59, 64]. In high dimensions, it can easily occur that two densities p and q will have a KL-divergence measuring in the hundreds of nats, and so the ratio may be virtually intractable to estimate with existing techniques.

In this paper, we propose a new framework for estimating density-ratios that can overcome the density-chasm problem. Our solution uses a 'divide-and-conquer' strategy composed of two steps. The first step is to gradually transport samples from p to samples from q, creating a chain of intermediate datasets. We then estimate the density-ratio between consecutive datasets along this

---

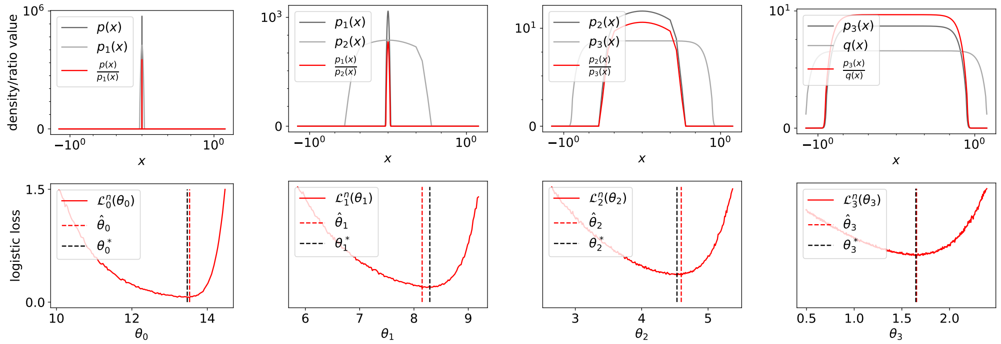
[1] 'nat' being a unit of information measured using the natural logarithm (base e)

Preprint. Under review.



(a) Density-ratio estimation between an extremely peaked Gaussian p ($\sigma = 10^{-6}$) and a broad Gaussian q ($\sigma = 1$) using a single-parameter quadratic classifier. **Left**: A log-log scale plot of the densities and their ratio. **Right**: the solid blue line is the finite-sample logistic loss (Eq. 2) for 10,000 samples. Despite the large sample size, the minimiser (dotted blue line) is far from optimal (dotted black line). The dotted red line is the newly introduced TRE solution, which almost perfectly overlaps with the dotted black line.

$$\frac{p}{q} = \frac{p}{p_1} \times \frac{p_1}{p_2} \times \frac{p_2}{p_3} \times \frac{p_3}{q}$$



(b) Telescoping density-ratio estimation applied to the problem in (a), using the same 10,000 samples from p and q. **Top row**: a collection of ratios, where p_1, p_2 and p_3 are constructed by deterministically interpolating between samples from p and q. **Bottom row**: the logistic loss function for each ratio estimation problem. Observe that the finite-sample minimisers of each objective (red dotted lines) are either close to or exactly overlapping their optima (black dotted lines). After estimating each ratio, we then combine them by taking their product.

Figure 1: Illustration of standard density-ratio estimation vs. telescoping density-ratio estimation.

chain, as illustrated in the top row of Figure 1b. Unlike the original ratio p/q, these 'chained ratios' can be accurately estimated via classification (see bottom row). Finally, we combine the chained ratios via a telescoping product to obtain an estimate of the original density-ratio p/q. Thus, we refer to the method as Telescoping density-Ratio Estimation (TRE).

We empirically demonstrate that TRE can accurately estimate density-ratios using deep neural networks on high-dimensional problems, significantly outperforming existing single-ratio methods. We show this for two important applications: representation learning via mutual information (MI) estimation and the learning of energy-based models (EBMs).

In the context of mutual information estimation, we show that TRE can accurately estimate large MI values of 30+ nats, which is recognised to be an outstanding problem in the literature [59]. However, obtaining accurate MI estimates is often not our *sole* objective; we also care about learning representations from e.g. audio or image data that are useful for downstream tasks such as classification or clustering. To this end, our experimental results for representation learning confirm that TRE offers substantial gains over a range of existing single-ratio baselines.

In the context of energy-based modelling, we show that TRE can be viewed as an extension of noise-contrastive estimation [22] that more efficiently scales to high-dimensional data. Whilst energy-based modelling has been a topic of interest in the machine learning community for some time [63], there has been a recent surge of interest, with a wave of new methods for learning deep EBMs in high dimensions [10, 6, 65, 40, 17, 76]. These methods have shown promising results for image and 3D shape synthesis [74], hybrid modelling [16], and modelling of exchangeable data [75].

However, many of these methods result in expensive/challenging optimisation problems, since they rely on approximate Markov chain Monte Carlo (MCMC) sampling during learning [10, 16, 76], or on adversarial optimisation [6, 17, 76]. In contrast, TRE requires no MCMC during learning and uses a well-defined, non-adversarial, objective function. Moreover, as we show in our mutual information experiments, TRE is applicable to discrete data, whereas all other recent EBM methods only work for continuous random variables. Applicability to discrete data makes TRE especially promising for domains such as natural language processing, where noise-contrastive estimation has been widely used [46, 36, 1].

2 Discriminative ratio estimation and the density-chasm problem

Suppose p and q are two densities for which we have samples, and that $q(\mathbf{x}) > 0$ whenever $p(\mathbf{x}) > 0$. We can estimate the density-ratio $r(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ by training a classifier to distinguish samples from p and q [23, 67, 22]. There are many choices for the loss function of the classifier [67, 58, 21, 45, 59], but in this paper we concentrate on the widely used logistic loss

$$\mathcal{L}(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x}_1 \sim p} \log \left(\frac{r(\mathbf{x}_1; \boldsymbol{\theta})}{1 + r(\mathbf{x}_1; \boldsymbol{\theta})} \right) - \mathbb{E}_{\mathbf{x}_2 \sim q} \log \left(\frac{1}{1 + r(\mathbf{x}_2; \boldsymbol{\theta})} \right), \tag{1}$$

where $r(\mathbf{x}; \boldsymbol{\theta})$ is a non-negative ratio estimating model. To enforce non-negativity, r is typically expressed as the exponential of an unconstrained function such as a neural network. For a correctly specified model, the minimiser of this loss, $\boldsymbol{\theta}^*$, satisfies $r(\mathbf{x}; \boldsymbol{\theta}^*) = p(\mathbf{x})/q(\mathbf{x})$, without needing any normalisation constraints [22]. Other classification losses do not always have this self-normalising property, but only yield a estimate proportional to the true ratio—see e.g. [59].

The density-chasm problem

We experimentally find that density-ratio estimation via classification typically works well when p and q are 'close' e.g. the KL divergence between them is less than ~ 20 nats. However, for sufficiently large gaps, which we refer to as *density-chasms*, the ratio estimator is often severely inaccurate. This raises the obvious question: what is the cause of such inaccuracy?

There are many possible sources of error: the use of misspecified models, imperfect optimisation algorithms, and inaccuracy stemming from Monte Carlo approximations of the expectations in (1). We argue that this mundane final point—Monte Carlo error due to finite sample size—is actually sufficient for inducing the density-chasm problem. Figure 1a depicts a toy problem for which the model is well-specified, and because it is 1-dimensional (w.r.t. θ), optimisation is straightforward using grid-search. And yet, if we use a sample size of $n = 10,000$ and minimise the finite-sample loss

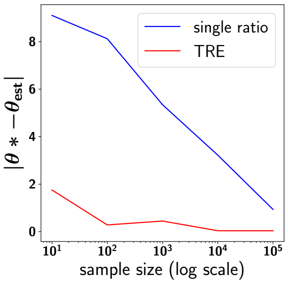


Figure 2: Sample efficiency curves for the experiment in Figure 1. Single ratio estimation can be extremely sample-inefficient.

$$\mathcal{L}^n(\theta) = \sum_{i=1}^{n} -\log \left(\frac{r(x_1^i; \theta)}{1 + r(x_1^i; \theta)} \right) - \log \left(\frac{1}{1 + r(x_2^i; \theta)} \right), \qquad x_1^i \sim p, \; x_2^i \sim q, \tag{2}$$

we obtain an estimate $\hat{\theta}$ that is far from the asymptotic minimiser $\theta^* = \arg\min \mathcal{L}(\theta)$. Repeating this same experiment for different sample sizes, we can empirically measure the method's sample efficiency, which is plotted as the blue curve in Figure 2. For the regime plotted, we see that an exponential increase in sample size only yields a linear decrease in estimation error. This empirical result is concordant with theoretical findings that density-ratio based lower bounds on KL divergences are only tight for sample sizes exponential in the the number of nats [44].

Whilst we focus on the logistic loss, we believe the density chasm problem is a broader phenomenon. As shown in the appendix, the issues identified in Figure 1 and the sample inefficiency seen in Figure 2 also occur for other commonly used discriminative loss functions.

Thus, when faced with the density-chasm problem, simply increasing the sample size is a highly inefficient solution and not always possible in practice. This begs the question: is there a more intelligent way of using a fixed set of samples from p and q to estimate the ratio?

3 Telescoping density-ratio estimation

We introduce a new framework for estimating density-ratios p/q that can overcome the density-chasm problem in a *sample-efficient* manner. Intuitively, the density-chasm problem arises whenever classifying between p and q is 'too easy'. This suggests that it may be fruitful to decompose the task into a collection of harder sub-tasks.

For convenience, we make the notational switch $p \equiv p_0, \;\; q \equiv p_m$ (which we will keep going forward), and expand the ratio via a telescoping product

$$\frac{p_0(\mathbf{x})}{p_m(\mathbf{x})} = \frac{p_0(\mathbf{x})}{p_1(\mathbf{x})}\frac{p_1(\mathbf{x})}{p_2(\mathbf{x})}\cdots\frac{p_{m-2}(\mathbf{x})}{p_{m-1}(\mathbf{x})}\frac{p_{m-1}(\mathbf{x})}{p_m(\mathbf{x})}, \tag{3}$$

where, ideally, each p_k is chosen such that a classifier cannot easily distinguish it from its two neighbouring densities. Instead of attempting to build one large 'bridge' (i.e. density-ratio) across the density-chasm, we propose to build many small bridges between intermediate 'waymark' distributions. The two key components of the method are therefore:

1. **Waymark creation.** We require a method for *gradually* transporting samples $\{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$ from p_0 to samples $\{\mathbf{x}_m^1, \dots, \mathbf{x}_m^n\}$ from p_m. At each step in the transportation, we obtain a new dataset $\{\mathbf{x}_k^1, \dots, \mathbf{x}_k^n\}$ where $k \in \{0, \dots m\}$. Each intermediate dataset can be thought of as samples from an implicit distribution p_k, which we refer to as a *waymark* distribution.
2. **Bridge-building:** A method for learning a set of parametrised density-ratios between consecutive pairs of waymarks $r_k(\mathbf{x}; \boldsymbol{\theta}_k) \approx p_k(\mathbf{x})/p_{k+1}(\mathbf{x})$ for $k = 0, \dots, m-1$, where each bridge r_k is a non-negative function. We refer to these ratio estimating models as *bridges*. Note that the parameters of the bridges, $\{\boldsymbol{\theta}_k\}_{k=0}^{m-1}$, can be totally independent or they can be partially shared.

An estimate of the original ratio is then given by the product of the bridges

$$r(\mathbf{x}; \boldsymbol{\theta}) = \prod_{k=0}^{m-1} r_k(\mathbf{x}; \boldsymbol{\theta}_k) \approx \prod_{k=0}^{m-1} \frac{p_k(\mathbf{x})}{p_{k+1}(\mathbf{x})} = \frac{p_0(\mathbf{x})}{p_m(\mathbf{x})}, \tag{4}$$

where $\boldsymbol{\theta}$ is the concatenation of all $\boldsymbol{\theta}_k$ vectors. Because of the telescoping product in (4), we refer to the method as Telescoping density-Ratio Estimation (TRE).

3.1 Waymark creation

In this paper, we consider two simple, deterministic waymark creation mechanisms: *linear combinations* and *dimension-wise mixing*. We find these mechanisms yield good performance and are computationally cheap. However, we note that other mechanisms are possible, and are a promising topic for future work.

Linear combinations. Given a random pair $\mathbf{x}_0 \sim p_0$ and $\mathbf{x}_m \sim p_m$, define the k^{th} waymark via

$$\mathbf{x}_k = \sqrt{1 - \alpha_k^2}\, \mathbf{x}_0 + \alpha_k \mathbf{x}_m, \qquad\qquad k = 0, \dots, m \tag{5}$$

where the α_k form an increasing sequence from 0 to 1, which control the distance of $\mathbf{x}_k$ from $\mathbf{x}_0$. For our applications, each dimension of p_0 and p_m has the same variance[2] and the coefficients in (5) are chosen to preserve this variance, with the goal being to match basic properties of the waymarks and thereby make consecutive classification problems harder.

Dimension-wise mixing. An alternative way to 'mix' two vectors is to concatenate different subsets of their dimensions. Given a d-length vector $\mathbf{x}$, we can partition it into m sub-vectors of length d/m, assuming d is divisible by m. We denote this as $\mathbf{x} = (\mathbf{x}[1], \dots, \mathbf{x}[m])$, where each $\mathbf{x}[i]$ has length d/m. Using this notation, define the k^{th} waymark via

$$\mathbf{x}_k = (\mathbf{x}_m[1], \, \dots, \, \mathbf{x}_m[k], \, \mathbf{x}_0[k+1], \, \dots, \, \mathbf{x}_0[m]) \qquad\qquad k = 0, \dots, m \tag{6}$$

where, again, $\mathbf{x}_0 \sim p_0$ and $\mathbf{x}_m \sim p_m$ are randomly paired.

---

[2] For MI estimation this always holds, for energy-based modelling this is enforceable via the choice of p_m.

Number and spacing. Given these two waymark generation mechanisms, we still need to decide the *number* of waymarks, m, and, in the case of linear combinations, how the α_k are spaced in the unit interval. We treat these quantities as hyperparameters, and demonstrate in the experiments (Section 4) that tuning them is feasible with a limited search budget.

3.2 Bridge-building

Each bridge $r_k(\mathbf{x}; \boldsymbol{\theta}_k)$ in (4) can be learned via binary classification using a logistic loss function as described in Section 2. Solving this collection of classification tasks is therefore a multi-task learning (MTL) problem—see [62] for a review. Two key questions in MTL are how to *share parameters* and how to define a *joint objective function*.

Parameter sharing. We break the construction of the bridges $r_k(\mathbf{x}; \boldsymbol{\theta}_k)$ into two stages: a (mostly) shared body computing hidden vectors $f_k(\mathbf{x})$[3], followed by bridge-specific heads. The body f_k is a deep neural network with shared parameters and pre-activation per-hidden-unit scales and biases for each bridge (see appendix for details). Similar parameter sharing schemes have been successfully used in the multi-task learning literature [7, 11]. The heads map the hidden vectors $f_k(\mathbf{x})$ to the scalar $\log r_k(\mathbf{x}; \boldsymbol{\theta}_k)$. We use either linear or quadratic mappings depending on the application; the precise parameterisation is stated in each experiment section.

TRE loss function. The TRE loss function is given by the average of the m logistic losses

$$\mathcal{L}_{\text{TRE}}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{k=0}^{m-1} \mathcal{L}_k(\boldsymbol{\theta}_k), \tag{7}$$

$$\mathcal{L}_k(\boldsymbol{\theta}_k) = -\mathbb{E}_{\mathbf{x}_k \sim p_k} \log \left(\frac{r_k(\mathbf{x}_k; \boldsymbol{\theta}_k)}{1 + r_k(\mathbf{x}_k; \boldsymbol{\theta}_k)} \right) - \mathbb{E}_{\mathbf{x}_{k+1} \sim p_{k+1}} \log \left(\frac{1}{1 + r_k(\mathbf{x}_{k+1}; \boldsymbol{\theta}_k)} \right). \tag{8}$$

This simple *unweighted* average works well empirically. More sophisticated multi-task weighting schemes exist [5], but preliminary experiments suggested they were not worth the extra complexity.

During optimisation, we approximate the expectations in (8) with mini-batch averages, which also results in an unbiased gradient estimate for $\boldsymbol{\theta}$. We first draw B samples each from p_0 and p_m, and then randomly pair members from each set. For each pair, we construct *trajectories* using either equation (5) or (6). The fact that we use trajectories, where any two points along a given trajectory will be correlated (except possibly the first & last), means that the gradient of (7) w.r.t. $\boldsymbol{\theta}$ is estimated using shared sources of randomness. Coupling sources of randomness in this way can act as a form of variance reduction [49], and we do indeed find that it enhances learning considerably.

3.3 TRE applied to mutual information estimation

The mutual information (MI) between two random variables $\mathbf{u}$ and $\mathbf{v}$ can be written as

$$I(\mathbf{u}, \mathbf{v}) = \mathbb{E}_{p(\mathbf{u}, \mathbf{v})} \Big[\log r(\mathbf{u}, \mathbf{v}) \Big], \qquad r(\mathbf{u}, \mathbf{v}) = \frac{p(\mathbf{u}, \mathbf{v})}{p(\mathbf{u}) p(\mathbf{v})}. \tag{9}$$

Given samples from the joint density $p(\mathbf{u}, \mathbf{v})$, one obtains samples from the product-of-marginals $p(\mathbf{u})p(\mathbf{v})$ by shuffling the $\mathbf{v}$ vectors across the dataset. This then enables standard density-ratio estimation to be performed.

For TRE, we require waymark samples. To generate these, we take a sample from the joint, $\mathbf{x}_0 = (\mathbf{u}, \mathbf{v}_0)$, and a sample from the product-of-marginals, $\mathbf{x}_m = (\mathbf{u}, \mathbf{v}_m)$, where $\mathbf{u}$ is held fixed and only $\mathbf{v}$ is altered. We then apply a waymark construction mechanism from Section 3.1 to generate $\mathbf{x}_k = (\mathbf{u}, \mathbf{v}_k)$, for $k = 0, \ldots, m$.

3.4 TRE applied to energy-based modelling

Discriminative density-ratio estimation can be used to learn energy-based models (EBM). All we require is that the denominator density q has an analytically tractable expression, and then $\phi(\mathbf{x}; \boldsymbol{\theta}) := r(\mathbf{x}; \boldsymbol{\theta}) q(\mathbf{x})$ is automatically a model of p. This model is typically unnormalised—or *energy-based*—because $\phi(\mathbf{x}; \boldsymbol{\theta})$ is not constrained to integrate to 1 for all values of $\boldsymbol{\theta}$.

---

[3] For simplicity, we suppress the parameters of f_k, and will do the same for r_k in the experiments section.

If we choose to estimate the ratio r via classification with a single logistic loss, then this corresponds to noise-contrastive estimation (NCE) [22], which has many appealing theoretical properties [22, 61, 72]. However, to avoid the density-chasm problem, we can estimate r with TRE instead. The resulting EBM will be a product-of-experts model [26] defined as the product of q (i.e. p_m) with the telescoping product in (4). Following [22], we refer to q as the *noise distribution*.

4 Experiments

We include two toy examples illustrating both the correctness of TRE and the fact that it can solve problems which verge on the intractable for standard density ratio estimation. We then demonstrate the utility of TRE on two high-dimensional complex tasks, providing clear evidence that it substantially improves on standard single-ratio baselines.

For experiments with continuous random variables, we use the linear combination waymark mechanisms in (5); otherwise, for discrete variables, we use dimension-wise mixing (6). We grid-search over the number of waymarks m and, for the linear combination mechanism, an extra spacing parameter, defining the α_k in (5). Grid-searches for all hyperparameters are stated in the appendix.

4.1 1d peaked ratio

The basic setup is stated in Figure 1a. For TRE, we use independently parameterised quadratic bridges $\log r_k(x) = w_k x^2 + b_k$, where this model was chosen to contain the ground-truth solution.

Figure 2 shows the results. The sample efficiency curves clearly demonstrate that, across all sample sizes, TRE is significantly more accurate than single ratio estimation. In fact, TRE obtains a better solution with 100 samples than single-ratio estimation does with 100,000 samples: a three orders of magnitude improvement.

4.2 High-dimensional ratio with large MI

This toy problem has been widely used in the mutual information literature [2, 59]. Let $\mathbf{x} \in \mathbb{R}^{2d}$ be a Gaussian random variable, with block-diagonal covariance matrix, where each block is 2×2 with 1 on the diagonal and 0.8 on the off-diagonal. We then estimate the ratio between this Gaussian and a standard normal distribution. This problem can be viewed as an MI estimation task or an energy-based modelling task—see the appendix for full details.

We apply TRE using quadratic bridges of the form: $\log r_k(\mathbf{x}) = \mathbf{x}^T \mathbf{W}_k \mathbf{x} + b_k$. The results in Figure 3 show that single ratio estimation becomes severely inaccurate for MI values greater than 20 nats. In contrast, TRE can accurately estimate MI values as large as 80 nats for 320 dimensional variables. To our knowledge, TRE is the first discriminative MI estimation method that can scale this gracefully.

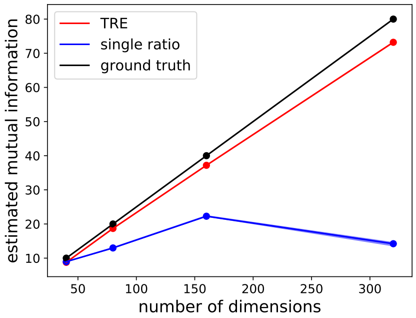


Figure 3: High-dimensional Gaussian results, showing estimated MI as a function of the dimensionality. Errors bars were computed over 5 random seeds, but are too small to see.

4.3 MI estimation & representation learning on SpatialMultiOmniglot

We applied TRE to the SpatialMultiOmniglot problem taken from [55][4] where characters from Omniglot are spatially stacked in an $n \times n$ grid, where each grid position contains characters from a fixed alphabet. Following [55], the individual pixel values of the characters are not considered random variables; rather, we treat the grid as a collection of n^2 categorical random variables whose realisations are the characters from the respective alphabet. Pairs of grids, $(\mathbf{u}, \mathbf{v})$, are then formed such that corresponding grid-positions contain alphabetically consecutive characters. Given this setup, the ground truth MI can be calculated (see appendix).

Each bridge in TRE uses a separable architecture [59] given by $\log r_k(\mathbf{u}, \mathbf{v}) = g(\mathbf{u})^T \mathbf{W}_k f_k(\mathbf{v})$, where g and f_k are 14-layer convolutional ResNets [24] and f_k uses the parameter-sharing scheme

---

[4] We mirror their experimental setup as accurately as possible, however we were unable to obtain their code.

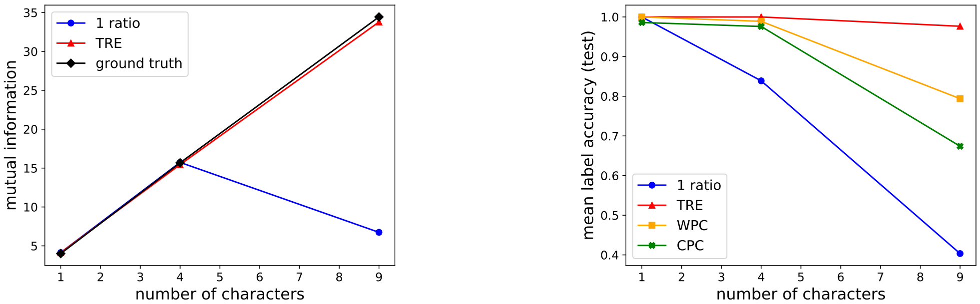


Figure 4: **Left**: mutual information results. TRE accurately estimates the ground-truth MI even for large values of ~ 35 nats. **Right**: representation learning results. All single density-ratio baselines (this includes CPC & WPC) degrade significantly in performance as we increase the number of characters from 4 to 9, dropping by 20-60% in accuracy. In contrast, TRE drops by only $\sim 3\%$.

described in Section 3.2. We note that separable architectures are standard in the MI-based representation learning literature [59]. We construct waymarks using the dimension-wise mixing mechanism (6) with $m = n^2$ (i.e. one dimension is mixed at a time).

After learning, we adopt a standard linear evaluation protocol (see e.g. [54]), where we train *supervised* linear classifiers on top of the output layer $g(\mathbf{u})$ to predict the alphabetic position of each character in $\mathbf{u}$. We compare our results to those reported in [55]. Specifically, we report their baseline method—contrastive predictive coding (CPC) [54], a state-of-the-art representation learning method based on single density-ratio estimation—along with their variant, Wasserstein predictive coding (WPC).

Figure 4 shows the results. The left plot shows that only TRE can accurately estimate high MI values of ~ 35 nats[5]. The representation learning results (right) show that all single density-ratio baselines degrade significantly in performance as we increase the number of characters in a grid (and hence increase the MI). In contrast, TRE always obtains greater than 97% accuracy.

4.4 Energy-based modelling on MNIST

We use TRE to learn energy-based models of the MNIST handwritten digit dataset [39]. We consider three choices of the noise distribution: a multivariate Gaussian, a Gaussian copula and a rational-quadratic neural spline flow (RQ-NSF) [12] with coupling layers [9, 33]. Each distribution is first fitted to the data via maximum likelihood estimation—see appendix for details.

Each of these noise distributions can be expressed as an invertible transformation of a standard normal distribution. That is, each random variable has the form $F(\mathbf{z})$, where $\mathbf{z} \sim \mathcal{N}(0, \mathbb{I})$. Since F already encodes useful information about the data distribution, it makes sense to leverage this when constructing the waymarks in TRE. Specifically, we can generate linear combination waymarks via (5) in $\mathbf{z}$-space, and then map them back to $\mathbf{x}$-space, giving

$$\mathbf{x}_k = F(\sqrt{1-\alpha_k^2}\, F^{-1}(\mathbf{x}_0) + \alpha_k F^{-1}(\mathbf{x}_m)). \tag{10}$$

For a Gaussian, F is linear, and hence (10) is identical to the original waymark mechanism in (5).

We use the parameter sharing scheme from Section 3.2 together with quadratic heads. This gives $\log r_k(\mathbf{x}) = -f_k(\mathbf{x})^T \mathbf{W}_k f_k(\mathbf{x}) - f_k(\mathbf{x})^T \mathbf{b}_k - c_k$, where we set f_k to be an 18-layer convolutional Resnet and constrain $\mathbf{W}_k$ to be positive definite. This constraint enforces an upper limit on the log-density of the EBM, which has been useful in other work [14, 51], and improves results here.

We evaluate the learned EBMs quantitatively via estimated log-likelihood in Table 1 and qualitatively via random samples from the model in Figure 5. For both of these evaluations, we employ NUTS [29] to perform annealed MCMC sampling as explained in the appendix. This annealing procedure provides two estimators of the log-likelihood: the Annealed Importance Sampling (AIS) estimator [50] and the more conservative Reverse Annealed Importance Sampling Estimator (RAISE) [3].

---

[5] [55] do not provide MI estimates for CPC & WPC, but [59] shows that they are bounded by log batch-size.

Table 1: Average negative log-likelihood in bits per dimension (bpd, smaller is better). Exact computation is intractable for EBMs, but we provide 3 estimates: Direct/RAISE/AIS. The 'Direct' estimate uses the NCE/TRE approximate normalising constant.

Noise distribution	Noise	Single ratio (NCE)			TRE		
		Direct	RAISE	AIS	Direct	RAISE	AIS
Gaussian	2.01	1.96	1.99	2.01	1.39	1.35	1.35
Gaussian Copula	1.40	1.33	1.48	1.45	1.24	1.23	1.22
RQ-NSF	1.12	1.09	1.10	1.10	1.09	1.09	1.09

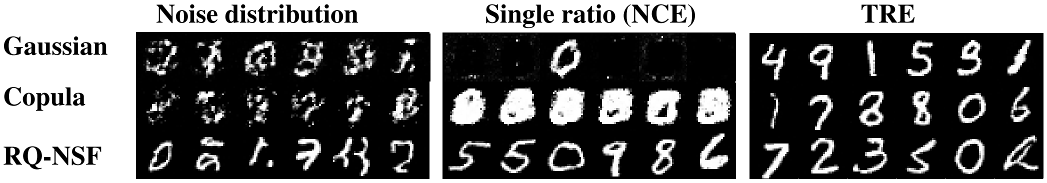


Figure 5: MNIST samples. Each row pertains to a particular noise distribution. The first block shows exact samples from that distribution. The second & third blocks show MCMC samples from an EBM learned with NCE & TRE, respectively.

The results in Table 1 and Figure 5 show that single ratio estimation performs poorly in high-dimensions for simple choices of the noise distribution, and only works well if we use a complex neural density-estimator (RQ-NSF). This illustrates the density-chasm problem explained in Section 2. In contrast, TRE yields improvements for all choices of the noise, as measured by the approximate log-likelihood and the visual fidelity of the samples. TRE's improvement over the Gaussian noise distribution is particularly large: the bits per dimension (bpd) is around 0.66 lower, corresponding to an improvement of roughly 360 nats. Moreover, the samples are significantly more coherent, and appear to be of higher fidelity than the RQ-NSF samples[6], despite the fact that TRE (with Gaussian noise) has a worse log-likelihood. This final point is not contradictory since log-likelihood and sample quality are known to be only loosely connected [68].

Finally, we analysed the sensitivity of our results to the construction of the waymarks and include the results in the appendix. Using TRE with a copula noise distribution as an illustrative case, we found that varying the number of waymarks between 5-30 caused only minor changes in the approximate log-likelihoods, no greater than 0.03 bpd. We also found that if we omit the $\mathbf{z}$-space waymark mechanism in (10), and work in $\mathbf{x}$-space, then TRE's negative log-likelihood increases to 1.33 bpd, as measured by RAISE. This is still significantly better than single-ratio estimation, but does show that the quality of the results depends on the exact waymark mechanism.

5 Conclusion

We introduced a new framework—Telescoping density-Ratio Estimation (TRE)—for learning density-ratios that, unlike existing discriminative methods, can accurately estimate ratios between extremely different densities in high-dimensions.

TRE admits many exciting directions for future work. Firstly, we would like a deeper theoretical understanding of why it is so much more sample-efficient than standard density-ratio estimation. The relationship between TRE and standard methods is structurally similar to the relationship between annealed importance sampling and standard importance sampling. Thus, exploring this connection further may be fruitful. Relatedly, we believe that TRE would benefit from further research on waymark mechanisms. We presented simple mechanisms that have clear utility for both discrete and continuous-valued data. However, we suspect more sophisticated choices may yield improvements, especially if one can leverage domain or task-specific assumptions to intelligently decompose the density-ratio problem. Lastly, whilst this paper has focused on the logistic loss, it would be interesting to more deeply investigate TRE with other discriminative loss functions.

---

[6] We emphasise here that the quality of the RQ-NSF model depends on the exact architecture. A larger model may yield better samples. Thus, we do not claim that TRE generally yields superior results in any sense.

Broader Impact

As outlined in the introduction, density-ratio estimation is a foundational tool in machine learning with diverse applications. Our work, which improves density-ratio estimation, may therefore increase the scope and power of a wide spectrum of techniques used both in research and real-world settings. The broad utility of our contribution makes it challenging to concretely assess the societal impact of the work. However, we do discuss here two applications of density-ratio estimation with obvious potential for positive & negative impacts on society.

Generative Adversarial Networks [15] are a popular class of models which are often trained via density-ratio estimation and are able to generate photo-realistic image/video content. To the extent that TRE can enhance GAN training (a topic we do not treat in this paper), our work could conceivably lead to enhanced ‘deepfakes’, which can be maliciously used in fake-news or identity fraud.

More positively, density-ratio estimation is being used to correct for dataset bias, including the presence of skewed demographic factors like race and gender [18]. While we are excited about such applications, we emphasise that density-ratio based methods are not a panacea; it is entirely possible for the technique to introduce new biases when correcting for existing ones. Future work should continue to be mindful of such a possibility, and look for ways to address the issue if it arises.

Acknowledgments and Disclosure of Funding

Benjamin Rhodes was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

References

[1] Bakhtin, A., Deng, Y., Gross, S., Ott, M., Ranzato, M., and Szlam, A. (2020). Energy-Based Models for Text. *arXiv preprint arXiv:2004.10188*.

[2] Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. (2018). Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540.

[3] Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Accurate and conservative estimates of MRF log-likelihood using reverse annealing. In *Artificial Intelligence and Statistics*, pages 102–110.

[4] Ceylan, C. and Gutmann, M. U. (2018). Conditional Noise-Contrastive Estimation of Unnormalised Models. *Proceedings of the 35th International Conference on Machine Learning*.

[5] Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. (2018). GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803.

[6] Dai, B., Liu, Z., Dai, H., He, N., Gretton, A., Song, L., and Schuurmans, D. (2019). Exponential family estimation via adversarial dynamics embedding. In *Advances in Neural Information Processing Systems*, pages 10977–10988.

[7] De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., and Courville, A. C. (2017). Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604.

[8] Dinev, T. and Gutmann, M. U. (2018). Dynamic likelihood-free Inference via Ratio Estimation (DIRE). *arXiv preprint arXiv:1810.09899*.

[9] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density Estimation using Real NVP. In *International Conference on Learning Representations*.

[10] Du, Y. and Mordatch, I. (2019). Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems*, pages 3603–3613.

[11] Dumoulin, V., Shlens, J., and Kudlur, M. (2016). A learned representation for artistic style. In *International Conference on Learning Representations*.

[12] Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural spline flows. In *Advances in Neural Information Processing Systems*, pages 7509–7520.

[13] Durkan, C., Murray, I., and Papamakarios, G. (2020). On contrastive learning for likelihood-free inference. *arXiv preprint arXiv:2002.03712*.

[14] Durkan, C. and Nash, C. (2019). Autoregressive energy machines. In *International Conference on Machine Learning*.

[15] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[16] Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. (2019). Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*.

[17] Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., and Zemel, R. (2020). Cutting out the middle-man: Training and evaluating energy-based models without sampling. *arXiv preprint arXiv:2002.05616*.

[18] Grover, A., Choi, K., Shu, R., and Ermon, S. (2019a). Fair generative modeling via weak supervision. *arXiv preprint arXiv:1910.12008*.

[19] Grover, A. and Ermon, S. (2018). Boosted generative models. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[20] Grover, A., Song, J., Kapoor, A., Tran, K., Agarwal, A., Horvitz, E. J., and Ermon, S. (2019b). Bias correction of learned generative models using likelihood-free importance weighting. In *Advances in Neural Information Processing Systems*, pages 11056–11068.

[21] Gutmann, M. and Hirayama, J.-I. (2011). Bregman divergence as general framework to estimate unnormalized statistical models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*.

[22] Gutmann, M. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361.

[23] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

[24] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

[25] Hénaff, O. J., Srinivas, A., De Fauw, J., Razavi, A., Doersch, C., Eslami, S. M., and Oord, A. v. d. (2019). Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*.

[26] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8).

[27] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

[28] Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., and Vasudevan, S. (2019). Neutra-lizing bad geometry in hamiltonian monte carlo using neural transport. *arXiv preprint arXiv:1903.03704*.

[29] Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.

[30] Hyvarinen, A. and Morioka, H. (2016). Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems*, pages 3765–3773.

[31] Hyvarinen, A., Sasaki, H., and Turner, R. (2019). Nonlinear ICA using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868.

[32] Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

[33] Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224.

[34] Kleinegesse, S. and Gutmann, M. U. (2019). Efficient Bayesian experimental design for implicit models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 476–485.

[35] Kleinegesse, S. and Gutmann, M. U. (2020). Bayesian experimental design for implicit models by mutual information neural estimation. *arXiv preprint arXiv:2002.08129*.

[36] Kong, L., d'Autume, C. d. M., Ling, W., Yu, L., Dai, Z., and Yogatama, D. (2019). A Mutual Information Maximization Perspective of Language Representation Learning. *arXiv preprint arXiv:1910.08350*.

[37] Kumar, M., Babaeizadeh, M., Erhan, D., Finn, C., Levine, S., Dinh, L., and Kingma, D. (2020). Videoflow: A conditional flow-based model for stochastic video generation. In *International Conference on Learning Representations*.

[38] Lazarow, J., Jin, L., and Tu, Z. (2017). Introspective neural networks for generative modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2774–2783.

[39] LeCun, Y., Cortes, C., and Burges, C. J. (1998). The MNIST database of handwritten digits. URL: http://yann.lecun.com/exdb/mnist/.

[40] Li, Z., Chen, Y., and Sommer, F. T. (2019). Annealed denoising score matching: Learning energy-based models in high-dimensional spaces. *arXiv preprint arXiv:1910.07762*.

[41] Liu, Q., Li, L., Tang, Z., and Zhou, D. (2018). Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pages 5356–5366.

[42] Loshchilov, I. and Hutter, F. (2017). SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

[43] Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Paul Smolley, S. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802.

[44] McAllester, D. and Stratos, K. (2018). Formal limitations on the measurement of mutual information. *arXiv preprint arXiv:1811.04251*.

[45] Menon, A. and Ong, C. S. (2016). Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning*, pages 304–313.

[46] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[47] Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.

[48] Mohamed, S. and Lakshminarayanan, B. (2016). Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*.

[49] Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. (2019). Monte Carlo gradient estimation in machine learning. *arXiv preprint arXiv:1906.10652*.

[50] Neal, R. M. (2001). Annealed importance sampling. *Statistics and computing*, 11(2):125–139.

[51] Ngiam, J. Z., Chen, P. W. K., and Andrew, Y. N. (2011). Learning deep energy models. In *International Conference on Machine Learning*.

[52] Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861.

[53] Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279.

[54] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[55] Ozair, S., Lynch, C., Bengio, Y., Van den Oord, A., Levine, S., and Sermanet, P. (2019). Wasserstein dependency measure for representation learning. In *Advances in Neural Information Processing Systems*, pages 15578–15588.

[56] Papamakarios, G., Pavlakou, T., and Murray, I. (2017). Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347.

[57] Pham, K. C., Nott, D. J., and Chaudhuri, S. (2014). A note on approximating ABC-MCMC using flexible classifiers. *Stat*, 3(1):218–227.

[58] Pihlaja, M., Gutmann, M., and Hyvärinen, A. (2010). A family of computationally efficient and simple estimators for unnormalized statistical models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*.

[59] Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. (2019). On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180.

[60] Rhodes, B. and Gutmann, M. U. (2019). Variational noise-contrastive estimation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2741–2750.

[61] Riou-Durand, L. and Chopin, N. (2018). Noise contrastive estimation: asymptotics, comparison with MC-MLE. *arXiv:1801.10381 [math.ST]*.

[62] Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

[63] Salakhutdinov, R. and Hinton, G. (2009). Deep Boltzmann machines. *J Mach Learn Res*, 24(5):448–455.

[64] Song, J. and Ermon, S. (2019a). Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*.

[65] Song, Y. and Ermon, S. (2019b). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907.

[66] Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. (2008). Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440.

[67] Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density ratio estimation in machine learning*. Cambridge University Press.

[68] Theis, L., Oord, A. v. d., and Bethge, M. (2015). A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.

[69] Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. U. (2016). Likelihood-free inference by ratio estimation. *arXiv preprint arXiv:1611.10242*.

[70] Tsuboi, Y., Kashima, H., Hido, S., Bickel, S., and Sugiyama, M. (2009). Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17:138–155.

[71] Tu, Z. (2007). Learning generative models via discriminative approaches. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

[72] Uehara, M., Matsuda, T., and Komaki, F. (2018). Analysis of noise contrastive estimation from the perspective of asymptotic variance. *arXiv preprint arXiv:1808.07983*.

[73] Ulyanov, D., Vedaldi, A., and Lempitsky, V. S. (2017). Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4105–4113.

[74] Xie, J., Zheng, Z., Gao, R., Wang, W., Zhu, S.-C., and Nian Wu, Y. (2018). Learning descriptor networks for 3d shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8629–8638.

[75] Yang, M., Dai, B., Dai, H., and Schuurmans, D. (2020). Energy-Based Processes for Exchangeable Data. *arXiv preprint arXiv:2003.07521*.

[76] Yu, L., Song, Y., Song, J., and Ermon, S. (2020). Training deep energy-based models with f-divergence minimization. *arXiv preprint arXiv:2003.03463*.

[77] Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019). Self-attention generative adversarial networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7354–7363, Long Beach, California, USA. PMLR.

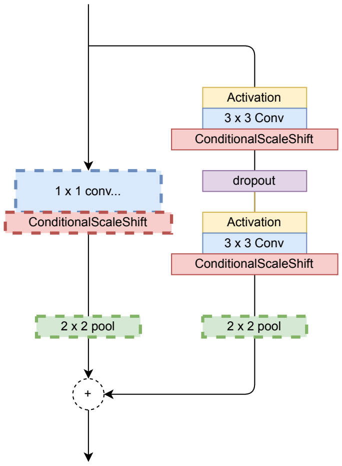
A ResNet architectures with parameter sharing

5×5 conv with 3×3 strides, 32n
CondionalScaleShift
CondResBlock down, 32n
CondResBlock, 32n
CondResBlock down, 64n
CondResBlock, 64n
CondResBlock down, 64n
CondResBlock, 64n
GlobalSumPooling
Dense, 300n

(a) SpatialMultiOmniglot architecture. The multiplier n refers to the width/height of a datapoint, which is an $n \times n$ grid.

3×3 conv, 64
CondionalScaleShift
CondResBlock down, 64
AttentionBlock
CondResBlock, 64
CondResBlock down, 64
CondResBlock, 64
CondResBlock down, 128
CondResBlock, 128
CondResBlock down, 128
CondResBlock, 128
GlobalSumPooling
Dense, 128

(b) MNIST architecture.



(c) CondResBlock. Dashed boxes denote layers that are not always present. The 1×1 conv layer (and the associated CondScaleShift) is only used in blocks where the channel size is altered. The 2×2 pool layer is only used for 'down' blocks.

Figure 6: Convolutional ResNet architectures

In Figure 6, we give the exact architectures for the f_k used in the two high-dimensional experiments on SpatialMultiOmniglot and MNIST. These f_k output a hidden vector for the k^{th} bridge, which is then mapped to the scalar value of the log-ratio, as stated in each experiment section. All convolution operations share their parameters across the bridges, and are thus independent of k.

The only difference between our conditional residual blocks (i.e. 'CondResBlocks') and a standard residual block is the use of 'ConditionalScaleShift' layers. These layers map a hidden vector $\mathbf{z}_k$ to a hidden vector of the same size, $\mathbf{z}'_k$, via

$$\mathbf{z}'_k = \mathbf{s}_k \odot \mathbf{z}_k + \mathbf{b}_k \tag{11}$$

where $\mathbf{s}_k$ and $\mathbf{b}_k$ are bridge-specific parameters and $\odot$ denotes element-wise multiplication. This operation could be thought of as class-conditional Batch Normalisation (BN) [7] without the normalisation. We did not investigate the use of BN, since many energy-based modelling papers (e.g. [10]) found it to harm performance. We did perform preliminary experiments with Instance Normalisation [73] in the context of energy-based modelling, finding it to be harmful to performance.

For the MNIST energy-based modelling experiments, we use average pooling operations since other work [65, 10] has found this to produce higher quality samples than max pooling. For the SpatialMultiOmniglot experiments, we grid-search over average pooling and max pooling. For both sets of experiments, we use LeakyRelu activations with a slope of 0.3.

The MNIST architecture includes an attention block [77] which has been used in GANs to model long-range dependencies in the input image. We found that this attention layer did not yield improvements in estimated log-likelihood, but we think it *may* yield slightly more globally coherent samples. We note that that another commonly used feature in recent GAN and EBM architectures is Spectral Normalisation (SN) [47]. Our preliminary experiments suggested that SN was not beneficial for performance. That said, all of our negative results should be taken with a grain of salt, given the preliminary nature of the experiments.

B Waymark number and spacing

As stated in the main text, the number and (in the case of linear combinations) the spacing of the waymarks are treated as hyperparameters. Finding good values of these hyperparameters is made simpler by the following observations.

- If any of the TRE logistic losses saturate close to 0 during learning, then this indicates that the density-chasm problem has occured for that bridge, and we can terminate the run.

- As illustrated by our sensitivity analysis for MNIST (see Figure 10) it seems that, past a certain point, performance plateaus with the addition of extra waymarks. The fact that it plateaus, and does not decrease, is good news since it indicates that there is little risk of 'overshooting', and obtaining a bad model by having too many waymarks.

We now recall the linear combinations waymark mechanism, given by

$$\mathbf{x}_k = \sqrt{1-\alpha_k^2}\,\mathbf{x}_0 + \alpha_k \mathbf{x}_m, \qquad k = 0, \ldots, m. \tag{12}$$

where m is the number of waymarks. We consider two ways of reducing the coefficients α_k to a function of a single spacing hyperparameter p via

$$\alpha_k = (k/m)^p, \qquad k = 0, \ldots, m, \tag{13}$$

$$\alpha_k = \left\{ \begin{array}{ll} (k/m)^p, & \text{for } k \leq m/2 \\ 1-((m-k)/m)^p, & \text{for } k \geq m/2 \end{array} \right\} \qquad k = 0, \ldots, m. \tag{14}$$

Both mechanisms yield linearly spaced α_k when $p = 1$. For the first mechanism in (13), setting $p > 1$ means the gaps between waymarks *increase* with k (and conversely decrease if $p < 1$). The spacing mechanism in (14) is a kind of symmetrised version of (13).

Table 2 shows the grid-searches we performed for all experiments. We note that these weren't always all performed in parallel. When using linear combinations, we typically set $p = 1$ initially and searched over values of m. If, for all values of m tested, one of the TRE logistic losses saturated close to 0, then we would expand our search space and test different values of p.

C 1d peaked ratio toy experiment

In this experiment we estimate the ratio p_0/p_m, where both densities are Gaussian, $p_0 = \mathcal{N}(0, \sigma_0^2)$ and $p_m = \mathcal{N}(0, \sigma_m^2)$, where $\sigma_0 = 10^{-6}$ and $\sigma_m = 1$. We generate waymarks using the linear combinations mechanism (12), which implies that each waymark distribution is Gaussian, since linear combinations of Gaussian random variables are also Gaussian. Specifically, the waymark distributions have the form

$$p_k(x) = \mathcal{N}(x; 0, \sigma_k^2), \qquad \text{where } \sigma_k = \left[(1-\alpha_k^2)\sigma_0^2 + \alpha_k^2\sigma_m^2\right]^{\frac{1}{2}}. \tag{15}$$

where the σ_k form an increasing sequence between σ_0 and σ_m. The log-ratio between two waymark distributions is therefore given by

$$\log \frac{p_k(x)}{p_{k+1}(x)} = \log\big(\frac{\sigma_{k+1}}{\sigma_k}\big) + \big(\frac{1}{2\sigma_{k+1}^2} - \frac{1}{2\sigma_k^2}\big)x^2. \tag{16}$$

We parameterise the bridges in TRE as

$$\log r_k(x; \theta_k) = \log\big(\frac{\sigma_{k+1}}{\sigma_k}\big) - \exp(\theta_k)x^2, \tag{17}$$

where the quadratic coefficient $-\exp(\theta_k)$ is always negative. We note that this model is well-specified since it contains the ground-truth solution in (16).

Table 2: Waymark hyperparameters for each experiment. Curly braces {} denote grid-searches.

experiment	mechanism	m	spacing	p
1d peaked ratio	linear combo	4	Eq. 13	$\{1, 2, \ldots, 7, 8\}$
high dim, high MI	linear combo	$\frac{d}{40} \times \{1, 2, 3, 4\}$	Eq. 13	1
SpatialMultiOmniGlot	dim-wise mix	d	N/A	N/A
MNIST (z-space)	linear combo	$\{5, 10, 15, 20, 25, 30\}$	Eq. 13	1
MNIST (x-space)	linear combo	$\{5, 10, 15, 20, 25, 30\}$	Eq. { 13, 14 }	$\{1, 1.5, 2\}$

Note: d refers to the dimensionality of the dataset, which is varied for certain experiments.

The bridges can then be combined via summation to provide an estimate of the original log-ratio

$$\log \frac{p_0(x)}{p_m(x)} \approx \sum_{k=0}^{m-1} \log r_k(x; \theta_k) \tag{18}$$

$$= \log \big(\frac{\sigma_m}{\sigma_0}\big) - \sum_{k=0}^{m-1} \exp(\theta_k) x^2 \tag{19}$$

$$= \log \big(\frac{\sigma_m}{\sigma_0}\big) - \exp(\theta_{TRE}) x^2 \tag{20}$$

Where $\theta_{TRE} = \log(\sum_{k=0}^{m-1} \exp(\theta_k))$. We observe that (20) has the same form as (17) if we were to set $m = 1$ in (17) (i.e. if we use a single bridge). Hence θ_{TRE} can be directly compared to the parameter value we would obtain if we used single density-ratio estimation. This is precisely the comparison we make in Figure 1a and Figure 2 of the main text.

C.1 The density chasm problem for non-logistic loss functions

In the main paper, we illustrated the density-chasm problem for the logistic loss using the 1d peaked ratio experiment. Here, we illustrate precisely the same phenomenon for the NWJ/MINE-f loss [52, 2] and a Least Squares (LSQ) loss used by [43]. The loss functions are given by

$$\mathcal{L}_{\text{NWJ}}(\boldsymbol{\theta}) = -\mathbb{E}_p \left[\log r(\mathbf{x}; \boldsymbol{\theta})\right] - 1 + \mathbb{E}_q \left[r(\mathbf{x}; \boldsymbol{\theta})\right] \tag{21}$$

$$\mathcal{L}_{\text{LSQ}}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_p \left[(\sigma(\log(r(\mathbf{x}; \boldsymbol{\theta}))) - 1)^2\right] + \frac{1}{2}\mathbb{E}_q \left[(\sigma(\log(r(\mathbf{x}; \boldsymbol{\theta}))))^2\right], \tag{22}$$

where the σ in (22) denotes the sigmoid function.

In Figures 7 & 8, we can see how single-density ratio estimation performs when using the NWJ and LSQ loss functions for 10,000 samples. the loss curves display the same ‘saturation’ effect seen for the logistic loss, where many settings of the parameter yield an almost identical value of the loss. Moreover, the minimiser of these saturated objectives is far from the ‘true’ minimiser (black dotted lines).

Figures 7 & 8 also show the performance of TRE when each bridge is estimated using the NWJ/LSQ losses. Each TRE loss has a quadratic bowl shape, where the finite-sample minimisers almost perfectly overlap with the true minimisers.

Finally, we plot sample efficiency curves for both the NWJ and LSQ losses, showing the results in Figure 9. We see that single density-ratio estimation with NWJ or LSQ performs poorly, with at best linear gains for exponential increases in sample size. In contrast, if we perform TRE using NWJ or LSQ losses, then we obtain significantly better performance with orders of magnitude fewer samples. These findings are essentially the same as those presented in the main paper for the logistic loss.

D High-dimensional ratio with large MI toy experiment

In this experiment we estimate the ratio p_0/p_m, where both densities are Gaussian, $p_0 = \mathcal{N}(0, \Sigma)$ and $p_m = \mathcal{N}(0, \mathbb{I})$, where Σ is a block-diagonal covariance matrix, where each block is 2×2 with 1 on the diagonal and 0.8 on the off-diagonal. Since we know its analytic form, we can view p_m as a noise distribution, and the ratio-estimation task as an energy-based modelling problem. Alternatively, we may view the problem as a mutual information estimation task, by taking the random variable $\mathbf{x} = (x_1, \dots, x_d) \sim p_0$, and defining $\mathbf{u} = (x_1, x_3, \dots, x_{d-1})$ and $\mathbf{v} = (x_2, x_4 \dots x_d)$. By construction, we therefore have $p(\mathbf{u})p(\mathbf{v}) = \mathcal{N}(\mathbf{x}; 0, \mathbb{I}) = p_m(\mathbf{x})$.

We generate $100,000$ samples for each of the train/validation/test splits. We use a total batch size of 1024, which includes all samples from the waymark trajectories. The bridges in TRE have the form $\log r_k(\mathbf{x}) = \mathbf{x}^T \mathbf{W}_k \mathbf{x} + b_k$, where we enforce that the diagonal entries of $\mathbf{W}_k$ are positive and that the matrix is symmetric. We use the Adam optimiser [32] with an initial learning rate of 0.0001 for TRE, and 0.0005 for single ratio estimation. We use the default Tensorflow settings for β_1, β_2 and ϵ. We train the models for $40,000$ iterations, which takes at most 1 hour.

E MI estimation & representation learning on SpatialMultiOmniglot

We here describe how we created the SpatialMultiOmniglot dataset and give the derivation for the ground truth mutual information values presented in the main paper[7]. We will share the dataset, along with code for the paper, upon publication. We also state the hyperparameter settings used in our experiments.

---

[7] The original work from which we borrow this experiment [55] did not not provide a detailed explanation or code.

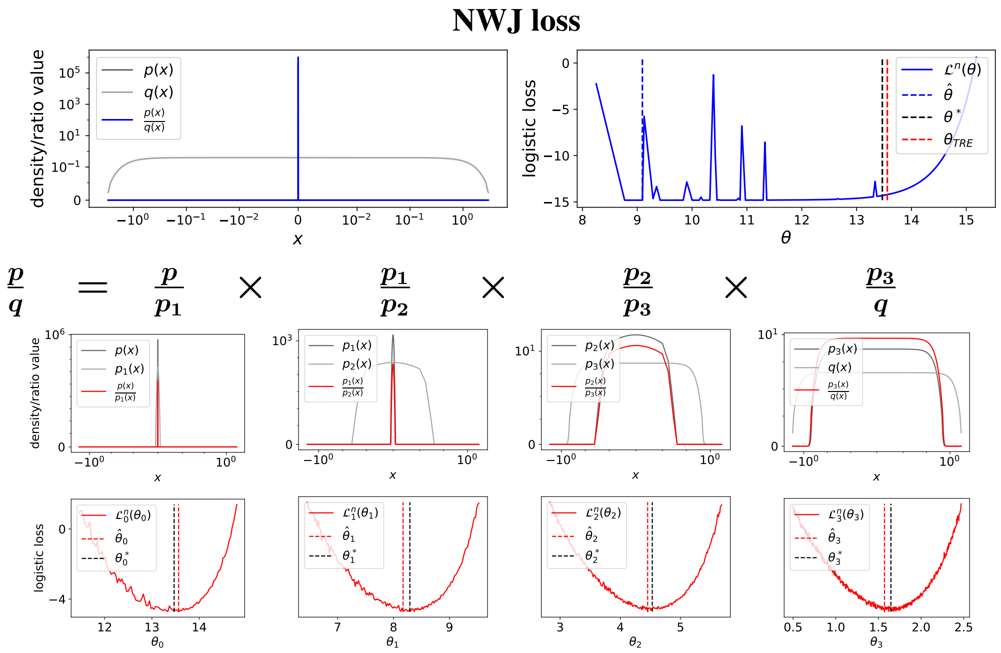

NWJ loss

Figure 7: Replica of Figure 1 from the main text, except that we use the NWJ/MINE-f loss [52, 2] for both the single ratio estimator & for each ratio in TRE.

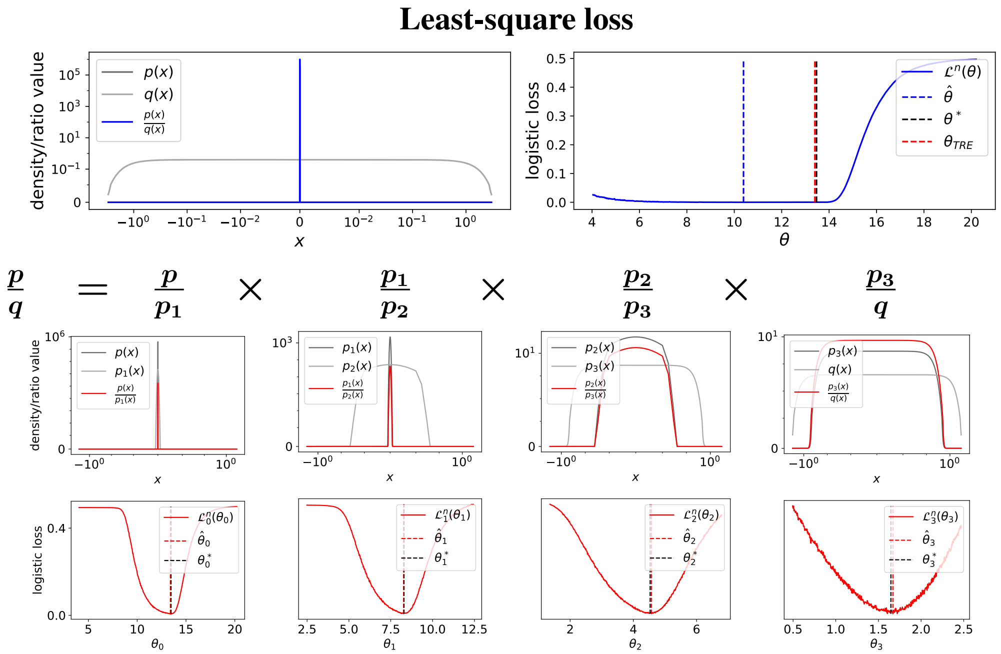
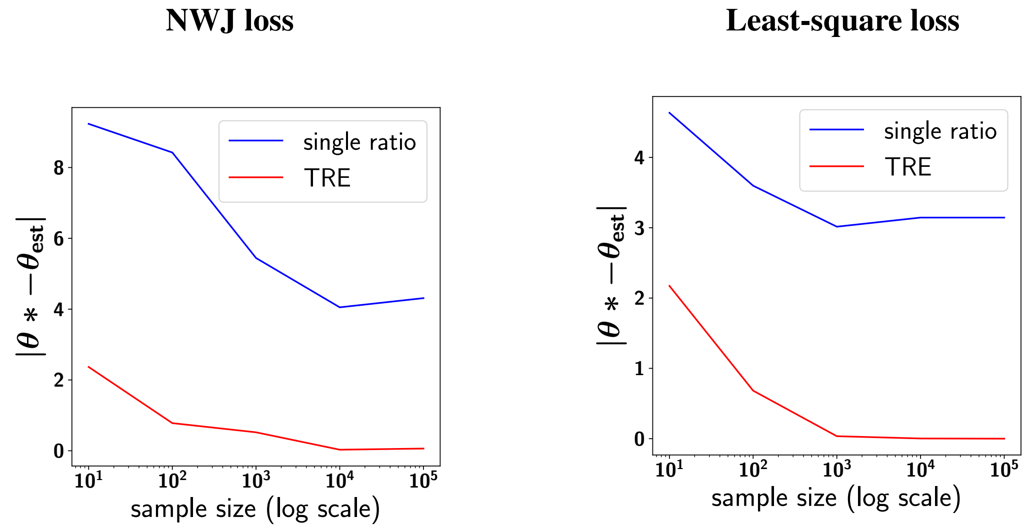

Least-square loss

Figure 8: Replica of Figure 1 from the main text, except that we use the least-square loss from the GAN literature [43] for both the single ratio estimator & for each ratio in TRE.


NWJ loss

Least-square loss

Figure 9: Sample efficiency curves for the 1d peaked ratio experiment, using different loss functions.

E.1 Dataset construction

We take the Tensorflow version of the Omniglot dataset (`https://www.tensorflow.org/datasets/catalog/omniglot`) and resize it to 28 × 28 using the `tf.image.resize` function. We arrange the data into alphabets $\{A_i\}_{i=1}^{l}$, where each alphabet contains n_i characters. The alphabets are sorted by size, so that $n_1 > n_2 > \ldots > n_l$. Each character in a alphabet has 20 different *versions* (e.g. there are 20 different images depicting the letter 'w'). Hence, we can express each alphabet as a set $A_i = \{\{a^i_{j,k}\}_{k=1}^{20}\}_{j=1}^{n_i}$, where $a^i_{j,k}$ refers to the k^{th} version of the j^{th} character of the i^{th} alphabet.

In order to construct the d-dimensional version of the SpatialMultiOmniGlot dataset, we restrict ourselves to the d largest alphabets $\{A_i\}_{i=1}^{d}$. We then sample a vector of categorical random variables

$$\mathbf{j} = (j_1, \ldots, j_d) \sim \text{Cat}(n_1) \times \ldots \text{Cat}(n_d) \tag{23}$$

where the i^{th} categorical distribution is uniform over the set $\{1, \ldots, n_i\}$ and is independent from the other categorical distributions. The vector $\mathbf{j}$ should be thought of as an index vector that specifies a particular character from each of the d alphabets.

We then sample two i.i.d random variables $\mathbf{k}$ and $\mathbf{k}'$, via

$$\mathbf{k} = (k_1, \ldots, k_d) \sim \prod_{i=1}^{d} \text{Cat}(20) \qquad \mathbf{k}' = (k'_1, \ldots, k'_d) \sim \prod_{i=1}^{d} \text{Cat}(20) \tag{24}$$

where, again, each Categorical distribution is independent from the rest. These vectors should be thought of as index vectors that specify a particular version of a character.

Now, we define a datapoint as a tuple $\mathbf{x} = (\mathbf{u}, \mathbf{v})$, where

$$\mathbf{u} = (a^1_{j_1,k_1}, \ldots, a^d_{j_d,k_d}) \qquad \mathbf{v} = (a^1_{j_1+1,k'_1}, \ldots a^d_{j_d+1,k'_d}). \tag{25}$$

In words, we construct $\mathbf{u}$ and $\mathbf{v}$ such that u_i and v_i are consecutive characters within their alphabet (whilst the precise *versions* of the characters are randomised). Finally, we arrange $\mathbf{u}$ and $\mathbf{v}$ into a grid using raster ordering. This is possible since we assume d to be a square number.

Importantly, we emphasise that $\mathbf{u}, \mathbf{v} \in \prod_{i=1}^{d} A_i$ are discrete random variables defined over a set of template images. They are not defined over a space of pixel values, as is usually the case in image-modelling.

E.2 Derivation of ground truth MI

By construction, we have that $\mathbf{u}$ and $\mathbf{v}$ are conditionally independent given $\mathbf{j}$. This means

$$p(\mathbf{u}|\, \mathbf{v}, \mathbf{j}) = p(\mathbf{u}|\, \mathbf{j}). \tag{26}$$

Furthermore, will assume that, for all $\mathbf{u}$ there exists a unique $\mathbf{j}_{\mathbf{u}}$ such that

$$p(\mathbf{j}_{\mathbf{u}}|\, \mathbf{u}) = 1. \tag{27}$$

Similarly, for any $\mathbf{v}$, there exists a unique $\mathbf{j}_{\mathbf{v}}$ satisfying the same condition. In words, this simply means that, given a grid of Omniglot images, we assume there is no ambiguity about which characters are present. Using Bayes' rule, and the fact that for a given $\mathbf{j}$, $\mathbf{u}$ is uniquely determined by $\mathbf{k}$, one can then deduce that

$$p(\mathbf{u}|\,\mathbf{j}) = \left\{ \begin{array}{ll} 0, & \text{for } \mathbf{j} \neq \mathbf{j}_{\mathbf{u}} \\ 20^{-d}, & \text{for } \mathbf{j} = \mathbf{j}_{\mathbf{u}} \end{array} \right\}. \tag{28}$$

and similarly for $\mathbf{v}$.

We now proceed to derive an analytical formula for the ground truth mutual information between $\mathbf{u}$ and $\mathbf{v}$. We show that the mutual information is equal to the sum of the log alphabet sizes $\mathcal{I}(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{d} \log n_i$.

This holds because

$$\begin{aligned}
\mathcal{I}(\mathbf{u}, \mathbf{v}) &= \mathbb{E}_{p(\mathbf{u},\mathbf{v})} \log \frac{p(\mathbf{u}, \mathbf{v})}{p(\mathbf{u})p(\mathbf{v})} && (29)\\
&= \mathbb{E}_{p(\mathbf{u},\mathbf{v})} \log \frac{p(\mathbf{u}|\,\mathbf{v})}{p(\mathbf{u})} && (30)\\
&= \mathbb{E}_{p(\mathbf{u},\mathbf{v})} \log \frac{\sum_{\mathbf{j}} p(\mathbf{u}, \mathbf{j}|\,\mathbf{v})}{\sum_{\mathbf{j}} p(\mathbf{u}, \mathbf{j})} && (31)\\
&= \mathbb{E}_{p(\mathbf{u},\mathbf{v})} \log \frac{\sum_{\mathbf{j}} p(\mathbf{u}|\,\mathbf{j})p(\mathbf{j}|\,\mathbf{v})}{\sum_{\mathbf{j}} p(\mathbf{u}|\,\mathbf{j})p(\mathbf{j})} \qquad \text{by (26)} && (32)\\
&= \mathbb{E}_{p(\mathbf{u},\mathbf{v})} \log \frac{p(\mathbf{u}|\,\mathbf{j}_{\mathbf{v}})}{\sum_{\mathbf{j}} p(\mathbf{u}|\,\mathbf{j})p(\mathbf{j})} \qquad \text{by (27)} && (33)\\
&= \mathbb{E}_{p(\mathbf{u},\mathbf{v})} \log \frac{p(\mathbf{u}|\,\mathbf{j}_{\mathbf{v}})}{p(\mathbf{u}|\,\mathbf{j}_{\mathbf{u}})p(\mathbf{j}_{\mathbf{u}})} \qquad \text{by (28)} && (34)\\
&= \mathbb{E}_{p(\mathbf{u},\mathbf{v})} \log \frac{1}{p(\mathbf{j}_{\mathbf{u}})} \qquad \text{since } \mathbf{j}_{\mathbf{v}} = \mathbf{j}_{\mathbf{u}} \text{ if } p(\mathbf{u}, \mathbf{v}) > 0 && (35)\\
&= \mathbb{E}_{p(\mathbf{u},\mathbf{v})} \log \big(\prod_{i=1}^{d} n_i\big) \qquad \text{since } p(\mathbf{j}) \text{ is uniform} && (36)\\
&= \sum_{i=1}^{d} \log n_i && (37)
\end{aligned}$$

E.3 Experimental settings

We generate 3 versions of the SpatialMultiOmniglot dataset for $d = 1, 4, 9$. For each version, we sample $50,000$ training points and $10,000$ validation and test points. As stated in the main text, we use a separable architecture given by

$$\log r_k(\mathbf{u}, \mathbf{v}) = g(\mathbf{u})^T \mathbf{W}_k f_k(\mathbf{v}), \tag{38}$$

where f_k is a convolutional ResNet whose architecture is given in Figure 6a. The function g is also a convolutional ResNet with almost the same architecture, except that none of its parameters are bridge-specific, and hence the 'ConditionalScaleShift' layers simply become 'ScaleShift' layers, with no dependence on k.

To construct a mini-batch, we first sample a batch from the joint distribution $p(\mathbf{u}, \mathbf{v})$. We then obtain samples from $p(\mathbf{u})p(\mathbf{v})$ by sampling a *second* batch from the joint distribution (which could overlap with the first batch), and shuffling the $\mathbf{v}$ vectors across this second batch. Finally, we construct waymark trajectories as described in the main text. For all experiments, the 'total' batch size is ~ 512, which includes all samples from the waymark trajectories. Thus, as the number of waymarks increases, the number of trajectories in a batch decreases.

We use the Adam optimiser [32] with an initial learning rate of 10^{-4} with default Tensorflow settings for β_1, β_2 and ϵ. We gradually decrease the learning rate over the course of training with cosine annealing [42]. All models are trained using a single NVIDIA Tesla P100 GPU card for $200,000$ iterations, which takes at most a day.

We grid-searched over the type of pooling (max vs. average) and the size of the final dense layer ($150n, 300n$ and $450n$, where $d = n^2$). Interestingly, average pooling was less prone to overfitting and often yielded better final performance, however it was often 'slow to get started', with the TRE losses hardly making any progress during the first quarter of training.

For the representation learning evaluations, we first obtained the hidden representations $g(\mathbf{u})$ for the entire dataset. We then trained a collection of *independent* supervised linear classifiers on top of these representations, in order to predict the alphabetic position of each character in $\mathbf{u}$. We used the L-BFGS optimiser to fit these classifiers via the `tfp.optimizer.lbfgs_minimize` function, setting the maximum iteration number to $10,000$.

F Energy-based modelling on MNIST

We here discuss the parameterisation of the noise distributions used in the experiments, the exact method for sampling from the learned EBMs, and the experimental settings used for TRE.

For all noise distributions and TRE models, we use the Adam optimiser [32] with an initial learning rate of 10^{-4} with default Tensorflow settings for β_1, β_2 and ϵ. We gradually decrease the learning rate over the course of training with cosine annealing [42]. All models are trained using a single NVIDIA Tesla P100 GPU card.

F.1 Noise distributions

As stated in the main text, we consider three noise distributions: a multivariate Gaussian, a Gaussian copula and a rational-quadratic neural spline flow (RQ-NSF), all of which are pre-trained via maximum likelihood estimation.

The full-covariance multivariate Gaussian is by far the simplest, and can be fitted in around a minute via `np.cov`. The Gaussian copula is slightly more complicated. Its density can be written as $p(\mathbf{x}) = \mathcal{N}([s_1(x_1), \ldots, s_d(x_d)]; \mu, \Sigma) \prod_{i=1}^{d} |s_i'(x_i)|$. The s_i are given by the composition of the inverse CDF of a standard normal and the CDF of the univariate x_i. It is possible to exploit this to learn the s_i—as well as μ and Σ—however, we found it slightly simpler to directly parametrise the s_i via flexible rational-quadratic spline functions [12] of which there are official implementations in Tensorflow and Pytorch and to jointly learn all parameters via maximum likelihood. We follow the basic hyperparameter recommendations in [12]. The hyperparameters that required tuning were the number of bins (we use 128) and the interval widths (which we set to 3 times the standard deviation of the data). For optimisation, we used a batch size of 512 and trained for $40,000$ iterations.

Finally, we turn to the RQ-NSF model [12]. We largely adopt the architectural choices of [12], and so for a more detailed explanation, we refer the reader to their work. We use a multi-scale convolutional architecture comprised of 2 levels, where each level contains 8 ‘steps’. A step consists of an actnorm layer, an invertible 1×1 convolution, and a rational-quadratic coupling transform. The coupling transforms are parameterised by a block of convolution operations following [37], which use 64 feature maps. The spline functions use 8 bins and the interval width is set to [-3, 3]. We do not ‘factor out’ half of the variables at the end of each level, but do perform ‘squeeze’ operation and an additional 1×1 convolution. For optimisation, we set the batch size to 256, the dropout rate to 0.1, and train for $200,000$ iterations, which takes under a day.

F.2 Annealed MCMC Sampling

We here describe how we leverage the specific products-of-experts structure of the TRE model to perform annealed MCMC sampling. Firstly, we initialise a set of MCMC chains with i.i.d samples from the noise distribution p_m. We could then run an MCMC sampler with the full TRE model as the target distribution. However, we instead use an annealing procedure, whereby we iteratively sample from a sequence of distributions that interpolate between p_m and p_0. Such distributions can be obtained by multiplying p_m with an increasing number of bridges

$$p_k(\mathbf{x}) = p_m(\mathbf{x}) \prod_{i=k}^{m-1} r_k(\mathbf{x}), \qquad\qquad k = m-1, \ldots 0. \tag{39}$$

To obtain an even smoother interpolation, we further define exponentially-averaged intermediate distributions $p_{k,t}(\mathbf{x}) = p_k(\mathbf{x})^{\beta_t} p_{k+1}(\mathbf{x})^{1-\beta_t}$, where $\{\beta_t\}$ is a decreasing sequence of numbers ranging from 1 to 0.

In addition to obtaining samples, we can simultaneously use this annealing procedure for estimating the log-likelihood of the model via annealed importance sampling (AIS) [50]. We may also run the annealing procedure ‘in reverse’, initialising a chain at a datapoint and iteratively removing bridges until the target distribution of the MCMC sampler is the noise distribution. Using this reverse sampling procedure, we can obtain a second, more conservative, estimate of the log-likelihood via the reverse annealed importance sampling estimator (RAISE) [3].

Whilst in principle any MCMC sampler could be used, the efficiency of different samplers can vary greatly. We choose to use the gradient-based No-U-turn sampler (NUTS) [29], which is a highly efficient method for many applications. We use the official Tensorflow implementation along with most of the default hyperparameter settings. We set the target acceptance rate to 0.6, and use a max tree depth of 6 during the annealed sampling. We also continue to run the sampler after the annealing phase is finished, using a max tree depth of 10. We use a total of 1000 intermediate distributions with 100 parallel chains.

Finally, recall from the main text that each noise distribution in our experiments can be expressed as invertible transformation F of a standard normal distribution. We use this F to further enhance the efficiency of the NUTS sampler, by performing the sampling in the $\mathbf{z}$-space, and then mapping the final results back to $\mathbf{x}$-space. Working in $\mathbf{z}$-space, by the rules of transformations of random variables, the intermediate distributions of (39) become

$$p_k(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \mathbb{I}) \prod_{i=k}^{m-1} r_k(F(\mathbf{z})). \tag{40}$$

AIS and RAISE can still be applied, just as before, to obtain an estimate of the log-likelihood in $\mathbf{z}$-space. The change of variables formula for probability density functions can then be applied to obtain estimated log-likelihoods for the original TRE model in $\mathbf{x}$-space. We note that when the noise distribution is a normalising flow, prior work has demonstrated that $\mathbf{z}$-space MCMC sampling can be significantly more effective than working in the original data space [28].

F.3 Experimental settings

We use the standard version of the MNIST dataset [39], with $50,000$ training points, and $10,000$ validation and test points. We follow the same preprocessing steps as [56], 'dequantizing' the dataset with uniform noise, re-scaling to the unit interval, and then mapping to the real line via a logit transformation.

The architecture for the TRE bridges is given in Figure 6b. The waymark mechanism and associated grid-search is given in Table 2. A consistent observation across all our MNIST experiments was that the first ratio-estimator between the data distribution p_0 and a slightly perturbed data distribution p_1 was extremely prone to overfitting. We found that the only way to mitigate this problem was to simply drop the ratio by setting the α_0 in (12) to a very small value (0.01) rather than exactly 0. Equivalently, this can be viewed as applying standard TRE to a very slightly perturbed data distribution. We note that this perturbation is small enough that is barely visible to the human eye when comparing samples. We conjecture that this problem may stem from the fact that the original MNIST dataset is actually discrete not continuous and the 'dequantizing' perturbation used to make the data continuous is perhaps not sufficient.

To form mini-batches, we sample 25 datapoints each from p_0 and p_m, and then generate waymark trajectories as described in the main text. Thus, the total batch size is $25 \times (m+1)$. We use the optimisation settings described at the beginning of this section, training for $200,000$ iterations, which takes about a day.

F.4 Additional results

In Figure 10, we present a sensitivity analysis showing how the quality of the learned EBM varies as we alter the number of waymarks, as well as the space in which the waymarks are generated. We found that working in $\mathbf{x}$-space yielded lower performance compared to working in $\mathbf{z}$-space, as measured by the most conservative estimator, RAISE. In particular, we found that the $\mathbf{x}$-space mechanism required more waymarks (around 15) to avoid any of the logistic losses saturating close to 0, and it was significantly harder to tune the spacing of the waymarks as indicated by Table 2.

Finally, for the models whose results were given in the main paper, we display extended image samples in Figure 11. Note that these samples are ordered by log-density (lowest density in top left corner, highest in bottom right).

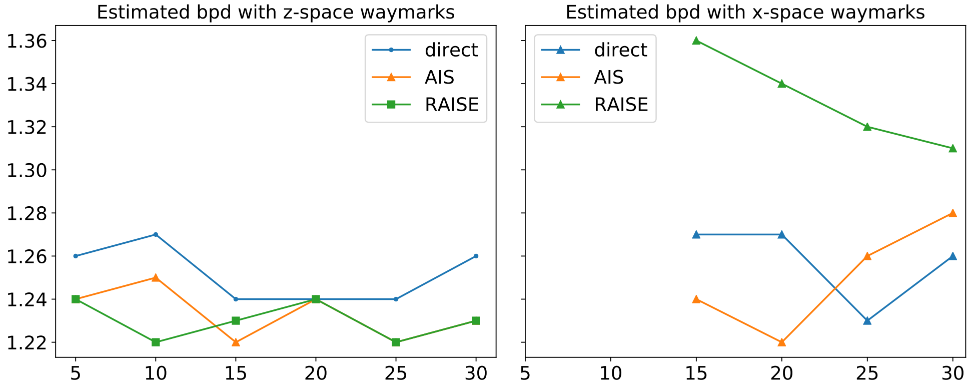
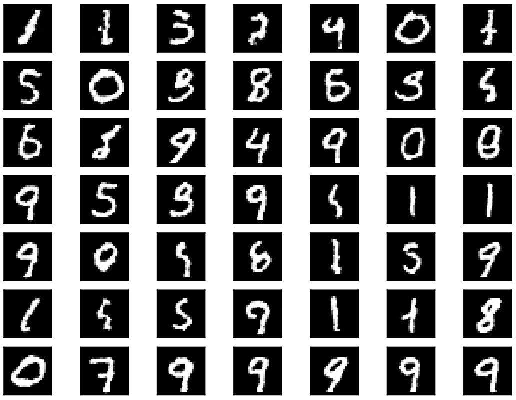
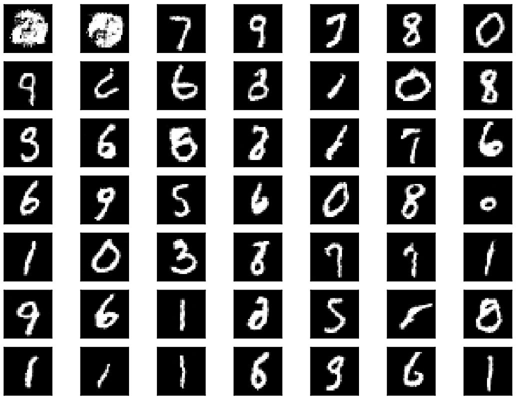


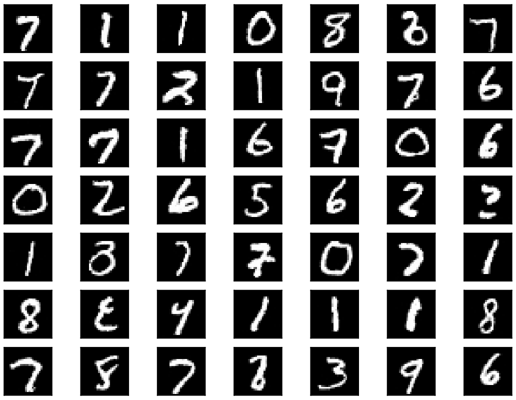
Figure 10: Waymark sensitivity analysis for TRE with copula noise distribution. Both plots show the estimated bits per dimension (bpd) as a function of waymark number. On the left we apply the linear combination waymark mechanism in $\mathbf{z}$-space, whilst on the right we apply it in $\mathbf{x}$-space. As described in Section B, we terminate runs where any of the TRE losses saturate close to 0, which is exactly what happened when using the $\mathbf{x}$-space mechanism for 5 and 10 waymarks.



(a) Samples from TRE model with Gaussian noise distribution, ordered by log-density.



(b) Samples from TRE model with copula noise distribution, ordered by log-density.



(c) Samples from TRE model with RQ-NSF noise distribution, ordered by log-density.

Figure 11: Extended MNIST samples