REMEDI: Corrective Transformations for Improved Neural Entropy Estimation

Viktor Nilsson [*1] **Anirban Samaddar** [*2] **Sandeep Madireddy** [2] **Pierre Nyquist** [1 3]

Abstract

Information theoretic quantities play a central role in machine learning. The recent surge in the complexity of data and models has increased the demand for accurate estimation of these quantities. However, as the dimension grows the estimation presents significant challenges, with existing methods struggling already in relatively low dimensions. To address this issue, in this work, we introduce `REMEDI` for efficient and accurate estimation of differential entropy, a fundamental information theoretic quantity. The approach combines the minimization of the cross-entropy for simple, adaptive base models and the estimation of their deviation, in terms of the relative entropy, from the data density. Our approach demonstrates improvement across a broad spectrum of estimation tasks, encompassing entropy estimation on both synthetic and natural data. Further, we extend important theoretical consistency results to a more generalized setting required by our approach. We illustrate how the framework can be naturally extended to information theoretic supervised learning models, with a specific focus on the Information Bottleneck approach. It is demonstrated that the method delivers better accuracy compared to the existing methods in Information Bottleneck. In addition, we explore a natural connection between `REMEDI` and generative modeling using rejection sampling and Langevin dynamics.

[*]Equal contribution [1]Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden [2]Mathematics and Computer Science Division, Argonne National Laboratory, Chicago IL, USA [3]Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden. Correspondence to: Viktor Nilsson <vikn@kth.se>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Information theoretic quantities such as entropy, cross-entropy, mutual information, relative entropy (Kullback-Leibler divergence), and conditional entropy are abundant in machine learning. Many learning algorithms are derived from such quantities, and recent advances have revealed that they can provide learning objectives on their own (Tishby & Zaslavsky, 2015; Alemi et al., 2016), or in combination with other terms (Sarra et al., 2021; Kingma & Welling, 2013). In some settings, an information theoretic objective may reduce to a simple expression in practical machine learning algorithms; for example: minimizing the forward relative entropy $R(\mathbb{P} \,||\, \mathbb{Q})$ with respect to $\mathbb{Q}$, having samples $\{x_i\} \sim \mathbb{P}$, can be achieved by minimizing the negative log-likelihood. However, these quantities are usually difficult to estimate even in moderately high dimensions (Gao et al., 2018).

In this work, we turn our attention toward the estimation of the differential entropy (DE). This quantity appears in many places throughout machine learning, such as reinforcement learning (Haarnoja et al., 2017), unsupervised learning (Sarra et al., 2021), the Information Bottleneck method (Alemi et al., 2016; Kolchinsky et al., 2019), and dimensionality reduction (Faivishevsky & Goldberger, 2008). Often, differential entropy serves the role of something to be maximized, perhaps under some constraints, such as in the maximum entropy approaches (Jaynes, 1957), or the Information Bottleneck method (Tishby et al., 2000; Tishby & Zaslavsky, 2015). Therefore, it is advantageous that an estimator is differentiable with respect to the data, something not true for many estimators, e.g., k-nearest neighbor-based estimators. There exist kernel-based plug-in estimators that are differentiable, however, they are prohibitively data inefficient in dimensions as low as 10 (see Chapter 20.3 in (Wasserman, 2004)).

Recent works (Schraudolph, 2004; Pichler et al., 2022), have introduced gradient-based learning objectives as upper bounds to the differential entropy, in an extension of classical kernel density estimation techniques (Rosenblatt, 1956; Parzen, 1962; Ahmad & Lin, 1976). However, these estimators still lie in the class of plug-in estimators and are affected by the data inefficiency in large dimensions, see Sec. 3.2 and Appendix C.1.

To this end, our contributions are as follows —

- We introduce the *Relative Entropy MixturE moDel corrective transformatIon* (REMEDI) approach that combines the strengths of recent advances in the estimation of information theoretic quantities (Pichler et al., 2022; Belghazi et al., 2018) to improve DE estimation. The approach takes modern plug-in entropy estimators and refines their estimates with corrections obtained via the Donsker-Varadhan formula.

- We present theoretical results proving the consistency of the proposed estimator, under the assumption that the data is sub-Gaussian, whereas existing related results put compactness assumptions on the support of the data.

- We demonstrate the limitations of the current state-of-the-art plug-in entropy estimators in moderately high dimensions. We show that applying REMEDI corrections to the existing differential entropy estimators significantly improves entropy estimation in benchmark datasets.

- We discuss the application of our approach in supervised learning with the Information Bottleneck framework, to better estimate the mutual information between inputs and the latent space. On classification tasks with the MNIST, CIFAR-10, and ImageNet datasets, we show that REMEDI achieves better classification accuracy than state-of-the-art approaches.

- We explore the generative modeling aspect of our approach by highlighting the connections of REMEDI with (i) rejection sampling, and (ii) stochastic differential equations.

2. Related works

Estimators of differential entropy are usually classified as *plug-in* estimates, *sample-spacings* based estimates, and *nearest neighbor* based estimates (Beirlant et al., 1997). A classical estimator for the differential entropy is Parzen-window estimation, which is provably consistent under weak assumptions (Ahmad & Lin, 1976). It is known that Parzen-window approximation is data-inefficient, see (Wasserman, 2004). Modern extensions of (Ahmad & Lin, 1976) come from (Schraudolph, 2004), where each kernel has its precision matrix parametrized by its lower Cholesky factors, with respect to which it is also differentiable, which enables gradient-based learning. In (Pichler et al., 2022), this is further extended to involve parametrized means and weights for each kernel. The three above-mentioned approaches are all plug-in estimators (Beirlant et al., 1997), where the latter two involve a training step, whereby a mixture of multivariate normal distributions are fitted to the data using gradient-based training.

In recent studies (Belghazi et al., 2018), a new method has been introduced for computing the mutual information $\mathrm{MI}(X;Y)$ between two random variables X, Y, based on neural networks and the Donsker-Varadhan representation formula. The method exploits that the mutual information is the same as the relative entropy between $\mathbb{P}_{XY}$ and $\mathbb{P}_X \otimes \mathbb{P}_Y$, which makes the Donsker-Varadhan formula a lower bound for the mutual information. However, recently (McAllester & Stratos, 2020) it has been shown that such bounds are statistically difficult to approximate when $\mathbb{P}_{XY}$ and $\mathbb{P}_X \otimes \mathbb{P}_Y$ are too different. Using a base distribution, the same type of bounds on the relative entropy can be exploited for computing the differential entropy, then appearing as upper bounds, see Sec. 3. Recent studies (Park & Pardalos, 2021; Aharoni et al., 2022) have taken this direction to neural differential entropy estimation. However, the authors only consider very simple base distributions, which implies poor sample efficiency, likely introducing the problems presented in (McAllester & Stratos, 2020).

In this work, we explore the combination of a more flexible mixture of Gaussian base distribution with the Donsker-Varadhan type bound to propose an improved neural density estimator. In addition, we theoretically prove that our estimator is consistent, under weaker assumptions on the data distribution compared to previous works (Belghazi et al., 2018).

3. Method

In this section, we present the REMEDI approach for entropy estimation. We state the limitations of the existing entropy estimation approaches and discuss the ways REMEDI addresses these limitations. Furthermore, we present the optimization algorithm for solving the loss function using finite data samples.

3.1. Preliminary concepts

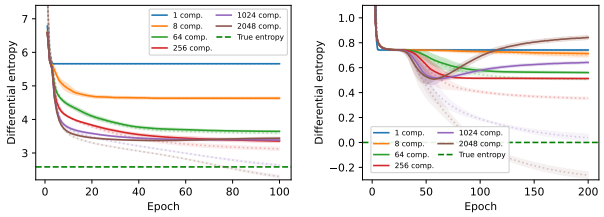
In this section, we define the necessary concepts that will be used throughout the paper.

Let $\mathbb{P}$ and $\mathbb{Q}$ be two probability measures, such that $\mathbb{P} \ll \mathbb{Q}$, on a Polish space $\mathcal{X}$ and its Borel σ-algebra Σ. $\mathcal{X}$ may be taken to be $\mathbb{R}^d$ or a (closed) subset thereof.

The relative entropy, or Kullback-Leibler divergence, from $\mathbb{P}$ to $\mathbb{Q}$ is defined as

$$R(\mathbb{P} \parallel \mathbb{Q}) = \mathbb{E}^{\mathbb{P}}\left[\log\left(\frac{d\,\mathbb{P}}{d\,\mathbb{Q}}\right)\right] = \int \log\left(\frac{d\,\mathbb{P}}{d\,\mathbb{Q}}\right) d\,\mathbb{P} \quad (1)$$

The Donsker-Varadhan representation for the relative en-



(a) Triangle (b) Ball

Figure 1: KNIFE training curves with error bars on 8-dimensional triangle and uniform ball datasets. It is observed that increasing the number of components M for KNIFE leads to overfitting in both datasets.

tropy states that

$$R(\mathbb{P} \parallel \mathbb{Q}) = \sup_{T:\mathcal{X}\to\mathbb{R}} \mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T], \tag{2}$$

where the supremum is taken over the class of continuous bounded functions, or the class of Borel-measurable functions from $\mathcal{X}$ to $\mathbb{R}$ (Donsker & Varadhan, 1983; Budhiraja & Dupuis, 2019).

Assume that $\mathbb{P}, \mathbb{Q} \ll \lambda$, where λ is the Lebesgue measure. Then they have densities p, q. The definition of the differential entropy of $\mathbb{P}$ is

$$H(\mathbb{P}) = -\mathbb{E}^{\mathbb{P}}\left[\log \frac{d\,\mathbb{P}}{d\lambda}\right] = -\mathbb{E}^{\mathbb{P}}\left[\log p\right]. \tag{3}$$

3.2. Limitations of existing entropy estimation approaches

Given n samples $x_1, x_2, ..., x_n$ from the data distribution $p(x)$ an oracle estimator of entropy is the Monte Carlo estimator: $\hat{H}_{Oracle}(x) = n^{-1}\sum_{i=1}^{n} -\log p(x_i)$. However, in practice, we rarely know the true data density $p(x)$. Therefore, a large body of literature focuses on finding an accurate plug-in estimator of the density $\hat{p}(x)$ which can replace $p(x)$ in the oracle estimator.

The Parzen window density estimation for $p(x)$ amounts to specifying a bandwidth h and then letting $\hat{p}(x) = \frac{1}{nh}\sum_{i=1}^{n}\phi(\frac{x-x_i}{h})$, where ϕ is the standard isotropic normal density in $\mathbb{R}^d$. In (Schraudolph, 2004) this is improved by parametrizing the positive definite covariance matrices of the individual kernels by their lower Cholesky factors, allowing for gradient-based learning of them.

In (Pichler et al., 2022), the authors propose a plug-in density estimator, KNIFE, that is a learnable mixture of M multivariate Gaussian kernels. However, the number of mixture components, M used in KNIFE, is treated as a hyperparameter to be optimized using discrete grid search. In our experiments, we found that training KNIFE requires delicate tuning of the number of components M. In the training, KNIFE minimizes the cross-entropy loss function which is an upper bound to the true entropy i.e. $H(\mathbb{P}) \leq \mathcal{L}_{\text{KNIFE}}$ (Eq. 6 in (Pichler et al., 2022)). In Fig. 1, we present the validation (solid) and training (dotted) loss curves by fitting KNIFE on 8-dimensional triangle and uniform ball (see Sec. D.3) datasets. We observe that increasing M leads to overfitting on both datasets. In addition, the best estimate of the entropy on the validation set lies significantly away from the true entropy (green dotted line). The estimation errors are even worse when the data dimension 20 (see Fig. 12 in Appendix C). This behavior is in line with the data inefficiency of the simpler kernel density estimators in moderately large dimensions (Wasserman, 2004). To this end, we propose `REMEDI` that applies a correction to any simple learnable base density. We show that the `REMEDI` estimator is theoretically consistent and is tractable using existing optimization tools.

3.3. The `REMEDI` approach

In this section, we derive a bound that is tight to the true entropy $H(\mathbb{P})$ and can be parametrized for efficient optimization. Using Eq. (2), we have,

$$\begin{aligned} H(\mathbb{P}) &= -\mathbb{E}^{\mathbb{P}}\left[\log p\right] \\ &= -\mathbb{E}^{\mathbb{P}}\left[\log q\right] - \mathbb{E}^{\mathbb{P}}\left[\log (p/q)\right] \\ &= -\mathbb{E}^{\mathbb{P}}\left[\log q\right] - R(\mathbb{P} \parallel \mathbb{Q}) \\ &= -\mathbb{E}^{\mathbb{P}}\left[\log q\right] - \sup_{T:\mathcal{X}\to\mathbb{R}} (\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]) \end{aligned} \tag{4}$$

The first term in Eq. (4) is known as the cross-entropy from $\mathbb{P}$ to $\mathbb{Q}$. Working with Eq. (4), we have,

$$H(\mathbb{P}) \leq -\mathbb{E}^{\mathbb{P}}\left[\log q\right] - \left(\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]\right) =: \mathcal{L}_{\texttt{REMEDI}} \tag{5}$$

where equality in Eq. (5) holds when taking the infimum on the right-hand side with respect to T. It can be shown that minimizing Eq. (5) is equivalent to minimizing a cross-entropy loss between the Gibbs density $\tilde{p}$ induced by T and true density p. Proposition B.1 in the appendix provides more insight into this bound in Eq. (5).

Proposition B.1 implies that, (i) for any choice of $\mathbb{Q}$ optimizing RHS of Eq. 5 is equivalent to optimizing a cross-entropy loss between $\tilde{p}$ and true density p, and (ii) the optimal solution is reached when the associated density is equal to the true density. Therefore, we use Eq. (5) as a loss function, denoted by $\mathcal{L}_{\texttt{REMEDI}}$, which we minimize to estimate $H(\mathbb{P})$.

3.4. Algorithm

Given a set of n samples $\{x_i\}_{i=1}^{n}$ from the data distribution $\mathbb{P}$ and m independent samples $\{\tilde{x}_j\}_{j=1}^{m}$ from $\mathbb{Q}$, we can minimize $\mathcal{L}_{\texttt{REMEDI}}$ with standard gradient-based optimization

tools, by considering its empirical counterpart:

$$\hat{\mathcal{L}}_{\texttt{REMEDI}} = \underbrace{\frac{1}{n}\sum_{i=1}^{n} -\log q(x_i)}_{\hat{\mathcal{L}}_{\text{KNIFE}}} - \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} T(x_i) - \log\left(\frac{1}{m}\sum_{i=1}^{m} e^{T(\tilde{x}_i)}\right)\right)}_{\hat{\mathcal{L}}_{\text{DV}}}. \tag{6}$$

The loss function $\hat{\mathcal{L}}_{\texttt{REMEDI}}$ has two components, (i) the cross-entropy loss $\hat{\mathcal{L}}_{\text{KNIFE}}$ for training a KNIFE base distribution, and (ii) the Donsker-Varadhan loss $\hat{\mathcal{L}}_{\text{DV}}$. Here, $\tilde{x}_1, ..., \tilde{x}_n$ represents n samples from the base distribution. Although we chose KNIFE as the base distribution $\mathbb{Q}$ it can be replaced by any distribution with a tractable likelihood and efficient sampling scheme. In our experiments, we found that KNIFE with a few components is a good candidate for the base distribution. Following (Belghazi et al., 2018) we parametrize T as a neural network, which provides a flexible class for function approximation (Hornik et al., 1989).

We present the implementation details of REMEDI in Algorithm 1. The parameters $\theta_{\text{KNIFE}}, \phi_T$ denote the parameters of KNIFE and the neural network T respectively. Note that, optimizing $\hat{\mathcal{L}}_{\text{DV}}$ using naive gradient routines in, e.g., Pytorch (Paszke et al., 2019) introduces bias in the stochastic gradient estimation of the log of expectation term in the DV loss (see Eq.12 in (Belghazi et al., 2018)). To alleviate this we use large batch-sizes and exponential moving average (Belghazi et al., 2018) in our experiments.

Algorithm 1 REMEDI algorithm

Draw n minibatch sample $(x_1, ..., x_n)$ from $p(x)$
Initialize $(\theta_{\text{KNIFE}}, \phi_T)$
for k_1 epochs **do**
  Update θ_{KNIFE} optimizing cross-entropy loss using the samples
end for
Draw m samples $(\tilde{x}_1, ..., \tilde{x}_m)$ from KNIFE
for k_2 epochs **do**
  Use $(x_1, ..., x_n)$ and $(\tilde{x}_1, ..., \tilde{x}_m)$ to update ϕ_T by optimizing the loss $\hat{\mathcal{L}}_{\text{DV}}$
end for
return $\hat{\theta}_{\text{KNIFE}}, \hat{\phi}_T$

3.5. Theoretical results

To assess the validity of the REMEDI approach to entropy estimation, we show in Appendix A that, under weak conditions on $\mathbb{P}$ and $\mathbb{Q}$, the estimator satisfies an appropriate form of consistency. This requires considering the limit of increasing both the amount of samples n taken from $\mathbb{P}$ and m, taken from $\mathbb{Q}$. By the law of large numbers, $\hat{\mathcal{L}}_{\text{KNIFE}} = n^{-1}\sum_{i=1}^{n} -\log q(x_i) \to C(\mathbb{P}\,||\,\mathbb{Q})$, and what needs to be shown for consistency is that, under appropriate conditions, $\hat{\mathcal{L}}_{DV} \to R(\mathbb{P}\,||\,\mathbb{Q})$. This is the done in Theorem A.15. Then $\hat{\mathcal{L}}_{\texttt{REMEDI}}$, by Eq. (6), converges to $H(\mathbb{P})$.

An analogous result for mutual information estimation is stated and proved in (Belghazi et al., 2018). However, they require that $\mathcal{X}$ be compact. This is much too strong for our purposes, since any linear combination or convolution of Gaussians has infinite tails. For example (see Sec. 4), the two moons dataset has this property and, even more importantly, a common modeling choice for the latent space conditional distribution in the Information Bottleneck framework is to use a multivariate normal.

4. Experiments

In this section, we perform experiments to evaluate REMEDI in various unsupervised and supervised tasks ranging from differential entropy estimation, deep latent variable models, and generative models. We demonstrate using synthetic and natural datasets that REMEDI can easily be applicable in these tasks.

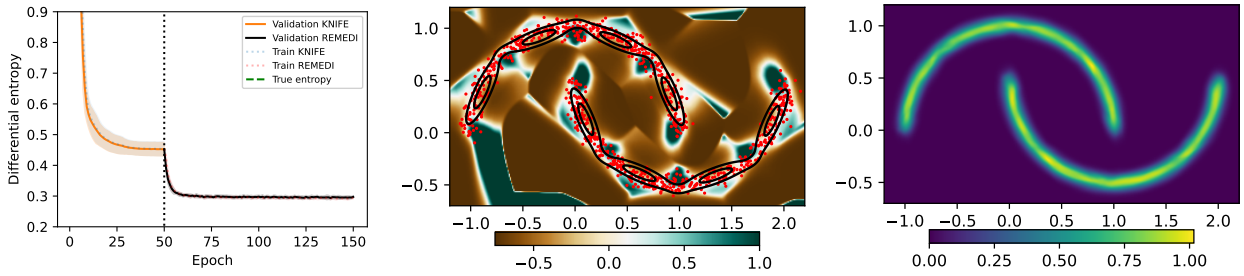
4.1. Entropy estimation

Entropy estimation is an important step in many real-world datasets. However, complex intrinsic features of the data distribution, e.g. high-dimensionality, and multi-modality can make the task challenging. In this section, we apply REMEDI for the entropy estimation of two such complex datasets: (1) Two moons, and (2) Triangle dataset (Pichler et al., 2022) (see Appendix C for additional benchmarks). On these datasets, we compare REMEDI with KNIFE (Pichler et al., 2022) the current state-of-the-art approach for entropy estimation. Furthermore, we analyze the inner workings of how REMEDI corrects the base distribution.

4.1.1. Two moons

We apply our method to the popular two moons dataset from Scikit-learn (Pedregosa et al., 2011). Here for illustration, we use 8 components for the KNIFE base-distribution. In Fig. 2a we see that the training process for KNIFE planes out around its final estimate at around 0.45. Here, the REMEDI takes over and manages to push the estimate down below 0.30. This is close to the true entropy of the dataset; here, the entropy offers no closed-form expression but an oracle computation, using kernel density estimation on one million samples, yields a value close to 0.29, see Appendix D.3.2.

To facilitate a better understanding of the learning mechanism of REMEDI, Fig. 2b shows a visualization of the correction given by the method, compared to the contours of KNIFE, i.e. $\mathbb{Q}$, and samples from $\mathbb{P}$. We can see that



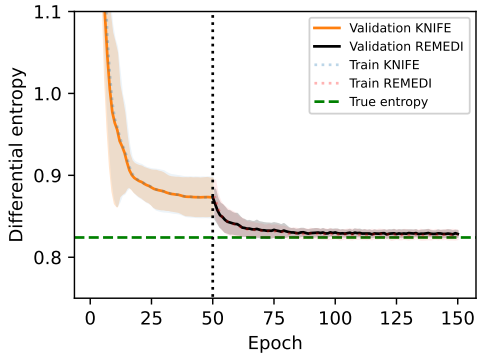
(a) Training curve (b) T (image) vs. q (contours). (c) Unnormalized density $qe^T(x)$

Figure 2: Results on two moons dataset. In the middle we see what direction (positive or negative) REMEDI affects the base distribution. To the right is the unnormalized distribution implied by $q(x)e^{T(x)}$.

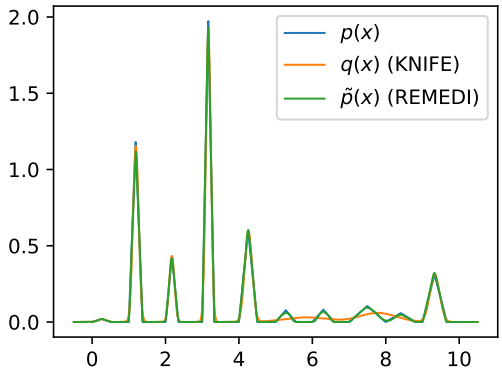
REMEDI reinforces low-probability regions of the data, with respect to $\mathbb{Q}$, by putting higher relative corrections there. This is in line with the interpretation of T as learning the (log) unnormalized Radon-Nikodym derivative $\frac{d\mathbb{P}}{d\mathbb{Q}}$. Exploiting this notion, we provide the corresponding learned density plot in Fig. 2c.

4.1.2. Triangle mixture

In this section, we compare REMEDI to KNIFE on the triangle mixture dataset (Pichler et al., 2022). This dataset consists of samples from a mixture of triangle distributions resulting in many modes. Applying REMEDI to the one-dimensional version of the dataset yields Fig. 3a. Since the normalizing constant is easily computed in one dimension, the normalized learned distribution along with the true density, is shown in Fig. 3b. The high-dimensional versions of this dataset constitute challenging targets for our model. In (Pichler et al., 2022), the authors showcase decent results in 8 dimensions, although still missing the target by several integer points. This is the eight-fold product distribution of one-dimensional two-component distributions, resulting in a 2^8-modal distribution, see Appendix D.3. The true entropy if this distribution is 2.585. Applying a 16-component KNIFE estimator to this task results in an estimates of around 4.36, where the improvement stops. Adding REMEDI on top of this model improves this estimate considerably to 3.08, see Fig. 4 for the full behavior. Both estimates can be further improved by adding more components to the KNIFE-based $\mathbb{Q}$. In Appendix C, we show that KNIFE with an increasing number of components easily fails before reaching a good estimate, due to overfitting issues, while using fewer with a REMEDI correction is much more efficient.



(a) Training curve with standard deviations from 10 repetitions.



(b) Comparison of REMEDI estimated pdf with KNIFE.

Figure 3: Results on one-dimensional triangle dataset. On the bottom is the data distribution, compared to the density that KNIFE and REMEDI (up to a constant) has learned.

4.2. Application to Information Bottleneck

Information Bottleneck (IB) (Tishby et al., 2000) is a popular latent variable model that aims to learn a representation

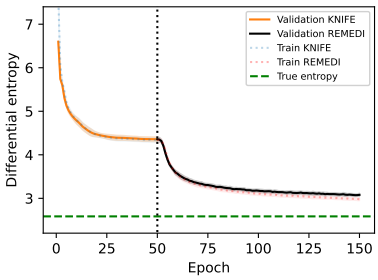


Figure 4: Training curve from the 8-dimensional triangle dataset. The horizontal dashed line indicates where the KNIFE training phase ends and REMEDI takes over.

Methods	Entropy estimate
KNIFE	4.3563 ± 0.0528
REMEDI	3.0798 ± 0.0368
True	2.5852

Table 1: Estimates on 8-dimensional triangle dataset.

Z from the input X that is maximally compressive of X and maximally predictive of the output Y. In practice, the learning problem is posed as a maximization problem (Tishby & Zaslavsky, 2015) of the IB loss function:

$$\mathcal{L}_{IB}(Z) = \mathrm{MI}(Z;Y) - \beta\,\mathrm{MI}(X;Z) \tag{7}$$

The loss function in Eq. (7) is in terms of the two mutual information quantities. $\mathrm{MI}(Z;Y)$ measures the predictive information contained in Z and $\mathrm{MI}(X;Z)$ measures the information about X contained in Z. Maximizing Eq. (7) implies compressing the inputs while simultaneously maximizing the predictive information in the representation Z. The Lagrange multiplier β serves to control the amount of compression.

The Information Bottleneck has been applied in deep learning (Alemi et al., 2016; Achille & Soatto, 2018) to learn compressed representation from high-dimensional inputs e.g. images and texts that are highly predictive of the low-dimensional targets e.g. labels. The usual practice is to parametrize the representation Z by a stochastic encoder $p_\psi(z|x)$. Following (Alemi et al., 2016), we assume $p_\psi(z|x)$ to be a multivariate Gaussian distribution with mean $\mu(X)$ and a diagonal covariance matrix $\Sigma(X)$. However, it is infeasible to calculate the mutual information $\mathrm{MI}(X;Z)$, especially for complex datasets. Therefore, there are many methods proposed in the literature for accurate estimation of $\mathrm{MI}(X;Z)$ ranging from parametric (Alemi et al., 2016), non-parametric (kernel density based) (Kolchinsky et al., 2019; Pichler et al., 2022), and adversarial f-divergences (Belghazi et al., 2018; Zhai & Zhang, 2022).

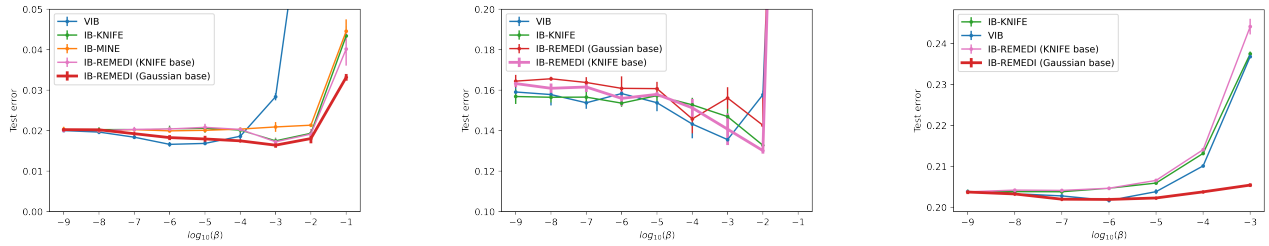
The application of REMEDI to estimate $\mathrm{MI}(X;Z)$ is straightforward. From the decomposition $\mathrm{MI}(X;Z) = H(Z) - H(Z|X)$ and the fact that we can analytically derive $H(Z|X)$, estimation of the mutual information boils down to accurately estimating the entropy $H(Z)$. We follow the Algorithm 1 to apply REMEDI by replacing $p(x)$ with the stochastic encoder $p_\psi(z|x)$. In applying REMEDI, we assume a coordinate-wise independent isotropic Gaussian and a 10-component KNIFE as the choices for the base distribution. Our choice for the number of components in KNIFE is based on the number of classes in the data (see Fig.2 in (Alemi et al., 2016)) and available GPU memory to fit the parameters. Additional details about the implementation are provided in the Appendix D.2.

We perform experiments to compare REMEDI against state-of-the-art mutual information estimation approaches: VIB (Alemi et al., 2016), KNIFE (Pichler et al., 2022), and MINE (Belghazi et al., 2018). In our experiments, we followed the open-source implementation of these approaches (the code for our implementation is provided in the supplementary materials). We evaluate REMEDI and other approaches in image classification tasks on MNIST, CIFAR-10, and ImageNet datasets. In our experiments, we use the standard training and test splits for these datasets and closely follow the network architectures from (Samaddar et al., 2023). We run all methods for three different seeds. See Appendix D.2 for additional details.

Classification accuracy: In this section, we compare the classification accuracy of each method on the test splits of the three datasets. For evaluation metrics, we compute the test set classification error and the log-likelihood ($\propto \mathrm{MI}(Z;Y)$). We present the "1 shot eval" from (Alemi et al., 2016) where we take one sample from the encoder and pass it to the decoder for prediction.

In Fig. 5, we plot the classification errors against the Lagrange multiplier β for the three datasets. On MNIST and ImageNet, we observe that REMEDI consistently exhibits the lowest classification errors for most of the β values. On CIFAR10, all methods are seen to perform similarly, however, REMEDI exhibits the lowest classification error across the β values. Across the datasets, we observe improvements in accuracy by REMEDI, especially around the values where the classification errors start to increase. This region is interesting because it contains the *minimum necessary information* (MNI) point (Fischer, 2020) where the model retains the necessary information from the inputs to predict the target and minimizes the redundant information from the inputs. Additionally, we present similar plots based on log-likelihood in Appendix G.1 where the conclusions remain unchanged.

In Table 2, we present the best test accuracy and corresponding log-likelihood across β for all the methods on the three



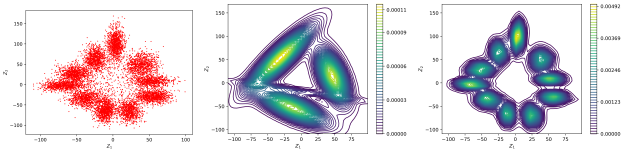
(a) MNIST (b) CIFAR10 (c) ImageNet

Figure 5: Plot showing test error of the Information Bottleneck methods vs β on benchmark image classification datasets (error bars represent standard deviations). For most β values, consistently REMEDI performs better than other methods on MNIST and ImageNet. On CIFAR10, the classification errors are similar for all the methods. However, REMEDI exhibits the lowest classification error across the β values.

datasets. On MNIST and CIFAR10, REMEDI performs better than all methods in terms of the metrics. On ImageNet, REMEDI performs similarly to the VIB method. However, it learns significantly less information about the inputs measured (160.67 bits) by $\widehat{\mathrm{MI}}(X;Z)$ than the VIB (237.71 bits). We note that on CIFAR10 and ImageNet we found stability issues with the MINE implementation perhaps due to the exponential term in the Donsker-Varadhan lower bound (McAllester & Stratos, 2020).

In these experiments, we present results of two different base distributions for REMEDI. We found that choosing an independent isotropic Gaussian base distribution is computationally more efficient than choosing a trainable base distribution such as KNIFE. Additionally, our results indicate that choosing a Gaussian base distribution increases the classification accuracy of REMEDI especially on MNIST and ImageNet.

Analysis of the latent space To understand the learning mechanism of REMEDI, we try to visualize the Information Bottleneck latent space. For the sake of visualization, we train an IB model using REMEDI with 2-d latent space on MNIST. Training using Algorithm 1 involves learning the KNIFE base distribution and the corrections applied by the REMEDI. Therefore, we analyze how each of the two components captures the marginal distribution induced by the IB latent space.



(a) Encoder samples (b) KNIFE (c) REMEDI

Figure 6: REMEDI marginal distribution of 2-d latent space on MNIST.

We observe that the samples from the marginal distribution of Z exhibit a clustering pattern in Fig. 6a where we can identify 10 cluster components. These clusters represent the 10 MNIST digits (Alemi et al., 2016). The KNIFE base distribution, although having 10 mixture components, learns mixture distribution in Fig. 6b with three identifiable modes. We note that KNIFE struggles with identifying clusters that are overlapping. In Fig. 6c, we observe that applying the REMEDI based corrections helped improve the density estimation significantly. Accurate marginal density estimation in Fig. 6c implies the latent space is properly regularized. In Appendix G.2, we perform this analysis throughout the training process from which the conclusions were similar. A similar analysis is also presented for CIFAR-10 in the Appendix G.3. This perhaps explains the improvement in accuracy shown by REMEDI over KNIFE in classification tasks on MNIST and CIFAR-10.

4.3. Generative Models

One useful by-product of fitting a neural network T is that we have implicitly defined a density $\tilde{p}(x) := \frac{1}{C} q(x) e^{T(x)}$, with an unknown C, that approximates $p(x)$. This can be used in two common strategies for Monte Carlo sampling: rejection sampling and sampling based on Langevin dynamics. These two strategies and corresponding experiments are briefly explained in the following subsections.

4.3.1. Rejection Sampling

Rejection sampling to sample from $\tilde{p}$, using the density q of $\mathbb{Q}$ for comparisons, amounts to the following procedure. First, draw a sample X from $\mathbb{Q}$. This sample is then accepted or rejected based on comparing $\tilde{p}(X)$ and $q(X)$. More concretely, if there is a constant $\tilde{C} \in (1, \infty)$ such that $\tilde{p}(x)/\tilde{C}q(x) \leq 1$ for all x such that $\tilde{p}(x) > 0$, then X is accepted with probability $\tilde{p}(X)/\tilde{C}q(X)$.

From the specific form of $\tilde{p}$, we see that rejection sampling

Methods	MNIST		CIFAR-10		ImageNet	
	Acc %	LL	Acc %	LL	Acc %	LL
KNIFE (Pichler et al., 2022)	98.25 (0.093)	3.22 (0.005)	86.73 (0.394)	2.57 (0.007)	79.62 (0.053)	8.55 (0.006)
MINE (Belghazi et al., 2018)	98.01 (0.040)	3.22 (0.001)	-	-	-	-
VIB (Alemi et al., 2016)	98.34 (0.070)	**3.24** (0.0003)	86.45 (0.135)	2.60 (0.013)	**79.83** (0.061)	**8.65** (0.003)
REMEDI (Gaussian base)	**98.36** (0.076)	**3.24** (0.003)	85.74 (1.671)	**2.61** (0.056)	79.81 (0.047)	8.63 (0.002)
REMEDI (KNIFE base)	98.28 (0.105)	3.22 (0.004)	**86.99** (0.176)	2.57 (0.005)	79.61 (0.015)	8.55 (0.006)

Table 2: Comparison of the best test accuracy and corresponding log-likelihood across the β values for MNIST, CIFAR-10, and ImageNet (standard deviations in the parenthesis). In terms of the classification accuracy, on ImageNet REMEDI performs close to the best performing method. However, on MNIST and CIFAR10 REMEDI shows improvement over the state-of-the-art IB method.

here requires a constant $\tilde{C}$ such that $e^{T(x)} \leq \tilde{C}$. For such a constant, a sample X from $\mathbb{Q}$ is accepted with probability $\phi(X)$, where $\phi(x) := \frac{e^{T(x)}}{\tilde{C}}$.

Revisiting the two moons dataset, using the learned T, rejection sampling is performed by taking $\tilde{C}$ as the maximum of e^T over the dataset, plus a small margin. In the experimental run illustrated in Fig. 7, 10000 samples from $\mathbb{Q}$ were used, out of which 2012 samples were accepted.

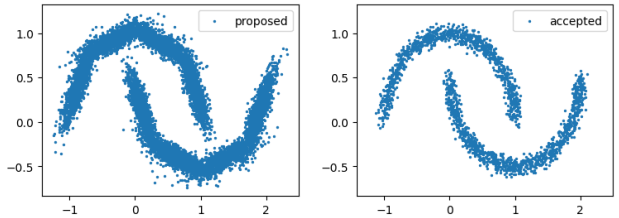


Figure 7: Left: 10000 proposals from $\mathbb{Q}$. Right: 2012 accepted samples.

4.3.2. Stochastic Differential Equations

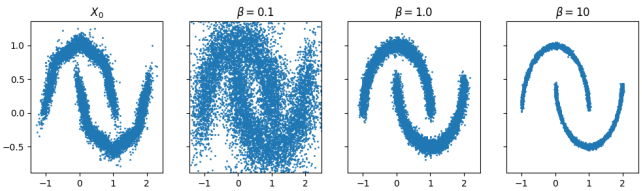


Figure 8: $\mathbb{Q}$-samples X_0 (leftmost) and X_{t_H} after simulating Eq. (8) with different β.

An alternative to rejection sampling, that does not require sampling a large number of random variables from $\mathbb{Q}$, is to use Langevin dynamics based on q and T to sample from $\tilde{p}$. This strategy is based on the stochastic differential equation

$$dX_t = -\nabla V(X_t)dt + \sqrt{2\beta^{-1}}dW_t, \quad X_0 = x_0, \tag{8}$$

where the drift is defined by $V(x) = -(\log q(x) + T(x))$; $\beta \in (0, \infty)$ is referred to as the inverse temperature. It is straightforward to show that the invariant distribution of $X = \{X_t\}_{t \geq 0}$ is given by

$$\frac{1}{Z} e^{-\beta V(x)}, \tag{9}$$

where Z is the normalizing constant. Note that for the choice $\beta = 1$, the exponent is precisely $-V(x) = (\log q(x) + T(x))$ and the density of the invariant distribution corresponds precisely to $\tilde{p}$. Based on this, the Langevin dynamics Eq. (8) can be used to obtain samples from $\tilde{p}$.

Fig. 8 illustrates the outcome of the following experiment: Taking $X_0 \sim \mathbb{Q}$, we set the time horizon to $t_H = 0.1$, and simulate Eq. (8) using the Euler-Maruyama method with a discretization parameter $\Delta t = 0.001$ (i.e., 100 time steps).

5. Conclusion / Future work

In this paper, we introduce REMEDI, a mixture model corrective transformation approach that combines recent plug-in based entropy estimators with Donsker-Varadhan based objectives, to improve the estimation of information theoretic quantities. We demonstrate the applicability of our approach to a variety of tasks ranging from entropy estimation, supervised learning, to generative models. We theoretically show the consistency of the REMEDI estimator under non-compactly supported data distributions, which is required by our framework. Using a range of benchmark datasets, we show that REMEDI outperforms the state-of-the-art entropy estimation approaches. We discuss the application of REMEDI to the Information Bottleneck. We show that using this approach improves the performance of Information Bottleneck in classification tasks on MNIST, CIFAR-10, and ImageNet compared to the current state-of-the-art. In addition, we show proof-of-concept that the REMEDI framework can be applied to generative tasks, using approaches such as rejection sampling and Langevin diffusion sampling.

It remains to be seen how this method performs in very high dimension, for example, what results can be acquired

when performing entropy estimation on image data such as MNIST. When using a uniform base distribution, (Park & Pardalos, 2021) finds that their method has some interpretable results on MNIST, but it seems that it only weakly learns to distinguish between in-distribution and out-of-distribution data. Most likely, the sample efficiency is much too low to learn much about the dataset. In these settings, instead using Gaussian mixture models such as KNIFE, copulas, or even normalizing flows may lead to base distributions that provide samples close to the low-dimensional data manifold, while also having tractable density functions. Also, following the successful application of the Langevin diffusion approach presented here, the connection to the recently popularized score-based diffusion models (Song et al., 2020b) should be explored.

Acknowledgements

The computations were enabled by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre, and the computational resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357, Laboratory Computing Resource Center (LCRC) at the Argonne National Laboratory. AS and SM were supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, under contract DE-AC02-06CH11357 and Advanced Scientific Computing Research, through the SciDAC-RAPIDS2 institute under Contract DE-AC02-06CH11357. VN and PN were supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. PN was also supporrted in part by the Swedish Research Council (VR-2018-07050, VR-2023-03484).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Achille, A. and Soatto, S. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2897–2905, 2018.

Aharoni, Z., Tsur, D., and Permuter, H. H. Density estimation of processes with memory via donsker vardhan. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 330–335. IEEE, 2022.

Ahmad, I. and Lin, P.-E. A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.). *IEEE Transactions on Information Theory*, 22(3):372–375, 1976.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Beirlant, J., Dudewicz, E. J., Györfi, L., Van der Meulen, E. C., et al. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Budhiraja, A. and Dupuis, P. Analysis and approximation of rare events. *Representations and Weak Convergence Methods. Series Prob. Theory and Stoch. Modelling*, 94, 2019.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Donsker, M. D. and Varadhan, S. S. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on pure and applied mathematics*, 36 (2):183–212, 1983.

Faivishevsky, L. and Goldberger, J. Ica based on a smooth estimation of the differential entropy. *Advances in neural information processing systems*, 21, 2008.

Fischer, I. The conditional entropy bottleneck. *Entropy*, 22 (9):999, 2020.

Gao, W., Oh, S., and Viswanath, P. Demystifying fixed k-nearest neighbor information estimators. *IEEE Transactions on Information Theory*, 64(8):5629–5661, 2018.

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Jaynes, E. T. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

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

Kolchinsky, A., Tracey, B. D., and Wolpert, D. H. Nonlinear information bottleneck. *Entropy*, 21(12):1181, 2019.

Kozachenko, L. F. and Leonenko, N. N. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

McAllester, D. and Stratos, K. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pp. 875–884. PMLR, 2020.

Park, S. and Pardalos, P. M. Deep data density estimation through donsker-varadhan representation. *arXiv preprint arXiv:2104.06612*, 2021.

Parzen, E. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3): 1065–1076, 1962.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Pichler, G., Colombo, P. J. A., Boudiaf, M., Koliander, G., and Piantanida, P. A differential entropy estimator for training neural networks. In *International Conference on Machine Learning*, pp. 17691–17715. PMLR, 2022.

Rodríguez Gálvez, B., Thobaben, R., and Skoglund, M. The convex information bottleneck Lagrangian. *Entropy*, 22(1):98, Jan. 2020. ISSN 1099-4300. doi: 10.3390/e22010098. URL http://dx.doi.org/10.3390/e22010098.

Rosenblatt, M. Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, pp. 832–837, 1956.

Samaddar, A., Madireddy, S., Balaprakash, P., Maiti, T., de los Campos, G., and Fischer, I. Sparsity-inducing categorical prior improves robustness of the information bottleneck. In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10207–10222. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/samaddar23a.html.

Sarra, L., Aiello, A., and Marquardt, F. Renormalized mutual information for artificial scientific discovery. *Physical Review Letters*, 126(20):200601, 2021.

Schraudolph, N. N. Gradient-based manipulation of nonparametric entropy estimates. *IEEE Transactions on Neural Networks*, 15(4):828–837, 2004.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Stimper, V., Liu, D., Campbell, A., Berenz, V., Ryll, L., Schölkopf, B., and Hernández-Lobato, J. M. normflows: A pytorch package for normalizing flows. *Journal of Open Source Software*, 8(86):5361, 2023. doi: 10.21105/joss.05361. URL https://doi.org/10.21105/joss.05361.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016. URL https://arxiv.org/abs/1602.07261.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pp. 1–5. IEEE, 2015.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

van de Geer, S. *Empirical processes in M-estimation*. Cambridge University Press, 2000.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Wasserman, L. *All of statistics: a concise course in statistical inference*, volume 26. Springer, 2004.

Zhai, P. and Zhang, S. Adversarial information bottleneck. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

A. Consistency of REMEDI

Techniques such as Parzen-window estimation and KNIFE are weaker than REMEDI but are known to be consistent under rather weak assumptions (Ahmad & Lin, 1976). This requires letting bandwidths go to zero and the number of mixture components to infinity, sometimes quickly becoming intractable in even moderate dimensions (Wasserman, 2004). To justify the method of letting ReLU-networks take some of the load and doing away with the infinitely many components, it is important to understand whether such a consistency result can be restated with the growing complexity of the function class of networks taking its place.

In this section, we will show that REMEDI is consistent, in the sense that it can approximate the relative entropy arbitrarily well with many samples. (Belghazi et al., 2018) show consistency of their mutual information estimator, under the (strong) assumption that the data and distributions are compactly supported. Such an assumption is undesirable in our case, since the data $\mathbb{P}$ may have infinite tails (e.g. two moons) and the base distributions $\mathbb{Q}$ are Gaussian (mixtures).

It is possible to adapt the results of (Belghazi et al., 2018), to our entropy estimation setting. However, the generalization to non-compact supports is non-trivial, and we provide it here for a specific class of neural networks, very similar to the ones that we have used.

Assume that $\mathbb{P}$ and $\mathbb{Q}$ are probability measures on $\mathbb{R}^d$ with the Borel σ-algebra $\mathcal{B}(\mathbb{R}^d)$, such that $\mathbb{P} \ll \mathbb{Q}$. Recall that the *relative entropy*, or KL-divergence, between $\mathbb{P}$ and $\mathbb{Q}$ is defined as

$$R(\mathbb{P} \parallel \mathbb{Q}) \coloneqq \mathbb{E}^{\mathbb{P}}\left[\log \frac{d\,\mathbb{P}}{d\,\mathbb{Q}}\right]. \tag{10}$$

For a measurable function $T : \mathbb{R}^d \to \mathbb{R}$, define the functional

$$\tilde{R}(T) \coloneqq \mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T], \tag{11}$$

and with a slight abuse of notation let, for a family $\mathcal{F}$ of functions $T : \mathbb{R}^d \to \mathbb{R}$,

$$\tilde{R}(\mathcal{F}) \coloneqq \sup_{T \in \mathcal{F}} \tilde{R}(T) = \sup_{T \in \mathcal{F}} (\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]). \tag{12}$$

The motivation behind these definitions is the central Donsker-Varadhan representation of the relative entropy, here restated in terms of $\tilde{R}$.

Proposition A.1 (Donsker-Varadhan)**.** *Let $\mathbb{P} \ll \mathbb{Q}$, then*

$$R(\mathbb{P} \parallel \mathbb{Q}) = \sup_{T \in C_b} \tilde{R}(T) = \tilde{R}(C_b), \tag{13}$$

where C_b is the set of bounded continuous functions. The supremum may also be taken over bounded measurable functions. The supremum is attained at $T = \log \frac{d\,\mathbb{P}}{d\,\mathbb{Q}}$, which may not be bounded or continuous.

Let $\mathbb{P}_n$ and $\mathbb{Q}_m$ be the empirical measures over n and m independent samples from $\mathbb{P}$ and $\mathbb{Q}$, respectively. We define, for $n, m \in \mathbb{N}$, the empirical version of (12), based on $\mathbb{P}_n, \mathbb{Q}_m$, as

$$\tilde{R}_{n,m}(\mathcal{F}) \coloneqq \sup_{T \in \mathcal{F}} (\mathbb{E}^{\mathbb{P}_n}[T] - \log \mathbb{E}^{\mathbb{Q}_m}[e^T]). \tag{14}$$

Note that in contrast to $R(\mathbb{P} \parallel \mathbb{Q})$ and $\tilde{R}(\mathcal{F})$, the quantity $\tilde{R}_{n,m}(\mathcal{F})$ is random, and it is not clear in which fashion it converges to $\tilde{R}(\mathcal{F})$, if at all. This question falls under the field of empirical process theory (van de Geer, 2000). A few common results from it will provide conditions on the class $\mathcal{F}$ such that $\tilde{R}_{n,m}(\mathcal{F})$ converges to $\tilde{R}(\mathcal{F})$, almost surely. We will give an overview of these and show how a class of ReLU networks fulfills them, while also being expressive enough for Eq. (12) to approximate $R(\mathbb{P} \parallel \mathbb{Q})$ arbitrarily well.

A.1. Assumptions

As previously stated, we relax the assumption of (Belghazi et al., 2018) about compactness of the data space $\mathcal{X} \subseteq \mathbb{R}^d$. Thus we simply assume that $\mathbb{P}$ and $\mathbb{Q}$ are probability measures on $\mathbb{R}^d$, with $\mathbb{P} \ll \mathbb{Q}$ and $R(\mathbb{P} \parallel \mathbb{Q}) < \infty$. We will require finite first moments, i.e., for X the identity mapping on $\mathbb{R}^d$, we have (where $||.||_p$ denotes the p-norm)

$$\mathbb{E}^{\mathbb{P}}[||X||_2], \mathbb{E}^{\mathbb{Q}}[||X||_2] < \infty. \tag{15}$$

We also assume that $\mathbb{Q}$ is sub-Gaussian (see definition below), which holds for Gaussian mixtures, and that our parameter space Θ is a compact subset of $\mathbb{R}^N$ for some N.

With the compactness assumption on Θ, it will follow that there is a common global Lipschitz constant L_Θ (with respect to the Euclidean norm $||.||_2$) and a common bound A_Θ at zero for the neural network functions $\{T_\theta : \mathbb{R}^d \to \mathbb{R}\}_{\theta \in \Theta}$, when using for example ReLU activation functions. With these assumptions, the proof will follow from arguments from empirical process theory, taken from (van de Geer, 2000), and a deeper look at the class of ReLU networks, specifically those with two hidden layers.

Below, we state the definitions of sub-Gaussian and sub-exponential random variables, of which especially the latter is of crucial importance to the proof. We follow the presentation in (Vershynin, 2018). Note that these definitions do not require centeredness.

Definition A.2. We say that the random variable X in $\mathbb{R}$ is **sub-Gaussian** if there is a constant K_1 such that

$$\mathbb{P}(|X| \geq t) \leq e^{-\frac{t^2}{K_1^2}}, \quad \forall t \geq 0. \tag{16}$$

Further, we say that the random vector $X \in \mathbb{R}^d$ is sub-Gaussian if $u^T X$ is sub-Gaussian for all $u \in \mathbb{R}^d$ with $||u|| = 1$. Additionally the probability measure $\mathbb{P}$ on $\mathbb{R}^d$ is sub-Gaussian if $X = \mathrm{id}_{\mathbb{R}^d}$ is sub-Gaussian under $\mathbb{P}$.

Definition A.3. We say that the random variable X in $\mathbb{R}$ is **sub-exponential** if there is a constant K_1 such that

$$\mathbb{P}(|X| \geq t) \leq e^{-\frac{t}{K_1}}, \quad \forall t \geq 0. \tag{17}$$

The following properties of sub-Gaussian and sub-exponential distributions will become necessary.

Proposition A.4. *The following facts hold for sub-Gaussian and sub-exponential variables.*

1. *The components of a sub-Gaussian vector are sub-Gaussian.*
2. *A sum of sub-exponential random variables is sub-exponential.*
3. *X is sub-Gaussian if and only if X^2 is sub-exponential.*
4. *If X is a sub-Gaussian random vector in $\mathbb{R}^d$, then the random variable $||X||_2$, its Euclidean norm, is sub-Gaussian.*
5. *If X is a centered ($\mathbb{E}[X] = 0$), sub-Gaussian random variable on $\mathbb{R}$, then there exists a **variance proxy** $\sigma^2 \geq 0$ such that $\mathbb{E}[e^{tX}] \leq e^{\frac{\sigma^2 t^2}{2}}$ for all $t \in \mathbb{R}$. We say then say that $X \sim \mathrm{subG}(\sigma^2)$.*
6. *If $X_1 \sim \mathrm{subG}(\sigma_1^2)$ and $X_2 \sim \mathrm{subG}(\sigma_2^2)$ are independent, then $X_1 + X_2 \sim \mathrm{subG}(\sigma_1^2 + \sigma_2^2)$.*
7. *If $X \sim \mathrm{subG}(\sigma^2)$ and $a \in \mathbb{R}$, then $aX \sim \mathrm{subG}(a^2\sigma^2)$.*
8. *If $X \sim \mathrm{subG}(\sigma^2)$, then for any $a \geq 0$ we have $\mathbb{P}(X \geq a) \leq \exp\left(-\frac{a^2}{2\sigma^2}\right)$.*

Proof. Property 1 follows immediately from the definition by taking $u = (1, 0, 0, \ldots), (0, 1, 0, \ldots)$ etc. Properties 2, 3 and 5 are standard, see (Vershynin, 2018). Property 4 follows from Properties 1, 2 and 3 since $||X||_2^2 = \sum_{i=1}^d X_i^2$ is the sum of sub-exponential random variables and thus sub-exponential. The following computation shows Property 6, while Property 7 is immediate.

$$\mathbb{E}[e^{t(X_1+X_2)}] = \mathbb{E}[e^{tX_1}]\,\mathbb{E}[e^{tX_2}] \leq e^{\frac{\sigma_1^2 t^2}{2}} e^{\frac{\sigma_2^2 t^2}{2}} = e^{\frac{(\sigma_1^2+\sigma_2^2)t^2}{2}} \tag{18}$$

To obtain Property 8, we minimize a Chernoff bound; we have, for any $t \in \mathbb{R}$,

$$\mathbb{P}(X \geq a) = \mathbb{E}[\mathbb{1}_{X\geq a}] \leq \mathbb{E}\left[\mathbb{1}_{X\geq a}\,\frac{e^{tX}}{e^{ta}}\right] \leq \mathbb{E}\left[\mathbb{1}_{X\geq a}\,\frac{e^{tX}}{e^{ta}}\right] = \mathbb{E}\left[\frac{e^{tX}}{e^{ta}}\right] = e^{-ta}\,\mathbb{E}\left[e^{tX}\right] \leq e^{-ta+\frac{\sigma^2 t^2}{2}}. \tag{19}$$

Minimizing this over t yields $t = \frac{a}{\sigma^2}$, which upon reinsertion into Equation Eq. (19) gives

$$\mathbb{P}(X \geq a) \leq e^{-ta+\frac{\sigma^2 t^2}{2}}|_{t=\frac{a}{\sigma^2}} = e^{-\frac{a^2}{2\sigma^2}}. \tag{20}$$

□

A.2. Uniform laws of large numbers

An overview over necessary results from (van de Geer, 2000) is given below. Consider a class $\mathcal{G}$ of functions $g : \mathcal{X} \to \mathbb{R}$. Let the *empirical measure* be $\mathbb{P}_n := \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$, where X_i are i.i.d. draws from $\mathbb{P}$. We are interested in asserting the convergence of

$$\mathbb{E}^{\mathbb{P}_n}[g] = \frac{1}{n}\sum_{i=1}^n g(X_i) \tag{21}$$

toward $\mathbb{E}^{\mathbb{P}}[g]$ for all g in $\mathcal{G}$. This is equivalent to the convergence of the "worst" g with respect to the data, represented by $\mathbb{P}_n$, i.e. of $\sup_{g\in\mathcal{G}} |\,\mathbb{E}^{\mathbb{P}_n}[g] - \mathbb{E}^{\mathbb{P}}[g]|$ toward zero. This is a *uniform law of large numbers* (ULLN), which is called *strong* if the convergence happens almost surely.

Let $p \geq 1$ be a norm exponent; we will be interested in $p = 1$. The bracketing number and bracketing entropy are defined as follows.

Definition A.5. Let $\delta > 0$ and $\mathbb{P}$ be a probability measure on $\mathcal{X}$. Assume that there exists a collection of pairs of functions $\{[g_j^L, g_j^U]\}_{j=1}^N$ in $\mathcal{G}$ such that $||g_j^U - g_j^L||_p \leq \delta$ and for each $g \in \mathcal{G}$ there exist j such that $g_j^L \leq g \leq g_j^U$. The smallest such N (or ∞) is the **bracketing number** $N_{p,B}(\delta, \mathcal{G}, \mathbb{P})$, and the **bracketing entropy** is $H_{p,B}(\delta, \mathcal{G}, \mathbb{P}) = \log N_{p,B}(\delta, \mathcal{G}, \mathbb{P})$.

The following law of large numbers will be used to prove the consistency. Note that the proof given below differs slightly from that provided in (van de Geer, 2000).

Lemma A.6 (Lemma 3.1 in van de Geer)**.** *Assume that $H_{1,B}(\delta, \mathcal{G}, \mathbb{P}) < \infty$ for all $\delta > 0$. Then $\mathcal{G}$ satisfies the strong uniform law of large numbers (ULLN): if $\{X_i\}_{i=1}^n$ are i.i.d. samples from $\mathbb{P}$, then*

$$\sup_{g\in\mathcal{G}} \left|\mathbb{E}^{\mathbb{P}_n}[g] - \mathbb{E}^{\mathbb{P}}[g]\right| = \sup_{g\in\mathcal{G}} \left|\frac{1}{n}\sum_{i=1}^n g(X_i) - \mathbb{E}^{\mathbb{P}}[g]\right| \xrightarrow{a.s.} 0. \tag{22}$$

Proof. Use the empirical process notation $\mathbb{P}\,f = \mathbb{E}^{\mathbb{P}}[f(X)]$ for expectation with respect to $\mathbb{P}$. Then Eq. (22) is $\sup_{g\in\mathcal{G}} |\mathbb{P}_n\,g - \mathbb{P}\,g| \xrightarrow{a.s.} 0$. Take $\delta > 0$. Then we have the above finite collection $\{[g_j^L, g_j^U]\}_{j=1}^{N_\delta}$, and for each $g \in \mathcal{G}$ there exists $j = j(g)$ such that $g_j^L \leq g \leq g_j^U$. Then it holds for any g that $|\,\mathbb{P}_n\,g - \mathbb{P}\,g| \leq |\,\mathbb{P}_n\,g_{j(g)}^U - \mathbb{P}\,g| \vee |\,\mathbb{P}_n\,g_{j(g)}^L - \mathbb{P}\,g| = \max_{B\in\{L,U\}} |\,\mathbb{P}_n\,g_{j(g)}^B - \mathbb{P}\,g|$, and thus we have

$$\begin{aligned}
\limsup_{n\to\infty} \sup_{g\in\mathcal{G}} |\mathbb{P}_n\,g - \mathbb{P}\,g| &\leq \limsup_{n\to\infty} \sup_{g\in\mathcal{G}} \max_{B\in\{L,U\}} \left|\mathbb{P}_n\,g_{j(g)}^B - \mathbb{P}\,g\right| \\
&= \limsup_{n\to\infty} \sup_{g\in\mathcal{G}} \max_{B\in\{L,U\}} \left|(\mathbb{P}_n - \mathbb{P})g_{j(g)}^B + \mathbb{P}(g_{j(g)}^B - g)\right| \\
&\leq \limsup_{n\to\infty} \sup_{g\in\mathcal{G}} \max_{B\in\{L,U\}} \left|(\mathbb{P}_n - \mathbb{P})g_{j(g)}^B\right| + \delta \\
&= \delta + \limsup_{n\to\infty} \max_{j\in\{1,\ldots,N_\delta\}} \max_{B\in\{L,U\}} \left|(\mathbb{P}_n - \mathbb{P})g_j^B\right| \\
&= \delta \text{ a.s.},
\end{aligned} \tag{23}$$

where the last equality follows from the strong law of large numbers applied to each $g_j^U(X_i)$. Taking intersection of the events where the SLLN holds for each pair $(j, B) \in \{1, \ldots, N_\delta\} \times \{L, U\}$, yields an event of probability one. Since they are finitely many pairs, a simple argument shows that $(\max_{j\in\{1,\ldots,N_\delta\}} |(\mathbb{P}_n - \mathbb{P})g_j^U|)(\omega)$ converges for all ω in this intersection.

Since δ is arbitrarily small, we have that $\limsup_{n\to\infty} \sup_{g\in\mathcal{G}} |\mathbb{P}_n\, g - \mathbb{P}\, g| = 0$ almost surely, which is the claim. □

Now we need to prove that $\{T_\theta : \theta \in \Theta\}$ satisfies the bracketing condition, $H_{1,B}(\delta, \mathcal{G}, \mathbb{P}) < \infty$ for all $\delta > 0$, with respect to a sub-Gaussian $\mathbb{P}$.

Definition A.7. The **envelope** of $\mathcal{G}$ is the function

$$G(x) = \sup_{g\in\mathcal{G}} |g(x)|. \tag{24}$$

We say that $\mathcal{G}$ satisfies the **envelope condition** if $G \in L_1(\mathbb{P})$.

Recall that $\mathcal{F}_\Theta$ shares a common Lipschitz constant L_Θ and bound A_Θ at zero.

Lemma A.8. *$\mathcal{F}_\Theta = \{T_\theta : \theta \in \Theta\}$ satisfies the envelope condition with respect to $\mathbb{P}$.*

Proof. Note first that, for any x,

$$\begin{aligned} G(x) &= \sup_{\theta\in\Theta} |T_\theta(x)| \\ &= \sup_{\theta\in\Theta} |T_\theta(0) + (T_\theta(x) - T_\theta(0))| \\ &\leq \sup_{\theta\in\Theta} |T_\theta(0)| + \sup_{\theta\in\Theta} |T_\theta(x) - T_\theta(0)| \\ &\leq A_\Theta + L_\Theta ||x||_2. \end{aligned} \tag{25}$$

Thus we obtain, by finite first moment of $\mathbb{P}$,

$$\mathbb{E}^{\mathbb{P}}[G] \leq \mathbb{E}^{\mathbb{P}}[A_\Theta + L_\Theta ||X||_2] = A_\Theta + L_\Theta\, \mathbb{E}^{\mathbb{P}}[||X||_2] < \infty. \tag{26}$$

□

Lemma A.9. *$\mathcal{F}_\Theta^{\exp} = \{e^{T_\theta} : \theta \in \Theta\}$ satisfies the envelope condition with respect to $\mathbb{Q}$.*

Proof. As noted in proof of Lemma A.8, $\sup_{\theta\in\Theta} |T_\theta(x)| \leq A_\Theta + L_\Theta ||x||_2$. By monotonicity of the supremum and the map $x \mapsto e^x$, we have

$$G^{\exp}(x) = \sup_{\theta\in\Theta} |e^{T_\theta(x)}| \leq \sup_{\theta\in\Theta} e^{|T_\theta(x)|} = e^{\sup_{\theta\in\Theta} |T_\theta(x)|} \leq e^{A_\Theta + L_\Theta ||x||_2}. \tag{27}$$

Thus we obtain,

$$\mathbb{E}^{\mathbb{Q}}[G^{\exp}] \leq \mathbb{E}^{\mathbb{Q}}[e^{A_\Theta + L_\Theta ||X||_2}] = e^{A_\Theta}\, \mathbb{E}^{\mathbb{Q}}[e^{L_\Theta ||X||_2}], \tag{28}$$

which by sub-Gaussianity of $\mathbb{Q}$ is finite. □

Lemma A.10 (Lemma 3.10 in van de Geer)**.** *Assume that $\mathcal{G} = \{g_\theta : \theta \in \Theta\}$ satisfies the envelope condition with respect to $\mathbb{P}$. Assume also that (Θ, d) is a compact metric space, and that $\theta \mapsto g_\theta(x)$ is continuous for $\mathbb{P}$-almost all $x \in \mathcal{X}$. Then for any $\delta > 0$,*

$$H_{1,B}(\delta, \mathcal{G}, \mathbb{P}) < \infty. \tag{29}$$

Proof. See (van de Geer, 2000). □

We now show that $\{T_\theta : \theta \in \Theta\}$ satisfies a *strong uniform law of large numbers* (ULLN).

Theorem A.11. *Suppose each element T_θ of the family of neural network functions $\mathcal{F}_\Theta = \{T_\theta : \mathcal{X} \to \mathbb{R}\}_{\theta\in\Theta}$ depends continuously on θ, for any fixed $x \in \mathcal{X}$. With the standing assumptions on $\mathbb{P}$, $\mathbb{Q}$, and $\mathcal{F}_\Theta$ (see Section A.1), we have that*

$$\tilde{R}_{n,m}(\mathcal{F}_\Theta) \xrightarrow{a.s.} \tilde{R}(\mathcal{F}_\Theta), \tag{30}$$

as $n, m \to \infty$.

Proof. Using the identity $|\sup f - \sup g| \leq \sup |f - g|$, we have

$$\begin{aligned}
|\tilde{R}_{n,m}(\mathcal{F}_\Theta) - \tilde{R}(\mathcal{F}_\Theta)| &= \left| \sup_{T\in\mathcal{F}_\Theta} (\mathbb{E}^{\mathbb{P}_n}[T] - \log \mathbb{E}^{\mathbb{Q}_m}[e^T]) - \sup_{T\in\mathcal{F}_\Theta} (\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]) \right| \\
&\leq \sup_{T\in\mathcal{F}_\Theta} \left| (\mathbb{E}^{\mathbb{P}_n}[T] - \log \mathbb{E}^{\mathbb{Q}_m}[e^T]) - (\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]) \right| \\
&\leq \sup_{T\in\mathcal{F}_\Theta} \left| \mathbb{E}^{\mathbb{P}_n}[T] - \mathbb{E}^{\mathbb{P}}[T] \right| + \sup_{T\in\mathcal{F}_\Theta} \left| \log \mathbb{E}^{\mathbb{Q}_m}[e^T] - \log \mathbb{E}^{\mathbb{Q}}[e^T] \right| \\
&= \sup_{\theta\in\Theta} \left| \mathbb{E}^{\mathbb{P}_n}[T_\theta] - \mathbb{E}^{\mathbb{P}}[T_\theta] \right| + \sup_{\theta\in\Theta} \left| \log \mathbb{E}^{\mathbb{Q}_m}[e^{T_\theta}] - \log \mathbb{E}^{\mathbb{Q}}[e^{T_\theta}] \right|.
\end{aligned} \tag{31}$$

By Lemma A.8, we have that $\mathcal{F}_\Theta$ satisfies the envelope condition with respect to $\mathbb{P}$. Thus by Lemma A.10 and A.6, we have that $\mathcal{F}_\Theta$ satisfies the strong ULLN. Hence we have for the first term of Equation (31),

$$\sup_{\theta\in\Theta} \left| \mathbb{E}^{\mathbb{P}_n}[T_\theta] - \mathbb{E}^{\mathbb{P}}[T_\theta] \right| \xrightarrow{a.s.} 0. \tag{32}$$

The second term of Equation (31) is more difficult, because the logarithm does not have a Lipschitz constant on $(0, \infty)$. To obtain one, we need to know that $\mathbb{E}^{\mathbb{Q}_m}[e^{T_\theta}]$ and $\mathbb{E}^{\mathbb{Q}}[e^{T_\theta}]$ can be suitably bounded from below. Let again L_Θ be the common Lipschitz constant of $\mathcal{F}_\Theta$ and $A_\Theta := \sup_{\theta\in\Theta} |T_\theta(0)|$. Note first that

$$\mathbb{E}^{\mathbb{Q}}[e^{T_\theta}] \geq \mathbb{E}^{\mathbb{Q}}[e^{-A_\Theta - L_\Theta ||X||_2}] > 0. \tag{33}$$

Let $\mu_2^{\mathbb{Q}} := \mathbb{E}^{\mathbb{Q}}[||X||_2]$. For $b := \exp(-A_\Theta - L_\Theta(\mu_2^{\mathbb{Q}} + 1))$, we have on Θ,

$$\begin{aligned}
\text{Prob}\left(\mathbb{E}^{\mathbb{Q}_m}[e^{T_\theta}] < b, \forall\theta \in \Theta\right) &\leq \text{Prob}\left(\mathbb{E}^{\mathbb{Q}_m}\left[e^{-A_\Theta - L_\Theta ||X||_2}\right] < b\right) \\
\text{(Jensen's)} &\leq \text{Prob}\left(\exp(\mathbb{E}^{\mathbb{Q}_m}[-A_\Theta - L_\Theta ||X||_2]) < b\right) \\
&= \text{Prob}\left(\mathbb{E}^{\mathbb{Q}_m}[-A_\Theta - L_\Theta ||X||_2] < \log b\right) \\
&= \text{Prob}\left(\mathbb{E}^{\mathbb{Q}_m}[||X||_2] > -\frac{A_\Theta + \log b}{L_\Theta}\right) \\
&= \text{Prob}\left(\mathbb{E}^{\mathbb{Q}_m}\left[||X||_2 - \mu_2^{\mathbb{Q}}\right] > -\frac{A_\Theta + \log b}{L_\Theta} - \mu_2^{\mathbb{Q}}\right) \\
&= \text{Prob}\left(\mathbb{E}^{\mathbb{Q}_m}\left[||X||_2 - \mu_2^{\mathbb{Q}}\right] > 1\right).
\end{aligned} \tag{34}$$

Now, notice that $||X||_2 - \mu_2^{\mathbb{Q}}$ is centered sub-Gaussian under $\mathbb{Q}$, with some variance proxy $(\sigma^2)_2^{\mathbb{Q}}$. Then it holds that

$$\mathbb{E}^{\mathbb{Q}_m}\left[||X||_2 - \mu_2^{\mathbb{Q}}\right] = \frac{1}{m}\sum_{i=1}^{m}(||X||_2 - \mu_2^{\mathbb{Q}}) \sim \text{subG}\left(\frac{(\sigma^2)_2^{\mathbb{Q}}}{m}\right). \tag{35}$$

Returning to Equation (34), we get

$$\begin{aligned}\text{Prob}\left(\mathbb{E}^{\mathbb{Q}_m}\left[e^{T_\theta}\right] < b, \forall\theta\in\Theta\right) &\leq \text{Prob}\left(\mathbb{E}^{\mathbb{Q}_m}\left[||X||_2 - \mu_2^{\mathbb{Q}}\right] > 1\right)\\ &\leq \exp\left(-1^2/\left(2\frac{(\sigma^2)_2^{\mathbb{Q}}}{m}\right)\right)\\ &= \exp\left(-\frac{1}{2(\sigma^2)_2^{\mathbb{Q}}}m\right).\end{aligned} \tag{36}$$

where in the second inequality we used the concentration inequality, Property 8, from Proposition A.4, with $a = 1$. Since $\sum_{m=1}^{\infty}\exp\left(-\frac{1}{2(\sigma^2)_2^{\mathbb{Q}}}m\right) < \infty$, we get by the (first) Borel-Cantelli lemma that

$$\text{Prob}\left(\mathbb{E}^{\mathbb{Q}_m}\left[e^{T_\theta}\right] < b, \forall\theta\in\Theta, \text{infinitely often}\right) = 0. \tag{37}$$

In other words, on our probability space $(\Omega, \Sigma, \text{Prob})$, there is an event $\Omega' \in \Sigma$, with $\text{Prob}(\Omega') = 1$ and a function $M : \Omega \to \mathbb{N}$, such that $m \geq M(\omega) \implies \mathbb{E}^{\mathbb{Q}_m}\left[e^{T_\theta}\right](\omega) > b$ for $\omega \in \Omega'$. Let $\tilde{b} := b \wedge \mathbb{E}^{\mathbb{Q}}[e^{-A_\Theta - L_\Theta||X||_2}]$. The function $\log|_{[\tilde{b},\infty)}$ has a Lipschitz constant $1/\tilde{b}$. On Ω', we have that for all $m \geq M(\omega)$,

$$\sup_{\theta\in\Theta}\left|\log\mathbb{E}^{\mathbb{Q}_m}[e^{T_\theta}] - \log\mathbb{E}^{\mathbb{Q}}[e^{T_\theta}]\right|(\omega) \leq \frac{1}{\tilde{b}}\sup_{\theta\in\Theta}\left|\mathbb{E}^{\mathbb{Q}_m}[e^{T_\theta}] - \mathbb{E}^{\mathbb{Q}}[e^{T_\theta}]\right|(\omega). \tag{38}$$

$\mathcal{F}_\Theta^{\exp} = \{e^{T_\theta} : \theta \in \Theta\}$ satisfies the envelope condition with respect to $\mathbb{Q}$, by Lemma A.9. By Lemma A.10 and A.6, we have then that $\mathcal{F}_\Theta^{\exp}$ satisfies the strong ULLN. Thus we have that the left-hand side of Equation (38) goes to zero. Then, since Ω' is a probability one event, we have for the second term of Equation (31),

$$\sup_{\theta\in\Theta}\left|\log\mathbb{E}^{\mathbb{Q}_m}[e^{T_\theta}] - \log\mathbb{E}^{\mathbb{Q}}[e^{T_\theta}]\right| \xrightarrow{a.s.} 0. \tag{39}$$

Because the intersection of the events in Equation (32) and Equation (39) is a probability one event, we have

$$\sup_{\theta\in\Theta}\left|\mathbb{E}^{\mathbb{P}_n}[T_\theta] - \mathbb{E}^{\mathbb{P}}[T_\theta]\right| + \sup_{\theta\in\Theta}\left|\log\mathbb{E}^{\mathbb{Q}_m}[e^{T_\theta}] - \log\mathbb{E}^{\mathbb{Q}}[e^{T_\theta}]\right| \xrightarrow{a.s.} 0, \tag{40}$$

so by Equation (31), $\tilde{R}_{n,m}(\mathcal{F}_\Theta) \xrightarrow{a.s.} \tilde{R}(\mathcal{F}_\Theta)$. □

Universal approximation

We will also need to show that the we can use the universal approximation theorem even when $\mathcal{X}$ is non-compact. We begin by proving that, for a constant $M \in [0, \infty]$, we can truncate the output of a ReLU network with one hidden layer to $[-M, M]$, by adding another layer.

Lemma A.12. *Let $T_{\theta^1} : \mathbb{R}^d \to \mathbb{R}$ be a neural network with one hidden layer, with ReLU activation functions. Then, for $M \geq 0$ there exists a two hidden layer ReLU network T_{θ^2} which truncates T_{θ^1} to $[-M, M]$, i.e.,*

$$T_{\theta^2}(x) = M \wedge (-M \vee T_{\theta^1}(x)), \quad \forall x \in \mathbb{R}^d. \tag{41}$$

Proof. The output of the one hidden layer neural network, with width l, can be written

$$T_{\theta^1}(x) = \sum_{i=1}^{l} \alpha_i (\beta_i^T x + \gamma_i)^+. \tag{42}$$

Let the first hidden layer of T_{θ^2} be identical to the hidden layer in T_{θ^1}. Let its second hidden layer consist of two nodes with weights $\alpha_i^{(1)} = \alpha_i^{(2)} = \alpha_i$, and biases M and $-M$, respectively. Then the activations of the second hidden layer are

$$\begin{aligned} o^{(1)} &= \left(\sum_{i=1}^{l} \alpha_i (\beta_i^T x + \gamma_i)^+ + M \right)^+ = (T_\theta(x) - (-M))^+ \\ o^{(2)} &= \left(\sum_{i=1}^{l} \alpha_i (\beta_i^T x + \gamma_i)^+ - M \right)^+ = (T_\theta(x) - M)^+, \end{aligned} \tag{43}$$

as well as the bias node $o^{(0)} = 1$. Let the output layer consist of one node with weights $\lambda^{(0)} = -M$, $\lambda^{(1)} = 1$, and $\lambda^{(2)} = -1$. Then the output of the network is

$$\begin{aligned} T_{\theta^2}(x) &= \lambda^{(0)} + \lambda^{(1)} o^{(1)} + \lambda^{(2)} o^{(2)} \\ &= -M + (T_\theta(x) - (-M))^+ - (T_\theta(x) - M)^+ \\ &= M \wedge (-M \vee T_{\theta^1}(x)). \end{aligned} \tag{44}$$

□

Lemma A.13. *Let $\mathbb{P}$ be a probability measure on $\mathbb{R}^d$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be bounded and continuous. Then, for $\varepsilon > 0$, there exists a two hidden layer neural network $T_\theta : \mathbb{R}^d \to \mathbb{R}$ such that $\mathbb{E}^{\mathbb{P}}[|f - T_\theta|] < \varepsilon$. Further, if f is bounded by $M' - 1$, then T_θ can be chosen to be bounded by M'.*

Proof. Let $M = \sup_{x \in R^d} |f(x)| + 1$. Take a compact set $C = [-K, K]^d \subseteq \mathcal{X}$ such that $\mathbb{P}(C^c) < \frac{\varepsilon}{3M}$.

Now, by the universal approximation theorem (Hornik et al., 1989), take a one hidden layer neural network T_{θ^1} such that $\sup_{x \in C} |T_{\theta^1}(x) - f(x)| < \frac{\varepsilon}{3}$. Then we have that

$$\sup_{x \in C} |T_{\theta^1}(x)| \leq \sup_{x \in C} |T_{\theta^1}(x) - f(x)| + \sup_{x \in C} |f(x)| < \frac{\varepsilon}{3} + (M - 1) < M. \tag{45}$$

Assuming that T_{θ^1} uses ReLU activation functions (which is compatible by simple extension of the result in (Hornik et al., 1989)), by Lemma A.12 there is an alternative two hidden layer network T_θ such that $T_\theta(x) = M \wedge (-M \vee T_{\theta^1}(x))$. Note that $T_\theta(x) = T_{\theta^1}(x)$ for $x \in C$, since $T_{\theta^1}(x) \in [-M, M]$ for $x \in C$.

Then we have that $\sup_{x \in C} |T_\theta(x) - f(x)| < \frac{\varepsilon}{3}$ and $|T_\theta(x)| \leq M$ for all $x \in \mathbb{R}^d$. This gives us that

$$\begin{aligned} \mathbb{E}^{\mathbb{P}}[|f - T_\theta|] &= \mathbb{E}^{\mathbb{P}}[|f - T_\theta| \, \mathbb{1}_C] + \mathbb{E}^{\mathbb{P}}[|f - T_\theta| \, \mathbb{1}_{C^c}] \\ &\leq \mathbb{E}^{\mathbb{P}}[|f - T_\theta| \, \mathbb{1}_C] + \mathbb{E}^{\mathbb{P}}[|f| \, \mathbb{1}_{C^c}] + \mathbb{E}^{\mathbb{P}}[|T_\theta| \, \mathbb{1}_{C^c}] \\ &\leq \mathbb{E}^{\mathbb{P}}[(\varepsilon/3) \, \mathbb{1}_C] + \mathbb{E}^{\mathbb{P}}[M \, \mathbb{1}_{C^c}] + \mathbb{E}^{\mathbb{P}}[M \, \mathbb{1}_{C^c}] \\ &< \frac{\varepsilon}{3} + \frac{\varepsilon}{3} + \frac{\varepsilon}{3} \\ &= \varepsilon. \end{aligned} \tag{46}$$

The last statement of the lemma follows, since we can take $M = M'$. □

Theorem A.14. *Let $\mathbb{P}$ and $\mathbb{Q}$ be probability measures on $\mathcal{X}$, such that $\mathbb{P} \ll \mathbb{Q}$ and $R(\mathbb{P} \parallel \mathbb{Q}) < \infty$. Then, for any $\varepsilon > 0$, there exists a two hidden layer neural network $T_\theta : \mathcal{X} \to \mathbb{R}$ such that $R(\mathbb{P} \parallel \mathbb{Q}) - \tilde{R}(T_\theta) < \varepsilon$.*

Proof. By the Donsker-Varadhan representation theorem, there exists a continuous, bounded (say by $M' - 1$) function $T : \mathcal{X} \to \mathbb{R}$ such that $R(\mathbb{P} \parallel \mathbb{Q}) - \tilde{R}(T) < \varepsilon/2$. Let it be normalized such that $\mathbb{E}^{\mathbb{Q}}[e^T] = 1$ (note that $\tilde{R}(T + c) = \tilde{R}(T)$ for $c \in \mathbb{R}$). By Lemma A.13, with the probability measure $(\mathbb{P} + \mathbb{Q})/2$, there exists a two hidden layer neural network T_θ, bounded by M', such that $\mathbb{E}^{\mathbb{P}}[|T - T_\theta|] < \varepsilon/4$ and $\mathbb{E}^{\mathbb{Q}}[|T - T_\theta|] < e^{-M'}\varepsilon/4$. Then we have that

$$\begin{aligned}
\tilde{R}(T) - \tilde{R}(T_\theta) &= \mathbb{E}^{\mathbb{P}}[T - T_\theta] + (\log \mathbb{E}^{\mathbb{Q}}[e^{T_\theta}] - \underbrace{\log \mathbb{E}^{\mathbb{Q}}[e^T]}_{0}) \\
&\leq \mathbb{E}^{\mathbb{P}}[T - T_\theta] + (\mathbb{E}^{\mathbb{Q}}[e^{T_\theta}] - \underbrace{\mathbb{E}^{\mathbb{Q}}[e^T]}_{1}) \\
&\leq \mathbb{E}^{\mathbb{P}}[|T - T_\theta|] + \mathbb{E}^{\mathbb{Q}}[|e^{T_\theta} - e^T|] \\
&\leq \mathbb{E}^{\mathbb{P}}[|T - T_\theta|] + \mathbb{E}^{\mathbb{Q}}[e^{M'}|T_\theta - T|] \\
&< \frac{\varepsilon}{4} + e^{M'}e^{-M'}\frac{\varepsilon}{4} \\
&= \frac{\varepsilon}{2},
\end{aligned} \tag{47}$$

where the first inequality follows from the identity $\log x \leq x - 1$, and the third from the Lipschitz constant $e^{M'}$ of the exponential function restricted to $[-M', M']$. Combining the two inequalities, we have

$$R(\mathbb{P} \,||\, \mathbb{Q}) - \tilde{R}(T_\theta) = \Big(R(\mathbb{P} \,||\, \mathbb{Q}) - \tilde{R}(T_\theta)\Big) + \Big(\tilde{R}(T) - \tilde{R}(T_\theta)\Big) < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon. \tag{48}$$

□

The result

We are now able to state and prove our main theorem.

Theorem A.15. *REMEDI is strongly consistent, up to an arbitrarily small precision $\varepsilon > 0$.*

Proof. Choose the number of parameters N, (appropriately assigned to the first and second layer) and a compact set $\Theta \subset \mathbb{R}^N$, both large enough such that $|R(\mathbb{P} \parallel \mathbb{Q}) - \tilde{R}(\mathcal{F}_\Theta)| < \varepsilon/2$, by Theorem A.14. By the triangle inequality,

$$\begin{aligned}
|R(\mathbb{P} \parallel \mathbb{Q}) - \tilde{R}_{n,m}(\mathcal{F}_\Theta)| &= |R(\mathbb{P} \parallel \mathbb{Q}) - \tilde{R}(\mathcal{F}_\Theta) + \tilde{R}(\mathcal{F}_\Theta) - \tilde{R}_{n,m}(\mathcal{F}_\Theta)| \\
&\leq |R(\mathbb{P} \parallel \mathbb{Q}) - \tilde{R}(\mathcal{F}_\Theta)| + |\tilde{R}_{n,m}(\mathcal{F}_\Theta) - \tilde{R}(\mathcal{F}_\Theta)| \\
&< \frac{\varepsilon}{2} + |\tilde{R}_{n,m}(\mathcal{F}_\Theta) - \tilde{R}(\mathcal{F}_\Theta)|.
\end{aligned} \tag{49}$$

The second term converges to zero almost surely by Theorem A.11, and therefore the error converges to strictly less than ε almost surely. □

B. Justification of the loss function

In this section, we present the proof of Proposition B.1 which provides insights into the `REMEDI` loss function and it's connection to density estimation. We restate the following notations for their use in the proof. Let $\mathbb{P}$ and $\mathbb{Q}$ be the target and base distribution, respectively, defined on the sample space $\mathcal{X}$, with densities p, q, let X be a random variable with distribution $\mathbb{P}$, and T be a function from the class of continuous bounded functions, or the class of Borel-measurable functions from $\mathcal{X}$ to $\mathbb{R}$. Using the function T we define the Gibbs distribution $\mathbb{G}$ on the sample space of $\mathcal{X}$, which has the density $\tilde{p}(x) = \frac{q(x)e^{T(x)}}{\mathbb{E}^{\mathbb{Q}}[e^T]}$.

Proposition B.1. *Given a base distribution $\mathbb{Q}$, assuming $\mathbb{E}^{\mathbb{Q}}[e^T]$ exists, consider the following density defined by T,*

$$\tilde{p}(x) = \frac{q(x)e^{T(x)}}{\mathbb{E}^{\mathbb{Q}}[e^T]}.$$

Then,

(i) the right-hand side in Eq. (5) is equal to $-\mathbb{E}^{\mathbb{P}} \log \tilde{p}$

(ii) T^* is the solution of

$$\sup_{T:\mathcal{X}\to\mathbb{R}} \left(\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]\right),$$

if and only if the associated density $\tilde{p}^*$ satisfies $\tilde{p}^* = p$ for any base distribution $\mathbb{Q}$.

Proof of Proposition 3.1. We prove Proposition 3.1 in the following two parts.

1. *Showing $\mathcal{L}_{\texttt{REMEDI}} = -\mathbb{E}^{\mathbb{P}} \log \tilde{p}$:* Using the form of $\tilde{p}$ we can write,

$$\begin{aligned}
-\mathbb{E}^{\mathbb{P}} \log \tilde{p}(X) &= -\mathbb{E}^{\mathbb{P}} \log \frac{q(X)e^{T(X)}}{\mathbb{E}^{\mathbb{Q}}[e^T(X)]} \\
&= -\mathbb{E}^{\mathbb{P}} \log q(X) - \left(\mathbb{E}^{\mathbb{P}} T(X) - \log \mathbb{E}_{\mathbb{Q}}\, e^{T(X)}\right) \\
&= \mathcal{L}_{\texttt{REMEDI}}
\end{aligned}$$

2. *Showing that optimal T^* is achieved if and only if $\tilde{p}^* = p$ for any $\mathbb{Q}$:* For any function $T : \mathcal{X} \to \mathbb{R}$, and corresponding, previously defined, distribution $\mathbb{G}$, we expand the following relative entropy,

$$\begin{aligned}
R(\mathbb{P}\,||\mathbb{G}) &= \mathbb{E}^{\mathbb{P}} \log \frac{p(X)}{\tilde{p}(X)} \\
&= \mathbb{E}^{\mathbb{P}} \log \frac{p(X)}{\frac{q(X)e^{T(X)}}{\mathbb{E}^{\mathbb{Q}}[e^T(X)]}} \\
&= \mathbb{E}^{\mathbb{P}} \log \frac{p(X)}{q(X)} - \mathbb{E}^{\mathbb{P}} T(X) + \log \mathbb{E}_{\mathbb{Q}}\, e^{T(X)}
\end{aligned} \tag{50}$$

If part : We show that, for any T^*, if the associated Gibbs density $\tilde{p}^*$ is equal to p then T^* is the solution of $\sup_{T:\mathcal{X}\to\mathbb{R}} \left(\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]\right)$.

Let, $\mathbb{G}^*$ be the distribution with density $\tilde{p}^*$. Then we have,

$$\tilde{p}^* = p \implies R(\mathbb{P}\,||\mathbb{G}^*) = 0 \implies \mathbb{E}^{\mathbb{P}} \log \frac{p(X)}{q(X)} = \mathbb{E}^{\mathbb{P}} T^*(X) - \log \mathbb{E}_{\mathbb{Q}}\, e^{T^*(X)}$$

The last equality is due to Eq. (50). Since the LHS of the last equation is $R(\mathbb{P}\,||\,\mathbb{Q})$ following Donsker-Varadhan representation (Donsker & Varadhan, 1983) we have,

$$R(\mathbb{P}\,||\,\mathbb{Q}) = \mathbb{E}^{\mathbb{P}} T^*(X) - \log \mathbb{E}_{\mathbb{Q}}\, e^{T^*(X)} \leq \sup_{T:\mathcal{X}\to\mathbb{R}} \left(\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]\right) = R(\mathbb{P}\,||\,\mathbb{Q})$$

Therefore, T^* is the maximizer of $\left(\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]\right)$.

Only If part : We show that, if T^* is the solution of $\sup_{T:\mathcal{X}\to\mathbb{R}} \left(\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]\right)$ then $\tilde{p}^* = p$.

Let, $\mathbb{G}^*$ be the distribution with density $\tilde{p}^*$. From the Donsker-Varadhan representation, we have,

$$R(\mathbb{P}\,||\,\mathbb{Q}) = \sup_{T:\mathcal{X}\to\mathbb{R}} \left(\mathbb{E}^{\mathbb{P}}[T] - \log \mathbb{E}^{\mathbb{Q}}[e^T]\right) = \mathbb{E}^{\mathbb{P}}\, T^*(X) - \log \mathbb{E}_{\mathbb{Q}}\, e^{T^*(X)} \tag{51}$$

Therefore, we have,

$$R(\mathbb{P}\,||\mathbb{G}^*) = \mathbb{E}^{\mathbb{P}} \log \frac{p(X)}{q(X)} - \mathbb{E}^{\mathbb{P}}\, T^*(X) + \log \mathbb{E}_{\mathbb{Q}}\, e^{T^*(X)} = 0$$

The first equality is due to Eq. (50) and the second is from Eq. (51). Hence the proof follows.

□

C. Additional synthetic benchmarks

To benchmark how REMEDI copes with increasing dimensionality, we provide two additional benchmarks, the d-dimensional ball and hypercube, see Section D.3. Since the investigation concerns the high-dimensional performance on the REMEDI correction and not Gaussian mixture models (see Section C.1 for that analysis), we stick to a single component KNIFE base model, which also trains quickly For comparison, we juxtapose the results with the validation set cross-entropy estimate of the 256-component KNIFE model, which performs the best on almost all benchmarks in Section C.1.

From Figures 9 and 10, we note that the `REMEDI` entropy estimates are consistently considerably closer to the true entropy than what the KNIFE framework can achieve. Further, although some overfitting can be seen, it is nowhere near as bad as for KNIFE (see Appendix C.1). This means that there is likely more for `REMEDI` to claim with additional hyperparameter tuning.

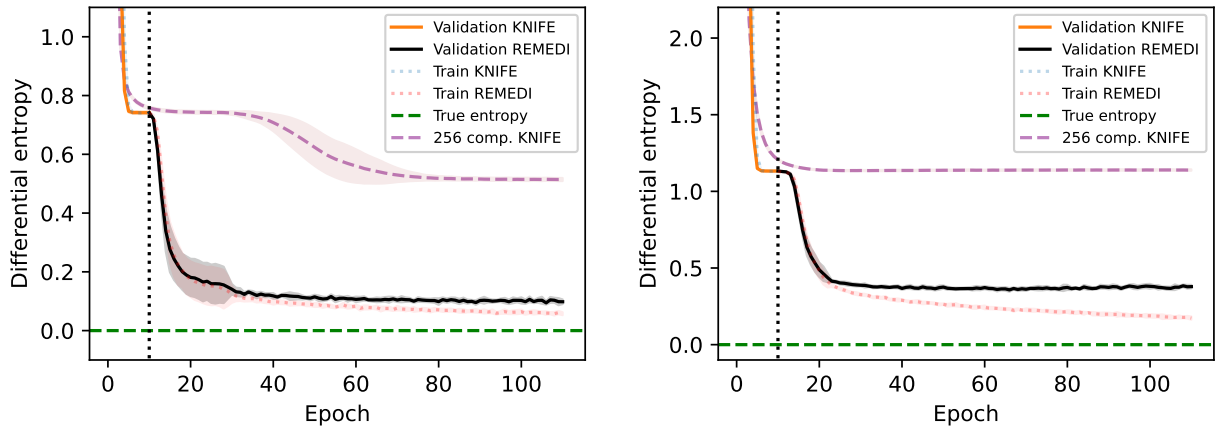


Figure 9: Training curves on ball dataset. The 8-dimensional dataset is shown to the left and the 20-dimensional to the right. The 256-component KNIFE model is also shown for reference.

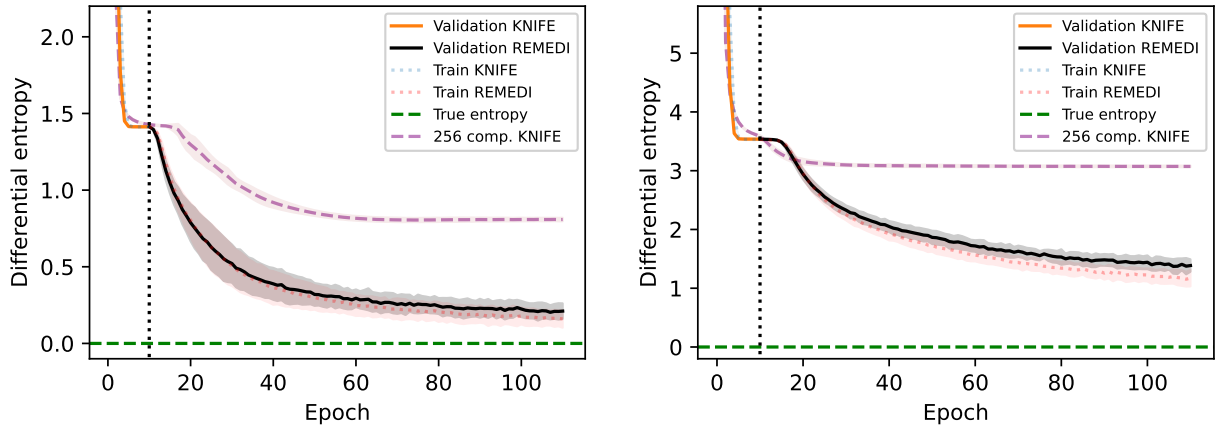


Figure 10: Training curves on cube dataset. The 8-dimensional dataset is shown to the left and the 20-dimensional to the right. The 256-component KNIFE model is also shown for reference.

C.1. Inefficiency of KNIFE

An often-cited folklore result in statistics is that Gaussian mixture models are dense in the space of probability measures equipped with the weak-* topology. Parzen-windowing is a subset of Gaussian mixture models with the same property. As a consequence, both model families can be considered for a wide range of tasks such as density and entropy estimation. (Ahmad & Lin, 1976) give conditions for when a certain Parzen-windowing-based estimator converges to the differential entropy. It is however well known that standard kernel density estimation suffers heavily from the curse of dimensionality, see e.g. the table on page 319 of (Wasserman, 2004), which shows that an unreasonable amount of data points is required to obtain low error on a multi-variate normal target $\mathbb{P}$. Since Gaussian mixture models such as KNIFE have more flexibility, they require less components to achieve the estimate. The question then remains whether they manage to be sufficiently data efficient in growing dimension. We show empirically that the KNIFE approach suffers from similar sample inefficiency problems, even in a moderate dimension.

Unlike (Wasserman, 2004) we cannot simply use a multi-variate normal as the target, since this is in the model class of KNIFE for any amount of components. Instead, we opt for the 8-dimensional targets comprised of uniforms over a hypercube/ball, as well as the 8-dimensional triangle dataset, with 50000 training and validation samples each, see Section D.3.

Fig. 1 and Fig. 11 show that KNIFE hits a barrier on all datasets when increasing the number of components. Increasing the number of components further leads to overfitting, implying that KNIFE is data-inefficient already in this dimension, especially on the cube and ball datasets where the best estimates are comparable with using just one component.

To give additional verification to this, we perform the experiment again in 20 dimensions, shown in Fig. 12. We note that additional components at best give an almost negligible improvement to all three datasets; in fact, we see no gain at all on the ball dataset.

It becomes clear that in both of these dimensions, that `REMEDI` can provide much more valuable entropy estimates, at least outside of gigantic sample sizes. Still, `REMEDI` like most algorithms will of course also suffer from problems like overfitting and the curse of dimension-

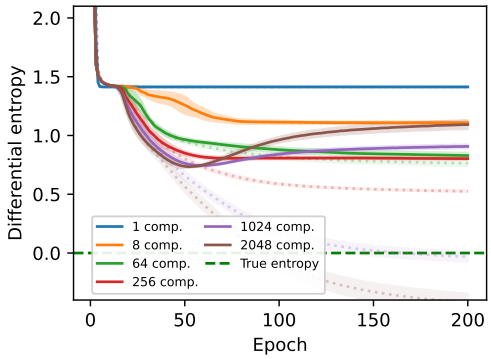
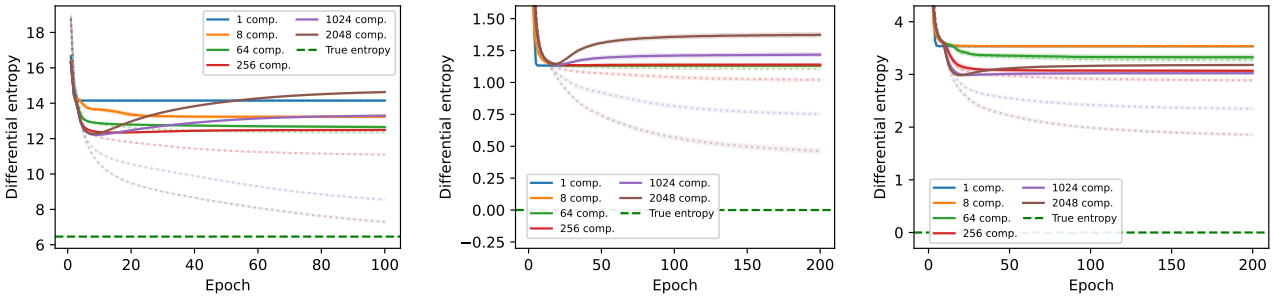


Figure 11: 8-dimensional cube

ality. However, the prior provided by the class of shallow neural networks trained with the Donsker-Varadhan target scales better to moderately high dimensions that Parzen-windowing and KNIFE struggle with.



(a) Triangle (b) Ball (c) Cube

Figure 12: KNIFE training curves on three 20-dimensional datasets.

C.2. Impact of the base distribution

One of the primary innovations of REMEDI compared to MINE or DDDE is the adaptive base distribution. The base distribution is optimized for cross-entropy, or equivalently in the large sample regime, to minimize the relative entropy

$$R(\mathbb{P} \,||\, \mathbb{Q}) = \mathbb{E}^{\mathbb{P}} \left[\log \left(\frac{d\,\mathbb{P}}{d\,\mathbb{Q}} \right) \right]. \tag{52}$$

This means that, on average, the samples from $\mathbb{Q}$ are closer to the support $\mathbb{P}$, and that regions where $\mathbb{P}$ is stronger than $\mathbb{Q}$, i.e. having large $\frac{d\,\mathbb{P}}{d\,\mathbb{Q}}$, are less pronounced. As (McAllester & Stratos, 2020) points out, these regions are hard to learn and account for in the Donsker-Varadhan estimate.

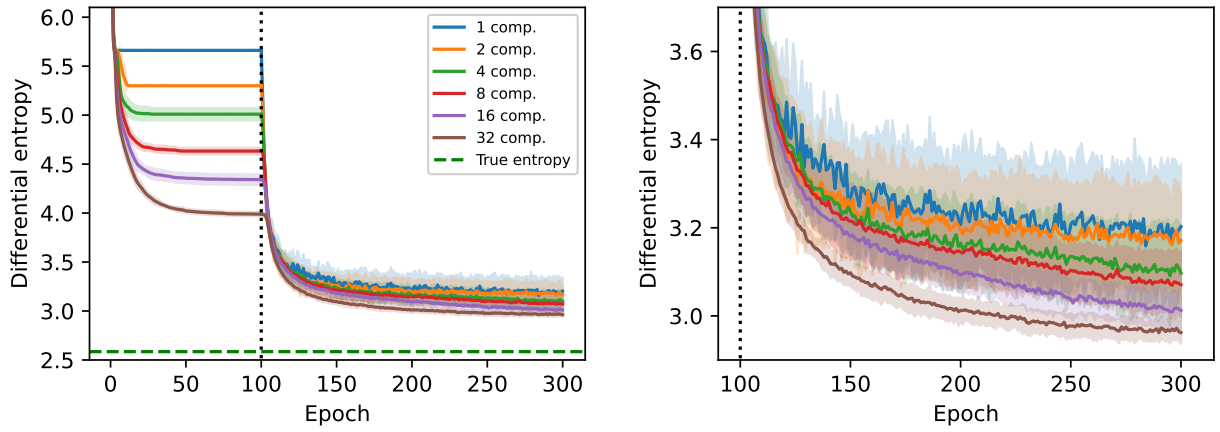


Figure 13: Training curves on 8-dimensional triangle dataset for different amounts of components in the base distribution, averaged over 10 runs. The right plot is zoomed into the REMEDI phase.

Components	KNIFE	REMEDI
1	5.6612 ± 0.0035	3.2025 ± 0.1414
2	5.2999 ± 0.0045	3.1713 ± 0.1039
4	5.0095 ± 0.0632	3.0968 ± 0.1068
8	4.6331 ± 0.0331	3.0708 ± 0.0750
16	4.3413 ± 0.0541	3.0124 ± 0.0449
32	3.9894 ± 0.0214	2.9621 ± 0.0026

Table 3: Entropy estimates and standard deviations on 8-dimensional triangle dataset, based on 10 runs.

To investigate what impact a good base distribution has, we rerun the experiments on the 8-dimensional triangle dataset. In order to get a better understanding of the long-term learning behavior, we double the number of epochs for both the base distribution and the REMEDI to 100 and 200, respectively. In Fig. 13 and Table 3 we see that learning with a higher amount of components is indeed easier. The estimates given by using 16 and 32 components clearly beat the others, and seem to be learning at a faster pace. Surprisingly, the difference is not as pronounced between 1 and 2 components, but we do note that the training stabilizes for more components, also there.

C.3. Comparison to normalizing flows

Other than mixture models such as KNIFE, more flexible methods of density estimation, such as normalizing flows, can be used for entropy computation. This allows for better estimates, at the cost of computational efficiency. Therefore, we provide a comparison between the performance of KNIFE/REMEDI and RealNVP (Dinh et al., 2016), a popular family of normalizing flows, at different computational budgets, measured in wall-clock time. For RealNVP, we use an out-of-the-box implementation provided by the 'normflows' (Stimper et al., 2023) package. In Fig. 14, the methods are compared on the multi-modal 8-dimension triangle dataset, for different depths (i.e. amount of flow steps) of RealNVP, while the REMEDI settings are untouched from earlier experiments. Taking the lower convex hull of all wall time-entropy estimate pairs, we see that KNIFE/REMEDI occupies practically the entire Pareto front. Note that there are other families of normalizing flows, continuous normalizing flows (CNF) (Grathwohl et al., 2018) such as diffusion models (Song et al., 2020b;a; Kingma et al., 2021) that may be able to better cope with this multi-modality, but since these require integrating the divergence of a vector field along the ODE solution (Chen et al., 2018), they have high complexity and we consider them out of scope.

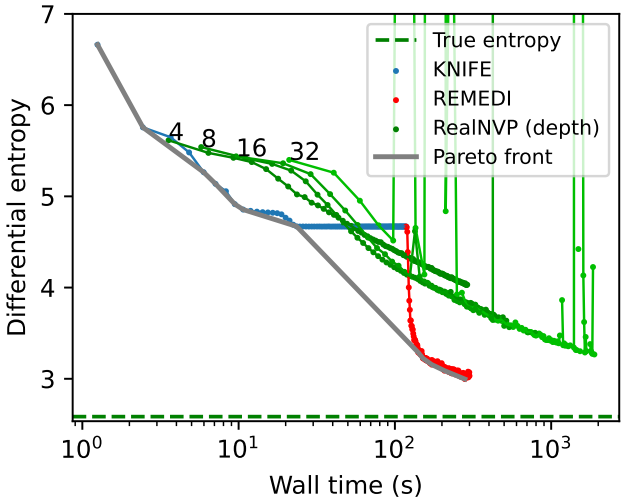


Figure 14: Performance of REMEDI and KNIFE given computational budget, compared to RealNVP.

D. Experimental details

D.1. Entropy estimation experiments

For the experiments on the triangular and two moons datasets, we opted for a neural network that outputs $f_\theta(x) = e^{T(x)}$ in the Donsker-Varadhan formula, letting $T(x)$ be obtained by taking the logarithm. Following (Park & Pardalos, 2021), this is implemented using the ELU activation function (Clevert et al., 2015) with $\alpha = 1$, by applying the transformation

$$\mathrm{ELU}(x) + 1 + \epsilon \tag{53}$$

to the final output of the model, where ϵ is a small scalar.

The architecture here exploits the structure of the KNIFE base distribution. For each of the M components i, with weight α_i, mean μ_i and covariance Σ_i (and precision $\Lambda_i = \Sigma_i^{-1}$), the input x is projected onto the uncorrelated components, with respect to Σ_i, by premultiplying by the lower triangular Cholesky factor L_i, of Λ_i. Hence, we obtain M component-wise decorrelated offset vectors

$$y_i = L_i(x - \mu_i) \in \mathbb{R}^d \,.$$

These are then transformed via learnable matrices $A_i \in \mathbb{R}^{d \times d}$, since the features produced by L_i are not consistent between dimensions, even when performing singular value decomposition. These are then propagated through a fully connected intermediate network with two (three for the 20-dimensional hypercube) shared linear layers followed by ReLU activation functions, reaching the penultimate layer with dimension d_p (in most experiments 500). They are then scalar multiplied with vectors $b_i \in \mathbb{R}^{d_p}$ to produce scalars, which are then combined via weighting by the component relevances

$$p(i|x) = \frac{p(i)p(x|i)}{p(x)} \propto w_i \exp\left(-\frac{1}{2}||L_i(x-\mu_i)||^2\right). \tag{54}$$

This weighted sum is finally input to the transformation in Eq. (53). In Table 4, the specific setups for each of the entropy estimation tasks are shown.

Hyperparameter	Two moons	Triangle		Ball		Hypercube	
Dimension	2	1	8	8	20	8	20
Train set size	50,000	50,000	50,000	50,000	50,000	50,000	50,000
Validation set size	50,000	50,000	50,000	50,000	50,000	50,000	50,000
# KNIFE components	8	16	16	1	1	1	1
Intermediate network layer widths	(500, 500)	(500, 500)	(500, 500)	(200, 200)	(200, 200)	(1000, 1000, 500)	(1000, 1000, 500)
# epochs KNIFE	50	50	50	10	10	10	10
# epochs REMEDI	100	100	100	100	100	100	100
Training batch size	1000	1000	1000	1000	1000	1000	1000
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Learning rate	1e-3	1e-3	1e-3	1e-3	1e-3	1e-4	1e-4
Weight decay	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4

Table 4: Hyperparameter settings used for the entropy estimation experiments.

D.2. Information Bottleneck experiments

For all the Information Bottleneck (IB) experiments, we have used the encoder-decoder architecture from (Samaddar et al., 2023). On MNIST, we used an MLP encoder with three fully connected layers. The first two layers each contain 800 nodes with ReLU activations and the last layer has $2K$ nodes predicting the $\mu(X)$ and diagonal of $\Sigma(X)$ of the encoder distribution. On CIFAR-10, we used a VGG16 encoder. For both datasets, we used a single-layer neural network as the decoder. We chose the latent space dimension $K = 32$ for both datasets. Note that, we apply the square transformation of $\mathrm{MI}(X; Z)$ term in the IB objective for all methods. This transformation makes the solution to IB objective function identifiable with respect to β (Rodríguez Gálvez et al., 2020). For evaluation, we chose the model at the final epoch on MNIST and the model with the best validation loss on CIFAR-10 and ImageNet.

For the CIFAR-10 data following (Samaddar et al., 2023), we perform a data augmentation step before training where we augmented the training data using random transformations. We use the padding of each training data point by 4 pixels on all sides and crop at a random location to return an original-sized image. We perform a flip of each training set image horizontally with a probability of 0.5. Furthermore, we transform the training and validation set image with mean = $(0.4914, 0.4822, 0.4465)$ and standard deviation = $(0.2023, 0.1994, 0.2010)$.

For ImageNet, we resize the input images to 299 $\times$ 299 pixels by cropping them at their center. Subsequently, we normalize the images to achieve a mean of (0.5, 0.5, 0.5) and a standard deviation of (0.5, 0.5, 0.5). Our approach aligns with the implementation of (Alemi et al., 2016), where we apply a transformation to the ImageNet data using a pre-trained Inception Resnet V2 (Szegedy et al., 2016) network, excluding the output layer. This transformation results in the original ImageNet images being reduced to a 1534-dimensional representation, which serves as the basis for all our obtained results. In accordance with (Alemi et al., 2016), we employ an encoder featuring two fully connected layers, each comprising 1024 hidden units, along with a single-layer decoder architecture. We chose the latent space dimension $K = 100$ for ImageNet.

The implementations of MINE and `REMEDI` require us to specify a neural network to approximate the function T. For MINE, following the implementation of (Belghazi et al., 2018) we chose a two-layer MLP with 512 nodes each and additive Gaussian noise and ELU activations. For `REMEDI`, we chose a two-layer MLP network with 100 nodes per layer and ReLU activations.

Implementing the mutual information estimators of KNIFE, MINE, and `REMEDI` requires us to perform sub-optimization of their parameters inside the main optimization of the encoder-decoder parameters. We follow the IB implementations of (Pichler et al., 2022) to freeze the parameters of the encoder-decoder before performing the sub-optimization of the mutual information estimators. For KNIFE and MINE, we run the sub-optimization for 5 epochs. We train the mutual information estimators and the encoder-decoder parameters using the same mini-batch.

For `REMEDI` with KNIFE base distribution, following Algorithm 1, we first train the KNIFE parameters for 5 epochs and then fix the KNIFE parameter to train the parameters of the `REMEDI` network for 5 epochs. Note that, although ImageNet has 1000 classes, we keep the choice of 10-component KNIFE consistent due to the heavy computational burden of fitting a KNIFE with many components in high dimensions. We use a constant learning rate of 0.001 for training the mutual information estimators.

For `REMEDI` with standard Gaussian base distribution, since we are facing a more challenging learning problem we train the parameters of the `REMEDI` network for 30 epochs with a reduced learning rate of 0.0001. In addition, during training, we consider an initial burn-in period of 5 epochs where we don't train the `REMEDI` network and introduce it after epoch 5. In our experiments, this improves the stability of the algorithm. Additional hyperparameter details regarding the encoder-decoder training are described in the below Table 5.

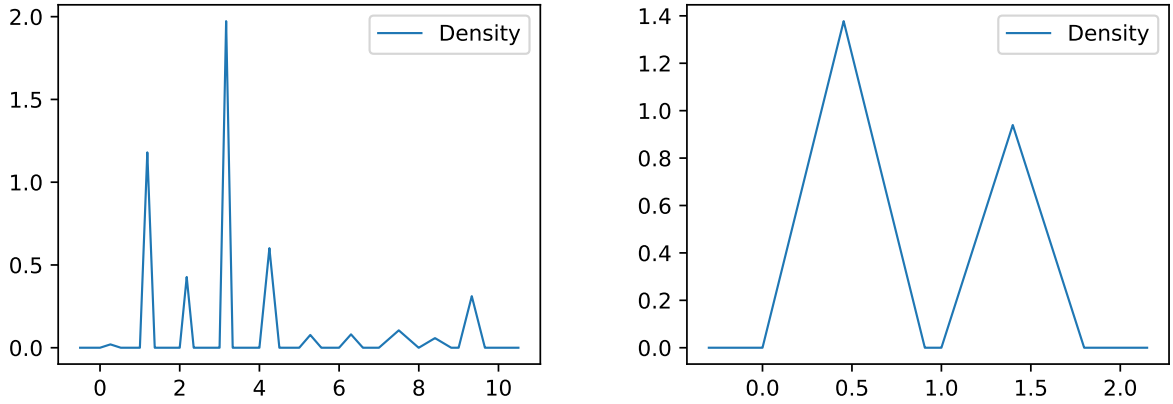
Hyperparameters	**MNIST**	**CIFAR-10**	**ImageNet**
Train set size	60,000	50,000	128,1167
Validation set size	10,000	10,000	50,000
# epochs	100	400	200
Training batch size	200	200	2000
Optimizer	Adam	SGD	Adam
Learning rate	1e-4	0.1	1e-4
Learning rate drop	0.6	0.1	0.97
Learning rate drop steps	10 epochs	100 epochs	2 epochs
Weight decay	Not used	5e-4	Not used

Table 5: Hyperparameter settings used for the IB experiments.

D.3. Datasets

D.3.1. TRIANGULAR

The triangular dataset is structurally the same as the one appearing in (Pichler et al., 2022). It is defined for any dimension $d > 1$ as the d-fold product distribution of bimodal the distribution pictured in Fig. 15b with itself, making it a 2^d-modal distribution. In one dimension, we use the 10-modal distribution in Fig. 15a, to match (Pichler et al., 2022).



(a) One-dimensional data distribution.

(b) Marginal data distribution, for $d > 1$.

Figure 15: Triangle dataset.

D.3.2. Two moons

The two moons dataset consists of samples from `sklearn.datasets.make_moons()`, from Scikit-learn (Pedregosa et al., 2011), with a noise level of 0.05. 5000 samples are plotted in Fig. 16. The entropy of this dataset does not offer a closed-form expression. To have an oracle baseline, we thus take a million samples from the dataset and run a kernel density estimator, with bandwidth 0.01, tested against one hundred thousand independent samples. This yields an entropy estimate of 0.2893, with a standard error of 0.0022. As an additional check, we run a Kozachenko-Leonenko k-nearest neighbor estimator (Kozachenko & Leonenko, 1987; Gao et al., 2018) on one hundred million samples, setting $k = 10$ and using the Euclidean distance, which yields a value of 0.2892. Given the consistency properties of these estimators, we can confidently say that the true entropy is close to 0.29.

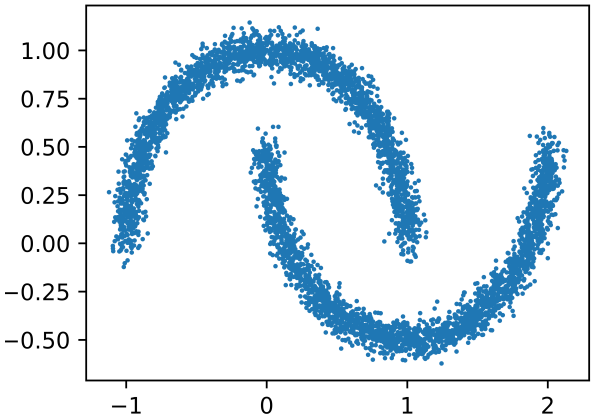
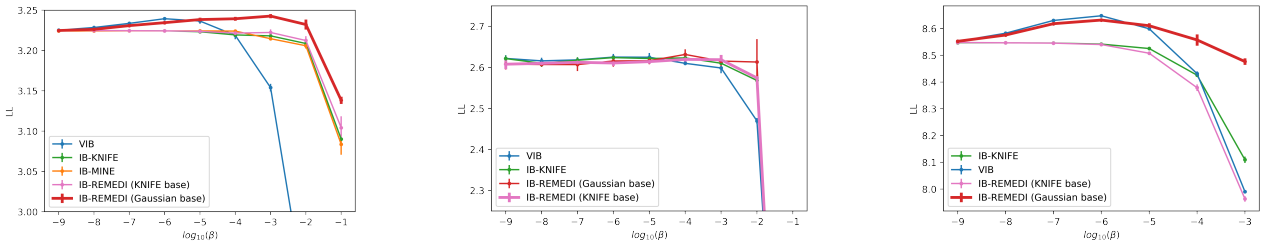


Figure 16: Samples from the two moons dataset.

D.3.3. Uniform hypercube/ball

The datasets used to compare dimensional scaling between KNIFE and REMEDI are from the uniform distribution over a centered unit volume d-dimensional ball B^d and cube H^d, respectively. This is easily scaled to any dimension d. By the unit volume, the true differential entropy of each dataset is 0, for all dimensions.



(a) MNIST (b) CIFAR10 (c) ImageNet

Figure 17: Plot showing log-likelihoods of the Information Bottleneck methods vs β on benchmark image classification datasets (error bars represent standard deviations). For most β values, consistently `REMDI` performs better than other methods on MNIST and ImageNet. On CIFAR10, the classification errors are similar for all the methods. However, `REMEDI` exhibits the highest log-likelihood across the β values.

E. Code

The code needed to replicate the experiments of this paper is found at `https://github.com/viktor765/REMEDI`. The project also uses code from the KNIFE repository, with the authors' permission, found at `https://github.com/g-pichler/knife/`.

F. Computational cost

The suite of pure entropy estimation tasks, including the two moons, triangle, ball, and hypercube datasets were run on NVIDIA A100 and finished in approximately 29 hours. All IB experiments were run also using NVIDIA A100 GPUs. One replication of IB with `REMEDI` run for a single β value took close to 2.5 hours for MNIST, 15 hours for CIFAR-10, and 35 hours for ImageNet.

G. Additional results on Information Bottleneck

In this section, we provide additional results from applying different mutual information estimation approaches to the Information Bottleneck.

G.1. Results based on log-likelihood

In this section, we evaluate the IB methods based on the log-likelihood metric on the three datasets. Fig 17, shows the plot of the log-likelihood vs the Lagrange multiplier β for the IB methods. We observe that similar to classification accuracy results in the main paper `REMEDI` exhibits the highest log-likelihood on MNIST and ImageNet for most β values especially those around the region where the log-likelihood starts to decrease. On CIFAR-10, although all the methods perform similarly in terms of log-likelihood `REMEDI` with Gaussian base distribution produces the highest log-likelihood.

G.2. Latent space evolution on MNIST

In Table 6, we plot different components of the 2-d latent space learned by IB-REMEDI with KNIFE base distribution throughout training on the MNIST dataset. The first column shows the latent space samples and the second and third columns show the KNIFE (10 components) and REMEDI contours as the training progresses. As the epoch increases, we observe that the latent space evolves into 10 clusters. We highlight that KNIFE struggles to learn the clusters especially when overlapping (e.g. epoch 0). To this end, REMEDI corrects the trained KNIFE and can locate the mass correctly around the clusters.

G.3. Analysis of the latent space on CIFAR-10

In this section, we present the 2-d latent space analysis on CIFAR-10. Similar to MNIST, we observe the samples from the latent space show clusters corresponding to the classes in Fig. 18a. In Fig. 18b, we observe that the KNIFE fits the density

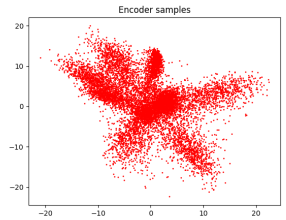
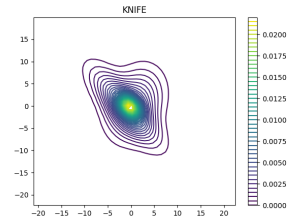
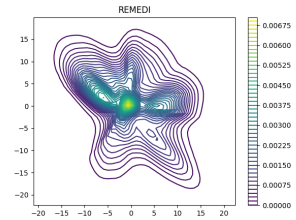
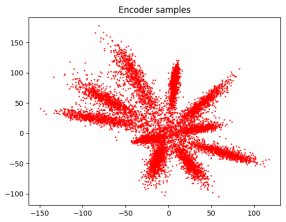
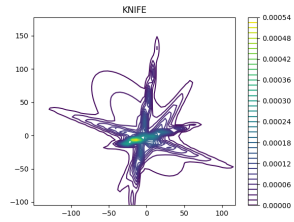
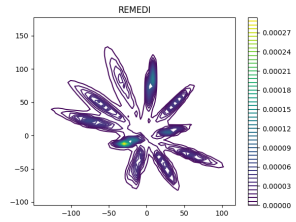
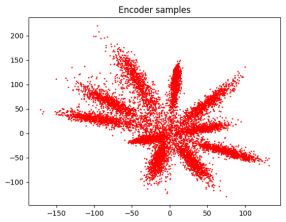
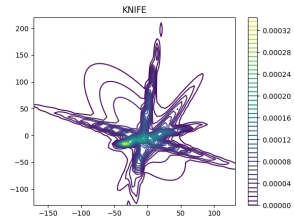
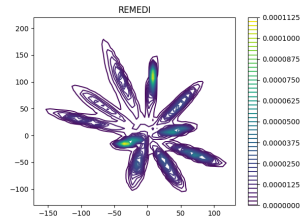
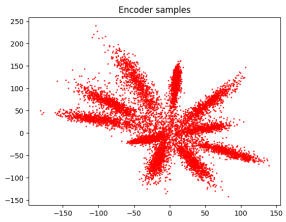
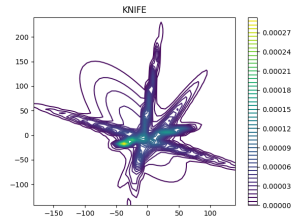
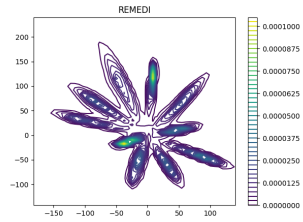
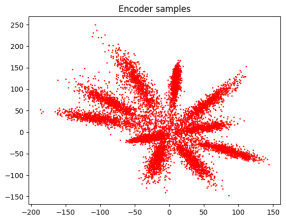
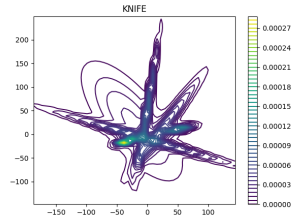
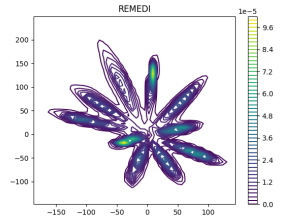
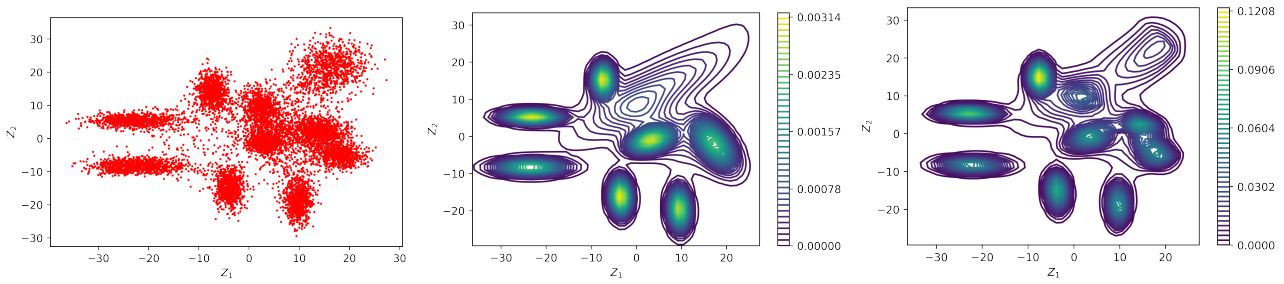
Epoch	Latent space	KNIFE	REMEDI
0			
20			
40			
60			
90			

Table 6: Sequence of latent space plots showing the evolution of the encoder samples and KNIFE and `REMEDI` density contours during training IB-REMEDI (KNIFE base) ($\beta = 1e-09$) on MNIST.



(a) Encoder samples (b) KNIFE contours (c) REMEDI contours

Figure 18: REMEDI marginal distribution of 2-d latent space on CIFAR-10.

well which is further improved by REMEDI corrections in Fig. 18c. However, the gain from applying REMEDI from the KNIFE step is less in CIFAR-10 than it is on MNIST.