Towards Robust Scale-Invariant Mutual Information Estimators

Anonymous authors
Paper under double-blind review

Abstract

Mutual information (MI) is hard to estimate for high dimensional data, and various estimators have been proposed over the years to tackle this problem. Here, we note that there exists another challenging problem, namely that many estimators of MI, which we denote as $I(X;Y)$, are sensitive to scale, i.e., $I(X;\alpha T) \neq I(X;T)$ where $\alpha \in \mathbb{R}$. Although some normalization methods have been hinted at in previous works, there is no in-depth study of the problem. In this work, we study new normalization strategies for MI estimators to be scale-invariant, particularly for the Kraskov–Stögbauer–Grassberger (KSG) and the neural network-based MI (MINE) estimators. We provide theoretical and empirical results and show that the original un-normalized estimators are not scale-invariant and highlight the consequences of an estimator's scale-dependence. We propose new global normalization strategies that are tuned to the corresponding estimator and scale invariant. We compare our global normalization strategies to existing local normalization strategies and provide intuitive and empirical arguments to support the use of global normalization. Extensive experiments across multiple distributions and settings are conducted, and we find that our proposed variants KSG-Global-L_∞ and MINE-Global-Corrected are most accurate within their respective approaches. Finally, we perform an information plane analysis of neural networks and observe clearer trends of fitting and compression using the normalized estimators compared to the original un-normalized estimators. Our work highlights the importance of scale awareness and global normalization in the MI estimation problem.

1 Introduction

Mutual information (MI), is a fundamental measure of dependency between two variables, which has become pivotal in various machine learning domains, including generalization (Xu & Raginsky, 2017; Bu et al., 2019; Russo & Zou, 2020), representation learning (Bachman et al., 2019; Tschannen et al., 2020) and fairness (Wang et al., 2023; Roh et al., 2020). Estimating MI for high-dimensional continuous variables (Xu et al., 2020) is particularly challenging, due to the hardness of accurately estimating the probability distribution in high dimensions (Goldfeld & Greenewald, 2021). For example, traditional estimators like Kraskov–Stögbauer–Grassberger (KSG) (Kraskov et al., 2004), rely on distance metrics, and for high dimensional data, the distances would have less variation due to the curse of dimensionality.

In this paper, we highlight a critical but underexplored factor that leads to inaccuracies in MI estimation: the scale of the variables (i.e., $|X|$). Specifically, when considering the mutual information $I(X;\alpha Y)$, where $\alpha \in \mathbb{R}^+$ is a scaling factor, we demonstrate that when $\alpha \ll 1$ or $\alpha \gg 1$, the MI estimates can deviate significantly from the true value. This is problematic since by definition, $I(X;\alpha Y) = I(X;Y)$ for any two continuous random variables X and Y. Moreover, a stronger result states that $I(X;f(Y)) = I(X;Y)$ for any continuous and invertible transformation f (Cover & Thomas, 2006). In this paper, we mainly focus on the specific impact of scale.

Most mutual information estimators, including the widely adopted KSG estimator (Kraskov et al., 2004) and its subsequent variants (Gao et al., 2017), lack scale invariance—a limitation that we rigorously demonstrate in this study. We provide a theoretical analysis explaining why this deficiency arises. We also show the

binning estimator (Paninski, 2003) for MI can be scale invariant when the number of bins used is fixed. However, the binning estimator itself is not well-suited for estimating high-dimensional continuous variables. Recently the mutual information neural estimator (MINE) (Belghazi et al., 2018) was proposed, which is a neural network-based estimator of MI that makes use of its Donsker-Varadhan (DV) representation. We demonstrate theoretically that ideally, MINE should be scale-invariant, but MINE fails in practice due to limitations introduced by stochastic gradient descent optimization.

Despite numerous surveys that have explored various methods of MI estimation (Walters-Williams & Li, 2009; McAllester & Stratos, 2020; Paninski, 2003), the critical importance of normalization (preprocessing) has been largely overlooked. A natural solution to ensure scale invariance is to pre-process the data using standard normalization, where each dimension is adjusted to have a variance of 1, and we refer as *local normalization*. This pre-processing step was hinted in (Kraskov et al., 2004) for the KSG estimator. Local normalization also has been commonly applied as a preprocessing step in many deep learning studies involving mutual information perspective (Hjelm et al., 2019; Xie et al., 2024). However, local normalization treats each dimension independently and normalizes them to have a variance of 1, which, as we demonstrate in Section 5.1, does not work well in the high-dimension setting especially in neural networks, across two separate experiments. This is because most high-dimensional feature representations in neural networks always contain some noisy dimensions, which are of low energy and contain irrelevant features. Thus, amplifying these low energy dimensions can lead to suboptimal MI estimates. We also note that the recent work by (Czyż et al., 2023), in addition to trying out local normalization approaches, also studied other preprocessing methods including the transformation of the margin distribution to uniform distribution (via converting to rank). We note that this conversion step also brings all individual dimensions to equal importance like local normalization, and thus would have the same pitfalls in this scenario.

To address this issue, in our work, we propose a set of *global normalization* approaches. Unlike local normalization, global normalization preserves the relative energies between the different dimensions, and thus avoids scaling up low-energy noisy dimensions. Our proposed estimator modifications do not only include new normalization approaches, however, and often also have an additional maximization step, which helps bias our estimators better. It is well known that KSG and other MI estimators have a tendency to have negative bias Czyż et al. (2023), especially in high dimensions. Our normalization approaches for KSG incorporate this observation via an additional maximization step, which also follows intuitively from one of our theoretical observations in Proposition 3.

We now summarize our contributions:

- We propose novel scale-invariant extensions of KSG and MINE-based estimators that effectively address the one-sided scale-invariance issue and substantially improve estimator accuracy. To the best of our knowledge, our work is the first comprehensive analysis of the effect of scale and various normalization methodologies, some of which are introduced for the first time in this work.
- We demonstrate that the KSG-Global-L_∞ and MINE-Global-Corrected variants consistently produce the most accurate estimations within their respective approaches, across a broad range of experiments involving synthetic data, which are targeted towards the high-dimensional and low-data regime. These experiments include multiple types of transformations, noise injections, and changes in dimensionality.
- We explore the dynamics of MI between inputs X and hidden layers T during neural network training. Our results highlight that unnormalized estimators significantly confound the scale of T in their estimates, while our normalized approaches can often capture distinct phases of training, such as fitting and compression.

The rest of the paper is organized as follows. In Section 2, we first provide a formal definition of mutual information and briefly review the common MI estimators employed in our study, highlighting the motivation behind enhancing these estimators. In Section 3, we then present an evaluation of the one-sided scale-invariance issue across three selected MI estimators. In Section 4, we introduce our proposed normalization strategies, accompanied by key implementation techniques for improving the KSG and MINE estimators. In Section 6 and Section 7, we conduct extensive experiments on both synthetic datasets and during neural network training to demonstrate the efficacy of our method. Comparisons are made against both the original estimators and standard normalization approaches (local normalization). Finally, we summarize our findings and discuss their implications in Section 8.

2 Background

2.1 Mutual Information

Mutual information of two variables is a measure that quantifies the mutual dependence between two random variables. Specifically, it measures the amount of information obtained about one random variable through the observation of another. The concept of mutual information is closely related to the Shannon entropy, which measures the average uncertainty or information of a random variable's possible outcomes. Given a continuous random variable X with a probability density function f from a set $\mathcal{X}$, the continuous entropy $h(X)$ is defined as:

$$h(X) := -\int_{\mathcal{X}} f(x) \log f(x)\, dx \tag{1}$$

Then, the mutual information between continuous random variables (RVs) X and Y is given by:

$$I(X;Y) = h(x) + h(Y) - h(X,Y) \tag{2}$$

where $h(X,Y)$ is the joint Shannon entropy of X and Y. This can be interpreted as the reduction in the uncertainty of X due to the knowledge of Y, or equivalently, as the amount of information that X and Y share.

In the case of jointly continuous random variables, the mutual information can be expressed in terms of Kullback–Leibler (KL-) divergence

$$I(X;Y) = D_{\mathrm{KL}}\left(P_{(X,Y)} \| P_X \otimes P_Y\right), \tag{3}$$

where $P_X \otimes P_Y$ is the dot product of two marginal distributions P_X and P_Y, $P_{(X,Y)}$ is their joint distribution. D_{KL} is defined as $D_{\mathrm{KL}}(P\|Q) := \mathbb{E}_P\left[\log \frac{dP}{dQ}\right]$.

Equation 2 and equation 3 are commonly used to describe mutual information. However, in practice, estimating the true distribution of continuous random variables is challenging, especially for high-dimensional data. In the following section, we will discuss various estimators used in other works to estimate the distribution of random variables and subsequently compute mutual information.

2.2 Mutual Information Estimators

In this section, we present several widely-used nonparametric MI estimators that are studied in our work and have been extensively applied in other research.

Binning Estimator: It is also called histogram based estimator in many research. The simplest approach to estimate MI is discretizing the continuous random variable into bins, counting the number of samples that fall into each bin, and computing the probability density (Paninski, 2003). The binning estimator for n samples can be expressed as $\widehat{I}^n(X;Y)_{bin} = H_{bin}(X) + H_{bin}(Y) - H_{bin}(X,Y)$. where $H_{bin}(X)$ represents the binned entropy given a RV X, such that $H_{bin}(X) = -\sum_i P(X_i)\ \log P(X_i)$. Let $n(X_i)$ be the number of samples that fall in i-th bin of X, and N is the total number of data points. Then we have $P(X_i) \approx n(X_i)/N$ for binning method. Similarly, we represent binned joint entropy as $H_{bin}(X,Y) = -\sum_{i,j} P(X_i,Y_j)\ \log P(X_i,Y_j)$, and $P(X_i,Y_j) \approx n(X_i,Y_j)/N$.

Kraskov–Stögbauer–Grassberger (KSG) Estimator: Another popular non-parametric approach to estimate MI in high dimensions is the KSG estimator in (Kraskov et al., 2004). Unlike the binning estimator, the KSG estimator uses the k-nearest neighbor (K-NN) statistic to estimate the probability function of continuous random variables, which also uses the joint entropy decomposition method to estimate MI. The KSG estimator is effectively estimating:

$$\widehat{H}_{KL}(X) + \widehat{H}_{KL}(Y) - \widehat{H}_{KL}(X,Y), \tag{4}$$

where H_{KL}represents the KL Entropy estimator which is proposed in Kozachenko & Leonenk (1987)'s work. Informally, in l_p distance, each k-NN distance $\rho_{k,i,p}$ along with the choice of k, provides a localized perspective

on the underlying probability distribution around the i-th sample. The probability density function of X_i under k-NN statistic can be approximately expressed as: $\widehat{f}_X(X_i)\, c_{d,p}\, (\rho_{k,i,p})^d \simeq \frac{k}{N}$, where N is the number of total samples. With this function and equation 1, we are able to get:

$$\widehat{H}_{\mathrm{KL}}(X) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{N c_{d,p}\, (\rho_{k,i,p})^d}{k}\right) + \log(k) - \psi(k), \tag{5}$$

where $c_{d,p}$ is the volume of d-dimensional balls in l_p distance, and $\psi(x)$ is the digamma function (i.e., $\psi(x) = \Gamma(x)^{-1} d\Gamma(x) dx$). Note that we have estimated the density of i-th sample, thus the integral in equation 1 can be rewritten in terms of summation over N samples. As the KSG estimator measures distances using the l_∞norm, as it can be written as:

$$\widehat{I}^n_{KSG}(X;T) = \psi(k) + \psi(n) - \frac{1}{k} - \frac{1}{n}\sum_{i=1}^{n} \left(\psi(n_{x,i,\infty}) + \psi(n_{t,i,\infty})\right) \tag{6}$$

In (Gao et al., 2017), authors proposed a bias-improved KSG (**BI-KSG**) that performs better than KSG when N is small and X and Y are not independent. It is also important to note that many other variants of KSG and other estimators (Pál et al., 2010; Gao et al., 2015) use k-NN approach.

Mutual Information Neural Estimator (MINE): In our work, we also look into neural network based MI estimators, specifically Mutual Information Neural Estimation (MINE) (Belghazi et al., 2018). This approach adopts the DV representation of KL-divergence (Donsker & Varadhan, 1983).Given RVs $X \sim P_X$, $Y \sim P_Y$, and $(X_i, Y_i) \sim P_{X,Y}$, we express equation 3 in terms of DV representation to get:

$$I(X;Y) = \sup_{F:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \mathbb{E}_{X,Y\sim P(X,Y)}[F(X,Y)] - \log\left(\mathbb{E}_{X,Y\sim P(X)\times P(Y)}\left[e^{F(X,Y)}\right]\right), \tag{7}$$

where F can be any class of functions that satisfying the integrability constraints of the theorem.

Assuming independent and identically distributed (i.i.d.) samples are drawn from $P(X,Y)$, and X_i and $\tilde{Y}_i$, where $\tilde{Y}_i$ is taken from the randomly shuffled set of all samples $(Y_i)_{i=1}^n$. When n is large enough, by applying the law of large numbers, we have:

$$\widehat{I}_{MINE}(X;Y) = \max_F \frac{1}{n}\sum_{i=1}^{n} T_\theta(X_i, Y_i) - \log\left(\frac{1}{n}\sum_{i=1}^{n} e^{T_\theta(X_i,\tilde{Y}_i)}\right), \tag{8}$$

where we choose F to be the family of functions $T_\theta : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ to be parameterized by a deep neural network with parameters $\theta \in \Theta$. By training a neural network to optimize the above equation (i.e., finding the optimal T_θ), the final output will yield MINE estimate of MI between X and Y.

2.3 Motivation for Improving MI Estimation

Estimating mutual information (MI) is fundamental to various domains, ranging from learning theory to practical applications such as medical analysis and wireless communication (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018). To motivate our proposed normalization strategy, this section outlines several desirable properties that effective MI estimators should possess. Let $S = \{(X_1,T_1),(X_2,T_2),...,(X_n,T_n)\}$ be the sampled data. With this, let $\widehat{I}^n_{est}(X;T)$ represent an estimate of the MI between X and T using the estimator est, given N sampled points from the joint distribution $P(X,T)$. Ideally, we seek the estimator to have the following properties:

1. **Global Scale Invariance:** For any arbitrary $\alpha \in \mathbb{R}$ and $n \in \mathbb{Z}^+$, $\widehat{I}^n_{est}(\alpha X;\alpha T) = \widehat{I}^n_{est}(X;T)$
2. **One-Sided Scale Invariance** For any arbitrary $\alpha \in \mathbb{R}$ and $n \in \mathbb{Z}^+$, $\widehat{I}^n_{est}(X;\alpha T) = \widehat{I}^n_{est}(X;T)$

We emphasize the importance of these properties because true mutual information inherently satisfies them. By definition, $I(\alpha X;\alpha T) = I(X;T)$ and $I(\alpha X;T) = I(X;T)$ for a scalar α. In the case of neural networks, where X represents the input and T the features, estimation of $I(X;T)$ becomes important, as it was hypothesized that it can predict the generalization behaviour of deep learning networks (Shwartz-Ziv & Tishby,

2017). Furthermore, (Shwartz-Ziv & Tishby, 2017) also predicts a two-phase behaviour of $I(X;T)$ during training: (a) fitting, where $I(X;T)$ and $I(T;Y)$ increases, and (b) compression where $I(X;T)$ decreases. However, this is often not observed (Saxe et al., 2018). We hypothesize that it could be because of the scale-sensitivity of the estimators, as the scale of T changes significantly during training.

We note that the current estimators may not obey one-sided scale invariance. First, we study three estimators theoretically: KSG, MINE, and binning.

3 Testing One-sided Scale-Invariance of MI Estimators

In this section, we theoretically test whether the common MI estimators are global-scale invariant and one-sided scale invariant. In Section 6, we also present an experimental test of one-sided scale-invariance on MI estimators. Note that for all results that follow, we assume every random variable is bounded. That is, if X is bounded, we have that $|X| \leq B$ for some finite $B < \infty$. Also, for the following results, let $X \in \mathbb{R}^d$ and $T \in \mathbb{R}^m$.

Binning: Let us denote the binning estimator described in (Paninski, 2003) by $\widehat{I}^n_{bin}$. Then we have the following result.

Proposition 1. It holds that $\widehat{I}^n_{bin}(\alpha X;\alpha T) = \widehat{I}^n_{bin}(X;T)$ and $\widehat{I}^n_{bin}(X;\alpha T) = \widehat{I}^n_{bin}(X;T)\ \forall \alpha \in \mathbb{R}^+$.

Proof Sketch 1. We note that the number of bins chosen for each dimension is fixed, and the locations of the bins are determined by the minimum and maximum values of the data in each dimension, i.e., they determine the edges of the bins. Let $X_{min} \in \mathbb{R}^d$ then denote the vector of minimum values across all dimensions, and vice-versa for X_{max}. When X scales to αX, as $\alpha > 0$, we have that the vector of minimum values for αX is simply αX_{min} and similarly for X_{max}, and the binning locations also get scaled by α. Thus, there is a bijection between the binning locations of X and αX. Since both the binning structure and the data points within each bin are scaled uniformly, the probability of data falling into any given bin remains unchanged. Therefore, the distribution of data across the bins is invariant under scaling, leading to the same binning estimator $\widehat{I}^n_{bin}(X;T) = \widehat{I}^n_{bin}(\alpha X;T) = \widehat{I}^n_{bin}(\alpha X;\alpha T)$ for any scaling factor α.

Remark 1. We note that even though the binning estimator is scale-invariant, it is not a good estimator for MI, more so in the high dimension setting (Kraskov et al., 2004). This is because in high dimensions the data occupies the space very sparsely, and most bins will yield zero datapoints and thus a zero probability. Due to this, it is common practice to use fewer bins overall, which instead leads to less accurate estimates of MI as more information is lost.

KSG: Let us denote the KSG estimator proposed in (Kraskov et al., 2004) by $\widehat{I}^n_{KSG}$. Then, we have the following results.

Proposition 2. It holds that $\widehat{I}^n_{KSG}(\alpha X;\alpha T) = \widehat{I}^n_{KSG}(X;T)$, $\forall \alpha \in \mathbb{R}^+$.

Proof Sketch 2. We note the expression for the KSG estimator (equation 3 from (Kraskov et al., 2004) as follows:

$$\widehat{I}^n_{KSG}(\alpha X;\alpha T) = \psi(k) + \psi(n) - \frac{1}{k} - \frac{1}{n}\sum_{i=1}^{n}\left(\psi(n_{\alpha x,i,\infty}) + \psi(n_{\alpha t,i,\infty})\right) \tag{9}$$

Here, ψ denotes the digamma function(Abramowitz, 1974), and $n_{\alpha x,i,\infty} = \sum_{j\neq i}\mathbb{I}\{||\alpha X_i - \alpha X_j|| \leq \rho'_{k,i,p}\}$, where $\rho'_{k,i,p}$ is the k-NN distance of the joint sample i, $\{\alpha X, \alpha T\}$ (this distance is computed in $d+m$ dimensions). Furthermore, $||\alpha X_i - \alpha X_j||$ represents the X-dimensions only distance (i.e. in d dimensional space). Let $\rho'_{k,i,p}$ be the k-NN distance of the joint sample i for the unscaled variables $\{X, T\}$. It is trivial to see that $\rho'_{k,i,p} = \alpha\rho_{k,i,p}$. Thus, $n_{\alpha x,i,\infty} = \sum_{j\neq i}\mathbb{I}\{||\alpha X_i - \alpha X_j|| \leq \alpha\rho_{k,i,p}\} = \sum_{j\neq i}\mathbb{I}\{||X_i - X_j|| \leq \rho_{k,i,p}\} = n_{x,i,\infty}$, and similarly $n_{\alpha t,i,\infty} = n_{t,i,\infty}$. This shows that $\widehat{I}^n_{KSG}(\alpha X;\alpha T) = \widehat{I}^n_{KSG}(X;T)$.

This proof also leads to the following result which states that one-sided scale invariance is not a property of the KSG estimator.

Proposition 3. It holds that $\lim_{\alpha,n\to\infty}\widehat{I}^n_{KSG}(X;\alpha T) = -\frac{1}{k}$ and $\lim_{\alpha\to 0^+, n\to\infty}\widehat{I}^n_{KSG}(X;\alpha T) = -\frac{1}{k}$, where k is the k-nearest neighbor parameter for the estimator. Thus, $\widehat{I}^n_{KSG}(X;\alpha T)$ need not be equal to $\widehat{I}^n_{KSG}(X;T)$.

Proof Sketch 3. Following from the proof of Proposition 2, we note that as $\alpha \to 0$, we have $n_{\alpha t,i,\infty} = \sum_{j \neq i} \mathbb{I}\{||\alpha T_i - \alpha T_j|| \leq \rho'_{k,i,p}\} \to n$. First, note that $\rho'_{k,i,p} = \rho_{k,i,p}$, as $\alpha \to 0$. Next, because X and T are bounded, and as $\alpha \to 0$, αT should contain all datapoints within the sphere of size $\rho_{k,i,p}$. Similarly, $n_{x,i,\infty} = k$ in this case, as the nearest neighbor distance is dominated by X, and there will be k datapoints within the nearest neighbor distance of $\rho'_{k,i,p}$, as $\rho_{k,i,p} = \rho'_{k,i,p}$. Thus we then have: $\lim_{\alpha,n \to \infty} \widehat{I}^n_{KSG}(X; \alpha T) = \psi(k) + \psi(n) - \frac{1}{k} + \frac{1}{n} \sum_{i=1}^{n} (\psi(k) + \psi(n))) = -\frac{1}{k}$.s

Lastly, as KSG is global scale-invariant (Proposition 2), we have that $\lim_{\alpha \to 0, n \to \infty} \widehat{I}^n_{KSG}(X; \alpha T) = \lim_{\alpha \to 0, n \to \infty} \widehat{I}^n_{KSG}(\frac{1}{\alpha} X; T) = \lim_{\alpha, n \to \infty} \widehat{I}^n_{KSG}(\alpha X; T) = -\frac{1}{k}$. The final result follows from the fact that $\widehat{I}^n_{KSG}(X; Y) = \widehat{I}^n_{KSG}(Y; X)$.

MINE: We first define two variants of the MINE estimator as follows:

MINE-Opt: This estimator refers to the MINE estimator where instead of training the neural network on the loss function defined in equation 7 by stochastic gradient descent (SGD), we pick the best neural network configuration that directly maximizes equation 7. Thus, we pick the global optimum.

MINE-SGD: This estimator refers to the MINE estimator where optimization of the loss function defined in equation 7, is performed using conventional stochastic gradient descent. This is the standard approach proposed originally by (Belghazi et al., 2018).

We denote the MINE-based MI estimators by $\widehat{I}^n_{MINE-opt}$ and $\widehat{I}^n_{MINE-sgd}$. We then have the following results.

Proposition 4. It holds that $\widehat{I}^n_{MINE-opt}(X; \alpha T) = \widehat{I}^n_{MINE-opt}(X; T)\ \forall \alpha \in \mathbb{R}^+$.

Proof Sketch 4. To demonstrate that $\widehat{I}^n_{MINE-opt}(X; \alpha T) = \widehat{I}^n_{MINE-opt}(X; T)$, we begin by considering any neural network function f that yields a specific value for the expression $\mathbb{E}_{X,Y \sim P(X,T)}[f(X,T)] - \mathbb{E}_{X,T \sim P(X) \times P(T)}\left[e^{f(X,T)}\right]$, there exists a corresponding neural network function f' such that $\mathbb{E}_{X,\alpha T \sim P(X,\alpha T)}[f'(X, \alpha T)] - \mathbb{E}_{X,\alpha T \sim P(X) \times P(\alpha T)}\left[e^{f'(X,\alpha T)}\right]$ has the same value of the expression involving f and vice-versa. To constructf', let W_T be the weights of the first layer of the network f that are attached to T, and similarly W'_T for f'. Define a new network function f' with the same architecture as f except that $W'_T = W_T/\alpha$. By construction, the function f' satisfies $f'(X, \alpha T) = f(X, T)$, which implies that for every function f that optimizes the expression in equation 7 there is a corresponding function f' for the variables X and αT. This also shows that the optimization for $I(X; T)$ and $I(X; \alpha T)$ as expressed in equation 7 is equivalent. As a result, the mutual information estimator $\widehat{I}^n_{MINE-opt}$, which corresponds to the supremum of the value of this expression over all possible neural network functions, is invariant under scaling of T. Therefore, we conclude that $\widehat{I}^n_{MINE-opt}(X; \alpha T) = \widehat{I}^n_{MINE-opt}(X; T)$.

Proposition 5. It holds that $\lim_{\alpha \to 0} \widehat{I}^n_{MINE-sgd}(X; \alpha T) = 0$. Thus, $\widehat{I}^n_{MINE-sgd}(X; \alpha T)$ need not be equal to $\widehat{I}^n_{MINE-sgd}(X; T)$.

Proof Sketch 5. Let f^* represent the neural network function which optimizes the expression for $\widehat{I}^n_{MINE-sgd}(X; \alpha T)$, which is $\mathbb{E}_{X,\alpha T \sim P(X,\alpha T)}[f'(X, \alpha T)] - \mathbb{E}_{X,\alpha T \sim P(X) \times P(\alpha T)}\left[e^{f'(X,\alpha T)}\right]$, via SGD. Let W_T be the weights of the first layer of the network f^* which are attached to αT. From Theorem 1 of (Ghosh et al., 2019), we have that $|W_T|^2 \leq \gamma \alpha^2 |T|^2$. Thus, as $\alpha \to 0$, the weights $W_T \to 0$ as well. This indicates that the contribution of T to the function f^*, as $\alpha \to 0$, will be negligible. Thus effectively, $\lim_{\alpha \to 0} \widehat{I}^n_{MINE-sgd}(X; \alpha T) = \widehat{I}^n_{MINE-sgd}(X; 0) = 0$. This proves the result.

4 Methodology

4.1 Normalization Strategies

In this section, we outline three normalization strategies that form the basis of our studies in this work. We define them as follows.

Definition 1. (Local Normalization) We are given a random variable $X = [x_1, x_2, .., x_d] \in \mathbb{R}^d$ where $x_i \in \mathbb{R}$. Let $X \sim P$ and $S = \{X_1, X_2, ..., X_n\} \sim P^n$. The locally normalized $X_{\sigma|S} = [x'_1, x'_2, .., x'_d] \in \mathbb{R}^d$ is constructed such that $x'_i = \frac{x_i}{\sqrt{\mathbb{E}\left[(x_i - \mathbb{E}[x_i])^2\right]}}$ for $1 \leq i \leq d$, where the expectations are over S.

Definition 2. (Global Normalization) Given a random variable $X \in \mathbb{R}^d$, let $X \sim P$ and $S = \{X_1, X_2, ..., X_n\} \sim P^n$. The globally normalized $X_{\Sigma|S} \in \mathbb{R}^d$ is then constructed as $X_{\Sigma|S} = \frac{X}{\sqrt{\mathbb{E}[\|X - \mathbb{E}[X]\|^2]}}$, where $\|.\|$ denotes the L_2 norm and the expectations are over S.

Definition 3. (Global L_∞ Normalization) Given a random variable $X \in \mathbb{R}^d$, let $X \sim P$ and $S = \{X_1, X_2, ..., X_n\} \sim P^n$. The globally L_∞ normalized $X_{\Sigma_\infty|S} \in \mathbb{R}^d$ is then constructed as $X_{\Sigma_\infty|S} = \frac{X}{\mathbb{E}[\|X - \mathbb{E}[X]\|_\infty]}$, where $\|.\|_\infty$ denotes the L_∞ norm and the expectations are over S.

Note that for any RV X, we denote by $X_{\sigma|S}$ and $X_{\Sigma|S}$ its locally and globally normalized versions respectively.

4.2 Studied Scale-Invariant Estimators

We are given the RVs $X \in \mathbb{R}^d$ and $T \in \mathbb{R}^m$, and sampled data $S = \{(X_1, T_1), (X_2, T_2), ..., (X_n, T_n)\} \sim P^n_{XT}$. All following estimates are for the MI between X and T, given S. With this, we propose the following normalization approaches for KSG and MINE estimators. We outline our approaches for scale-invariant KSG and MINE extensions in Table 1.

Table 1: Proposed scale-Invariant KSG and MINE variants

KSG	MINE
KSG-Local: $\widehat{I}^n_{KSG}(X_{\sigma\|S}; T_{\sigma\|S})$	**MINE-Local**: $\widehat{I}^n_{MINE}(X_{\sigma\|S}; T_{\sigma\|S})$
KSG-Global: $\max_{c \in 0.1, 0.2, .., 2} \left[\widehat{I}^n_{KSG}(X_{\Sigma\|S}; cT_{\Sigma\|S})\right]$	**MINE-Global**: $\widehat{I}^n_{MINE}(X_{\Sigma\|S}; T_{\Sigma\|S})$
KSG-Global-L_∞: $\max_{c \in 0.1, 0.2, .., 2} \left[\widehat{I}^n_{KSG}(X_{\Sigma_\infty\|S}; cT_{\Sigma_\infty\|S})\right]$	**MINE-Global-Corrected**: $\widehat{I}^n_{MINE}(\sqrt{d_X} X_{\Sigma\|S}; \sqrt{d_T} T_{\Sigma\|S})$

Remark 2. In addition to the above approaches, we compare the standard baselines of KSG and MINE. Furthermore, we also include a recent variant of KSG in our comparisons, called BI-KSG (Gao et al., 2017), which has smaller bias levels for highly correlated data. We do not include binning-based measures in our experimental results, as we find that they fare poorly for almost all of our studied cases. Thus, we only study the KSG and MINE variants empirically in this work. Also, note that c is a tunable parameter and the choice of the parameter c is fixed to the range between 0.1 and 2 for all experiments.

5 Additional Motivation for Normalization Variants

In this section, we provide both intuitive and empirical arguments for our proposed variants in the previous section. First, we provide intuitive and empirical reasons for when and why global normalization approaches could be preferred. Next, we provide a rationale for our proposed global normalization variants for KSG and MINE estimators.

5.1 Global over Local

KSG: We argue in this work that global normalization should be the preferred choice, especially for estimating MI of high dimensional data, such as high dimensional feature representations in neural networks. This is mostly because local normalization makes each variable equally important, which can detrimentally affect the k-nearest neighbor based estimation of MI in high dimensions. In the context of neural networks, where X represents the inputs and T the features, often T is very sparse (i.e., most values are near zero and irrelevant). By scaling these irrelevant dimensions to unit variance, it can lead to worse estimates of MI.

To investigate the above scenario, we conduct the experiment as follows. Given two RVs $X, T \in \mathbb{R}^2$ such that $T = X + \epsilon$, where $\epsilon \sim \mathcal{N}\left(0, \sigma^2 I_2\right)$ (I_2 is 2×2 identity matrix).

Next, a series of independent RVs represented by $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, ...\epsilon_k]$ where $\epsilon_i \sim \mathcal{N}\left(0, \sigma'^2\right)$ (simulated noise), were appended to the input X, and concatenated to become a $2+k$ dimensional RV $X' = [X, \boldsymbol{\epsilon}]$. Note that $I(X;T) = I(X';T)$. We impose the constraint that $\sigma' \ll \sigma$, as in neural networks, the irrelevant variables have less energy than the relevant ones (as a consequence of training). With this experimental setting, we simulate and plot the MI estimates of KSG, BI-KSG, KSG-Local, and KSG-Global-L_∞ in Figure 1, as a function of the noise dimension k. It is clear from the figure that while KSG and KSG-Global-L_∞ maintain their estimates, KSG-Local yields significantly lower estimates with more noise dimensions. This is mainly due to the fact that KSG-Local will scale up the added noise variables and increase their importance.

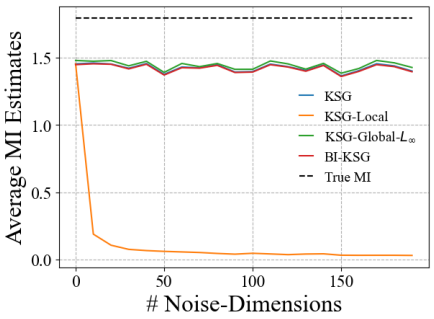


Figure 1: Average MI estimates for KSG-based measures for a varying number of noise dimensions.

MINE: We conduct a similar experiment to test whether MINE-Local also has similar artifacts as KSG-Local, as a result of assigning equal importance to low-energy noise dimensions. Here, we consider RVs $X, T \in \mathbb{R}^2$ which are correlated Gaussian variables with a correlation coefficient ρ randomly chosen from a certain range. Same as before, we then append noise variables $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, ...\epsilon_k]$ where $\epsilon_i \sim \mathcal{N}\left(0, \sigma'^2\right)$ to X to generate $X' = [X, \boldsymbol{\epsilon}]$. We then plot the average bias of all MINE variants, when the average is conducted over 10 trials for each choice of k. In each trial, we choose a random correlation co-efficient ρ. In this way, we can clearly see the average bias of all estimators over a broad range of correlation.

Intuitively, we might expect similar behavior as the KSG experiment, as when dimensionality d increases, the estimates of MI should reduce and the bias should become increasingly negative. However, our findings present an unexpected insight. As summarized in Figure 2, local normalization dramatically affects the bias positively rather than negatively in the case of MINE-Local. Specifically, the figure shows that as more noise dimensions are appended, the MINE-Local estimates tend to grow significantly beyond the true MI, whereas the other measures, including the MINE-Global variants, which remain stable. Our explanation is as follows. Unlike KSG, for MINE there is a neural network that actively seeks to maximize the DV objective. It is well known in the literature that neural networks can fit random noise data very well Zhang et al. (2017). Furthermore, as noise dimension increases, the overall data dimension increases and so does the number of network parameters, which enables the network to maximize the DV objective better. For MINE, we don't see an increase because the added variables are of very low energy, and thus the network's *effective* input dimensionality doesn't change as the added noise variables have a negligible impact on the output of the network.

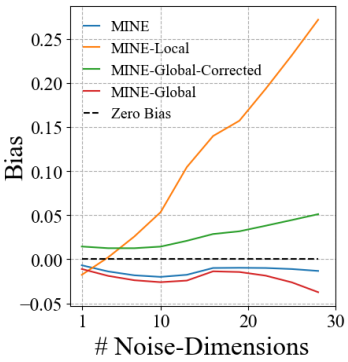


Figure 2: Bias of MINE-based measures for varying noise dimensions.

5.2 KSG-Global: Why the Maximization Step?

We outline two main arguments behind the maximization step for KSG variants in Table 1.

1. **Negative bias in high dimensions:** We find that the KSG estimator has a significant negative bias for data in high dimensions. More specifically, we find that the bias of the KSG estimator grows significantly with data dimension in the negative direction. We show this in two ways. First, we conduct an experiment where $X \in \mathbb{R}^d$ and $T \in \mathbb{R}^d$ are correlated Gaussians with a correlation coefficient ρ chosen such that the ground truth MI is fixed at a certain value (around 0.8). The dimensionality d is increased from 1 to 9 in steps of one. We chose to fix the ground truth MI across dimensions, as otherwise average MI would grow with data dimension, and we did not want the negative bias to be a result of ground truth MI growing faster than the KSG estimates. For each d, we run 20 trials, where in each trial 1000 data points were sampled from the joint distribution $P(X, T)$. We record KSG's average estimate of MI for each d, and the results

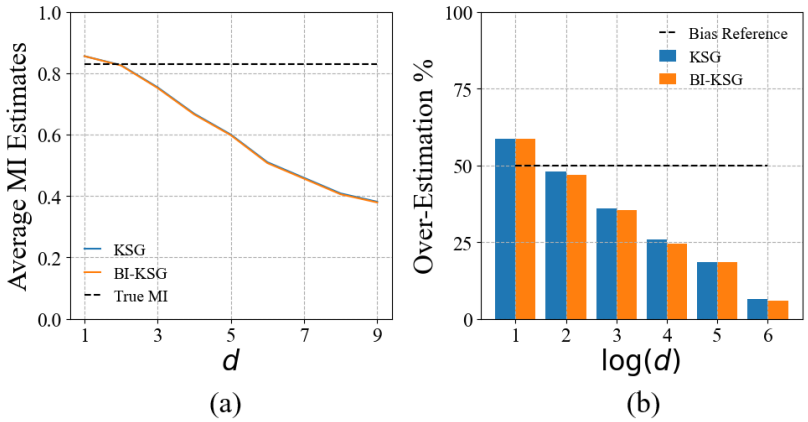


Figure 3: Analyzing the dependency of the bias of the KSG estimator with the data dimension. Please note that log here is in base 2.

are shown in Figure 3(a). It is clearly evident that the KSG estimate has a growing negative bias with dimensionality in this scenario.

To get a more general idea of the trends of the bias of the KSG estimator in response to increasing data dimension, we conduct another experiment. Here, same as before, $X \in \mathbb{R}^d$ and $T \in \mathbb{R}^d$ are correlated Gaussians. However, the correlation coefficient ρ is randomly chosen from a pre-determined range. Furthermore, we conduct 100 trials for each choice of dimensionality d, and in each trial, the estimator has access to 200 sampled datapoints from $P(X, T)$. We choose a broader range of dimensions, such that $d = [2, 4, 8, 16, 32, 64]$. Lastly, we plot the percentage of trials in which the estimated MI was greater than or equal to the ground truth MI. The results are shown in Figure 3(b). Note that for both KSG and BI-KSG, the proportion of samples where the estimated MI was lower than the ground truth MI increases significantly as d increases. To that end, we see that when $d = 64$, most estimates of MI are strictly less than the ground truth values. These two studies indicate that taking the maximum of multiple estimates of MI from KSG can potentially reduce the negative bias and improve accuracy, especially for high-dimensional data. After all, if $\widehat{I}^n_{KSG}(X_{\Sigma|S}, cT_{\Sigma|S})$ is always less than the true MI irrespective of c, the maximum value in these cases will always have the least bias. This is supported by our empirical results in Tables 2 and 3.

2. **Consequence of Proposition 3:** Proposition 3 finds that the KSG estimator for $I(X; \alpha T)$ converges to a negative value at either end of the scale spectrum w.r.t α. This motivated us to consider the maximum estimate of MI $\widehat{I}^n_{KSG}(X_{\Sigma|S}, cT_{\Sigma|S})$ across a range of scales in c. Note that as both $X_{\Sigma|S}$ and $T_{\Sigma|S}$ represent the global normalized versions of X and T, we can fix this pre-determined range of scales in c. We later see that the $\widehat{I}^n_{KSG}(X_{\Sigma|S}, cT_{\Sigma|S})$ follows an almost Gaussian like trend w.r.t c (Figure 6a).

Remark 3. Note that MINE has an implicit maximization over relative scales in the way it is optimized. This is mainly because the weights of the first layer can be any arbitrary set of real numbers as per the optimization objective. Furthermore, scaling the weights associated with one of the input RVs X or T is equivalent to scaling X or T respectively, as $(\alpha W)^T X = W^T(\alpha X)$. It is important to note that the maximization goes beyond just relative scales though, as the network function should ideally be invariant to affine transformations of the input. This suggests that MINE intrinsically considers a maximization of MI over all affine transformations of both variables. However, due to the nature of the gradient descent approach used to optimize the network, and its preference for flatter minima Keskar et al. (2017), this may not materialize to the fullest.

5.3 Motivation for Global Normalization Variants

In this section, we provide motivation for the specific global normalization variants proposed in this work: KSG-Global-L_∞ and MINE-Global-Corrected.

KSG: One of the objectives of global normalization is to put both RVs X and T on equal footing w.r.t nearest neighbor distances, such that the KSG estimator is not biased towards any one variable, which leads to low

and potentially even negative estimates (Proposition 3 and Figure 6a). In contrast, local normalization puts every dimension of X and T on an equal footing, which risks amplifying the impact of noisy and irrelevant dimensions, as demonstrated before. However, it is important to consider that KSG's nearest neighbor distances are computed using the L_∞-norm, instead of L_2-norm. Therefore, it is possible that KSG-Global may not put X and T on an equal footing w.r.t nearest neighbor distances. In fact, when $d_X \gg d_T$, the L_∞-nearest neighbor distances for $X_{\Sigma|S}$ will be significantly lower than for $T_{\Sigma|S}$. This is because global normalization will ensure that the average energy of dimensions sum to 1, and as $d_X \gg d_T$, the scale of individual dimensions in X will be significantly smaller than in T. As L_∞-norm only considers the largest element in the vector, this implies that L_∞ distances of $X_{\Sigma|S}$ can turn out much smaller than for $T_{\Sigma|S}$ in this case.

To illustrate this, we conduct an experiment, following the same setup as before with the Gaussian noise addition dataset. We have $X, T \in \mathbb{R}^2$ where $T = X + \epsilon$ and $\epsilon \sim \mathcal{N}(0, \sigma^2 I_2)$. We then increase the dimensionality of X by simply appending a number of its duplicates to yield $X' = [X, X, X, ..X]$(k times). This ensures that we preserve the distance structure of X in X'. Let the number of duplicate copies be denoted by k. Note that $I(X'; T) = I(X; T)$. For every k we undergo 10 trials, and in every trial we sample 200 data points from $P(X, T)$ and obtain MI estimates of KSG and KSG-Global variants. The results are shown in Figure 4. As hypothesized, we see that KSG-Global shows a clear reduction as k increases. In contrast, both KSG and KSG-Global-L_∞ are steady and have roughly consistent average MI estimates. This shows the importance of using L_∞-norm to estimate distances instead of L_2 in the case of KSG, as it uses L_∞-norm for estimating nearest neighbor distances. Next, we discuss the MINE variants.

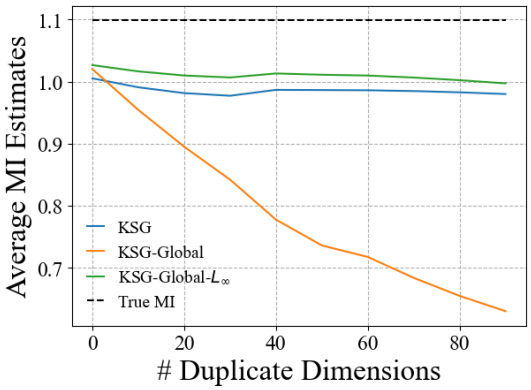


Figure 4: Data duplication: Average MI estimates for KSG-based approaches.

MINE: As discussed in KSG's case, global normalization can yield low individual energy per dimension if the data dimensionality is large. In the case of MINE, if $d_X \gg d_T$, we will have that $E_i[X_{\Sigma|S}(i)^2] \ll E_i[T_{\Sigma|S}(i)^2]$, where X_i and T_i denote individual dimensions of X and T respectively. In fact, $E_i[X_{\Sigma|S}(i)^2] = \frac{1}{d_X}$ and $E_i[T_{\Sigma|S}(i)^2] = \frac{1}{d_T}$. From the perspective of gradient descent and backpropagation, this implies that most error signals will focus on T, and X will be relatively neglected. Furthermore, if both d_X and d_T are large, the network input will have low energy per dimension, which may affect the optimization adversely. So, to avoid this, we rescale the global normalized data $X_{\Sigma|S}$ and $T_{\Sigma|S}$ to $X'_{\Sigma|S}$ and $T'_{\Sigma|S}$, such that the average energy of every dimension $E[X'_{\Sigma|S}(i)^2] = E[T'_{\Sigma|S}(j)^2] = 1$, $\forall i, j$. Thus, $X'_{\Sigma|S} = \sqrt{d_X} X_{\Sigma|S}$ and similarly $T'_{\Sigma|S} = \sqrt{d_T} X_{\Sigma|S}$, which yields the MINE-Global-Corrected approach. Note that rescaling still preserves the relative energies between different dimensions, i.e., $E[X(i)^2]/E[X(j)^2] = E[X'_{\Sigma|S}(i)^2]/E[X'_{\Sigma|S}(j)^2]$.

To showcase the importance of rescaling the globally normalized variables, we conduct an experiment where $X, T \in \mathbb{R}^d$ are correlated Gaussian variables with a correlation coefficient of ρ between the corresponding dimensions of X and T. Like in previous experiments, ρ is chosen randomly from a specified range for each trial. We vary the dimensionality d from 1 to 9, and for each d we conduct ten trials. In each trial, we generate 1000 data points from $P(X, T)$, and compare MINE estimates with its global normalization variants. For every d, we ultimately compute the average bias of each estimator. Results are shown in Figure 5. Our observations are two-fold. First, we observe that in general MINE estimates also yield a growing negative bias with larger input dimensionality. Next, we observe that MINE-Global grows negative at a faster rate compared to MINE, but MINE-Global-Corrected shows similar bias trends as MINE. The results imply that when the input signals are low due to global normalization,

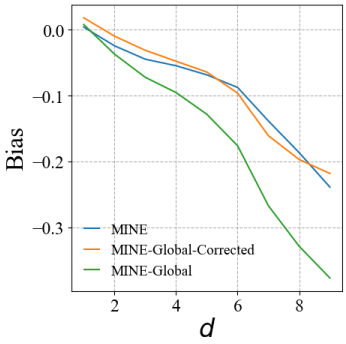


Figure 5: Estimator bias versus dimension: Comparing MINE with MINE-Global variants.

MINE tends to yield lower estimates on average. Also, they show that our rescaling approach is able to address this and yield similar bias levels as the original MINE.

6 Experimental Studies

6.1 Summary

Our empirical studies can be categorized into roughly three broad sections:

1. **Scale dependence and Signal to Noise Ratio (SNR) analysis of estimators:** We perform some basic tests and analyses of all estimators. First, we study their overall responses to scale changes, and then we study their responses to changes in noise levels (SNR).
2. **Accuracy analysis of estimators:** We conduct an extensive accuracy-bias-correlation analysis of all estimators in two different settings where ground truth MI is known. In each setting, we generate synthetic data using a diverse set of transformations to simulate different distribution scenarios.
3. **Studying neural network training using estimators:** We study the MI dynamics of neural networks during training. Specifically, we analyze the MI between input and features and compare the trends resulting from various estimators.

For our experiments, we use two *base* distributions for generating the random variables X and T. We refer to them in various parts of the experiments. They are as follows:

- **Correlated Gaussians:** Here, $X \in \mathbb{R}^d \sim \mathcal{N}(0, I_d)$ and $T \in \mathbb{R}^d \sim \mathcal{N}(0, I_d)$, and $E[X_i T_i] = \rho$ for $1 \leq i \leq d$ and $E[X_i T_j = 0]$ when $i \neq j$. Note that I_d denotes the identity matrix of size $d \times d$. This is a standard setting used in many prior MI estimation works.
- **Additive Gaussian Noise:** Here $X \in \mathbb{R}^d \sim \mathcal{N}(0, I_d)$ and $T = X + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2 I_d)$.

We used the NPEET MI estimator toolbox for estimating KSG and KSG-based measures [1]. For MINE, we used the popular pytorch-based package [2].

6.2 Scale and SNR analysis

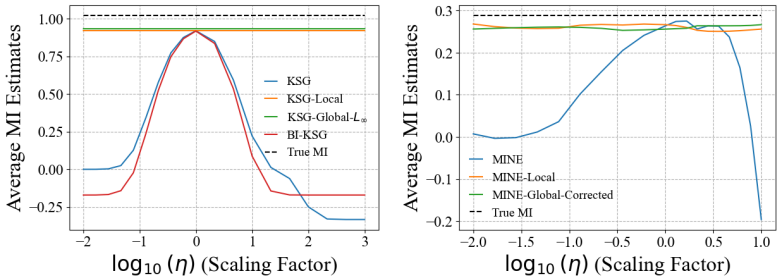
6.2.1 Scale

We conduct two sets of experiments. First, we generate X, T using a correlated Gaussian base. Then, we generate datasets following $P(X, T)$ across 20 trials. In each trial, we generate 1000 samples from $P(X, T)$. Using this set of 20 datasets, we construct many other copies of this set by scaling $X' = \eta X$, where η represents the scaling factor. We choose 20 different η between 10^{-2} and 10^3 for KSG, and between 10^{-2} and 10 for MINE, such that they are equispaced in a $\log_{10}$ scale. We choose different ranges for KSG and MINE, because the MINE estimates fall rapidly around $\eta = 10$ and yield highly negative values after that. For each η, we compute the average values of the estimators across the 20 datasets and report the average estimates as a function of η in Figures 6a and 6b. This concludes the first part of our experiments.

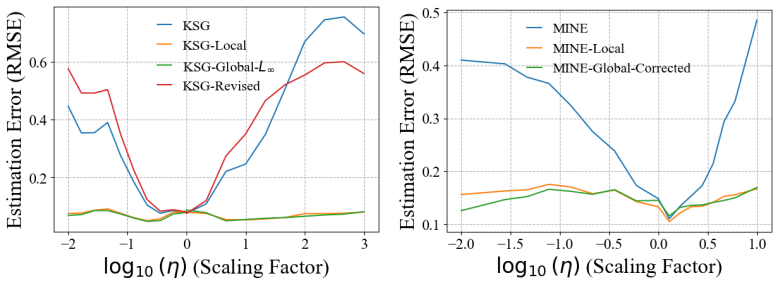
Next, we analyze the degree of estimation error, as the scale of X varies via η. To get a robust measure of error, we conduct 20 trials for every choice of η, and in each trial, we sample X, T from a correlated Gaussian base and set ρ randomly within a specified interval. After sampling X, T, we generate X', T by scaling X, as $X' = \eta X$. We then measure the root-mean-squared error (RMSE) between the MI estimates and the ground truth MI across the 20 trials and repeat the process for every choice of η. To get more general trends of error, for every choice of η, and for every trial, we choose to sample ρ randomly. This gives a wider range of ground truth MI. We plot the RMSE values for every measure for every scale factor η in Figures 6c and 6d.

Takeaways: We see that the standard MI estimators for KSG and MINE are significantly affected by scale. Furthermore, we see that KSG estimates converge to very low and even negative values as η reaches either extreme. Interestingly, as η grows, we find that KSG estimates indeed converge to around -0.33 which is $1/k$ as $k = 3$ for our experiments. This validates the result in Proposition 3. However, on the other side, when η reduces to very small values, we find that the estimates reach zero. This is because the estimator we

[1] https://github.com/gregversteeg/NPEET [2] https://github.com/gtegner/mine-pytorch



(a) Scale Trend: KSG (b) Scale Trend: MINE



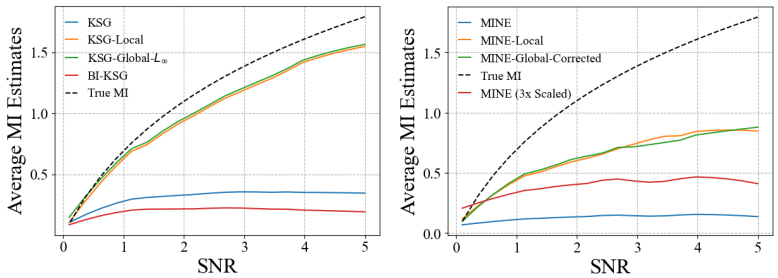
(c) Error v/s Scale: KSG (d) Error v/s Scale: MINE

Figure 6: Analysis of MI Estimators in response to data scaling. Estimates are for $I(\eta X; T)$, where η is the scaling factor.

used has a small distance correction in its k-nearest neighbor search. When we remove that correction, we find that the estimates converge to -0.33 for both extremes of η. We also see that MINE estimates converge to zero as η reduces. This validates our result in Proposition 5. For both KSG and MINE, we see that the local and global variants stay robust in terms of scale. For KSG, the values stay essentially level as η varies, but for MINE there are some small fluctuations. For both cases, we see that the local and global variants show significantly less estimation error as the scaling factor diverges from one.

6.2.2 SNR

In this section, we show how the scale dependence of MI Estimators can lead to other scenarios where they exhibit trends which are not ideal. We consider X, T sampled from the additive Gaussian noise base. Thus, we have $T = X + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. We additionally scale T to obtain $T' = 0.1T$. We scale down T to T' so that the scale-dependent bias of MI estimators becomes a factor in our experiments. Note that $I(X; T) = I(X; T')$, and thus when σ^2 increases, $I(X; T)$ should continue to decrease and vice versa. We vary σ^2 such that the Signal-to-noise ratio (SNR) ranges between 0 and 5. For every choice of SNR, we conduct 10 trials. In each trial, we generate 1000 samples of X, T' according to $P(X, T')$. Lastly, we average



(a) SNR Trend: KSG (b) SNR Trend: MINE

Figure 7: Average MI estimates for various estimators across different values of SNR. Estimates are for $I(X; \eta T)$, where T' is generated by adding Gaussian noise to X, the level of which controls the SNR. We fix $\eta = 0.1$.

the MI estimates for each estimator across the trials. The process is repeated for all values of SNR in this range. Results are shown in Figures 7a and 7b.

Takeaways: We first note that the global and local variants of the measures follow a similar trend compared to the ground truth MI; the average MI estimates grow with SNR. However, interestingly, we see that for the vanilla KSG and MINE estimators, their average MI estimates stop growing after a while and seem to converge. This shows that the scale-dependence of unnormalized estimators can potentially yield incorrect trends of true MI in other settings where scale of the variables can confound the true MI.

6.3 Comparing MI Estimators: Error Analysis

In this section, we undergo a comprehensive series of experiments, where we compute various error measures of all estimators on a diverse range of datasets. To create these datasets, we follow the two base distributions described in Section 6.1. After we've generated X, T according to the base distributions, we then make X undergo some (or none) of the following transformations, which are all MI preserving. For what follows, let $X \in \mathbb{R}^d$ and $T \in \mathbb{R}^d$.

1. **Randmat (rm):** $X' = \alpha W^T X$, where $\alpha \sim Unif(0,1)$ and $W \in \mathbb{R}^{d \times d}$ where $W(i,j) \sim Unif(0,1)$. $Unif(a,b)$ denotes a uniform distribution over $[a,b)$. If the randomly generated W is not invertible, we keep generating until we get an invertible W.
2. **Cube (cb):** $X' = X \circ X \circ X$, where $\circ$ denotes element wise multiplication (Hadamard Product).
3. **Sigmoid (sg):** $X' = \sigma(X)$, where $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ is such that $X'[i] = \frac{1}{1+e^{-X[i]}}$, where $X[i]$ denotes the i^{th} dimension of X and similarly for X'.
4. **Duplicate-self (ds):** $X' = [X, X, ...X] \in \mathbb{R}^{Kd}$. We set $K = 20$ in our experiments.
5. **Duplicate-noise (dn):** $X' = [X, \boldsymbol{\epsilon}] \in \mathbb{R}^{d+k}$, where $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, ...\epsilon_k]$ where $\epsilon_i \sim \mathcal{N}\left(0, \sigma'^2\right)$. We set $\sigma' = 0.2$ and $k = 20$.

Remark 4. Note that as each transformation is MI preserving, we can combine them in arbitrary ways and generate completely new transformations and data distributions. As we know the ground truth MI of the original base distribution, the transformed data will also have the same ground truth MI. This framework allows us to model a flexible set of distributions, which allow us to create high dimensional data with a low dimensional intrinsic dimension, which is often the case for neural network features. To illustrate, our data dimension can reach up to 200 dimensions, with very low intrinsic dimension (<10), compared to the experiments in Czyż et al. (2023) which go up to 25 dimensions. Furthermore, our choice of transformations is motivated by the choice of estimators tested in this work, and the normalization strategies compared in this work. For instance, local normalization typically performs poorly with added noise variables (duplicate-noise), and the KSG-Global variant isn't consistent in response to addition of duplicate dimension (duplicate-self). KSG itself is also affected by transformations such as sigmoid and cube as that can drastically change the distances in the nearest neighbor computation, and thus alter the structure of the data.

6.3.1 Studied Performance Measures

We study three different measures of performance in our experiments. For what follows, let $\hat{\mu}_1, \hat{\mu}_2, ...\hat{\mu}_k$ denote the estimated values of MI for any estimator across k trials, and let $\mu_1, \mu_2, ...\mu_k$ denote the ground truth values. With this, we summarize our performance measures as follows:

- **Normalized RMSE:** We first estimate the RMSE as $RMSE(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) = \sqrt{\mathbb{E}_i[(\hat{\mu}_i - \mu_i)^2]}$. Then we estimate a basline RMSE as $RMSE\_Base(\boldsymbol{\mu}) = \sqrt{\mathbb{E}_{i,j}[(\mu_i - \mu_j)^2]}$. With this, we can estimate the final measure as: $RMSE\_Norm(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) = \frac{RMSE(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu})}{RMSE\_Base(\boldsymbol{\mu})}$.
- **Spearman Correlation:** This is the Spearman correlation between $\hat{\boldsymbol{\mu}}$ and $\boldsymbol{\mu}$ (Zar, 2005). This is estimated as the Pearson's correlation coefficient between the rank values of $\hat{\boldsymbol{\mu}}$ and $\boldsymbol{\mu}$.
- **Bias:** We estimate the bias as $\mathbb{E}_i\left[\mu_i - \hat{\mu}_i\right]$.

Table 2: Comparing performance measures of MI Estimators: Additive Gaussian Noise Base

Transformation					d	measure	KSG-Based Measures					MINE-Based Measures			
rm	cb	sg	ds	dn			ksg	bi-ksg	ksg-loc	ksg-glo	ksg-glo-L_∞	mine	mine-loc	mine-glo	mine-glo-corr
	✓	✓	✓		2	rmse-norm	0.351	0.404	0.147	0.208	**0.110**	0.470	0.292	0.337	**0.278**
						spearman	0.985	0.985	0.983	0.982	0.983	**0.938**	0.920	0.889	0.928
						bias	-0.191	-0.226	-0.072	-0.105	**-0.047**	-0.246	-0.130	-0.161	**-0.117**
		✓	✓		2	rmse-norm	0.286	0.335	0.060	0.084	**0.050**	0.445	0.255	0.275	**0.233**
						spearman	0.992	0.992	0.988	0.989	0.990	**0.938**	**0.929**	0.917	**0.932**
						bias	-0.160	-0.192	-0.013	-0.035	**0.004**	-0.233	-0.108	-0.126	**-0.093**
	✓	✓			2	rmse-norm	0.458	0.533	0.145	**0.113**	**0.113**	1.036	**0.560**	0.658	**0.565**
						spearman	0.982	0.981	0.986	0.987	0.987	0.548	0.816	0.853	**0.884**
						bias	-0.253	-0.302	-0.071	**-0.050**	**-0.050**	-0.559	**-0.283**	-0.340	**-0.292**
		✓			4	rmse-norm	1.275	1.424	0.312	**0.300**	**0.301**	1.302	0.720	0.968	**0.684**
						spearman	0.988	0.991	0.993	0.995	0.995	0.805	**0.930**	0.901	**0.922**
						bias	-1.078	-1.235	-0.254	**-0.240**	**-0.240**	-1.105	-0.566	-0.781	**-0.543**
✓				✓	4	rmse-norm	0.862	0.932	0.445	0.594	**0.396**	0.930	0.438	0.803	**0.381**
						spearman	0.599	0.549	**0.995**	**0.990**	**0.994**	0.660	0.897	0.944	**0.965**
						bias	-0.653	-0.714	-0.363	-0.490	**-0.322**	-0.738	-0.296	-0.640	**-0.265**
			✓		4	rmse-norm	0.332	0.342	**0.304**	0.520	**0.297**	0.276	**0.369**	0.642	0.375
						spearman	0.994	0.994	0.995	0.994	0.994	0.931	0.922	0.922	**0.951**
						bias	-0.282	-0.298	-0.247	-0.426	**-0.239**	-0.122	**-0.223**	-0.494	**-0.232**
				✓	4	rmse-norm	0.332	0.342	1.327	**0.298**	**0.297**	0.622	0.423	0.895	**0.269**
						spearman	**0.994**	**0.994**	0.954	**0.995**	**0.994**	0.876	0.865	0.917	**0.942**
						bias	-0.282	-0.298	-1.107	**-0.239**	**-0.239**	-0.486	**0.002**	-0.719	0.118
✓	✓				4	rmse-norm	1.131	1.233	0.977	0.959	**0.931**	1.574	1.099	1.219	1.162
						spearman	0.588	0.534	**0.975**	0.961	**0.969**	0.542	**0.905**	0.837	0.769
						bias	-0.942	-1.045	-0.816	-0.796	**-0.773**	-1.198	**-0.914**	-1.012	-0.960
		✓		✓	4	rmse-norm	1.275	1.424	1.334	**0.300**	**0.301**	1.335	0.423	0.895	**0.286**
						spearman	**0.988**	**0.991**	0.916	**0.995**	**0.995**	0.807	0.870	0.914	**0.939**
						bias	-1.078	-1.235	-1.113	**-0.239**	**-0.240**	-1.119	**-0.004**	-0.718	0.148
	✓	✓		✓	6	rmse-norm	1.981	2.129	1.904	1.021	1.021	1.881	0.570	1.516	**0.437**
						spearman	0.969	0.966	0.905	**0.989**	**0.989**	0.852	0.786	**0.881**	0.855
						bias	-1.999	-2.174	-1.919	**-1.021**	**-1.021**	-1.898	**-0.179**	-1.502	**-0.181**
		✓			6	rmse-norm	1.983	2.131	**0.816**	**0.811**	**0.812**	1.843	1.147	1.535	1.185
						spearman	0.959	0.957	**0.995**	**0.995**	**0.995**	0.842	0.771	**0.862**	0.815
						bias	-2.001	-2.175	**-0.816**	**-0.809**	**-0.810**	-1.861	**-1.116**	-1.527	-1.155
	✓		✓		6	rmse-norm	1.377	1.408	1.343	1.605	1.290	1.213	0.991	1.444	1.004
						spearman	**0.983**	**0.981**	0.979	0.968	**0.989**	0.694	0.783	0.831	**0.881**
						bias	-1.377	-1.410	-1.347	-1.609	**-1.291**	-1.158	**-0.933**	-1.432	-0.971
✓			✓		6	rmse-norm	1.643	1.730	1.275	1.660	1.153	1.014	1.088	1.441	1.058
						spearman	0.338	0.314	0.937	0.871	**0.976**	0.775	0.798	**0.841**	0.818
						bias	-1.631	-1.730	-1.275	-1.666	**-1.150**	-0.967	-1.042	-1.431	-1.028
		✓		✓	6	rmse-norm	1.983	2.131	1.905	**0.814**	**0.816**	1.831	**0.517**	1.499	0.545
						spearman	0.956	0.955	0.926	**0.995**	**0.995**	0.813	0.806	0.811	**0.817**
						bias	-2.001	-2.175	-1.921	**-0.811**	**-0.814**	-1.847	**-0.068**	-1.471	0.391

Table 3: Comparing performance measures of MI Estimators: Correlated Gaussian Base

Transformation					N	d	measure	KSG-Based Measures					MINE-Based Measures			
rm	cb	sg	ds	dn				ksg	bi-ksg	ksg-loc	ksg-glo	ksg-glo-L_∞	mine	mine-loc	mine-glo	mine-glo-corr
					200	2	rmse-norm	0.125	0.125	**0.122**	**0.113**	**0.115**	0.573	0.579	0.601	**0.555**
							spearman	0.918	0.922	0.924	**0.938**	**0.936**	0.478	0.711	0.596	**0.801**
							bias	-0.021	-0.023	-0.020	**0.014**	**0.015**	**-0.238**	-0.247	-0.257	**-0.238**
			✓		200	2	rmse-norm	0.138	0.140	0.140	0.160	**0.116**	0.359	**0.321**	0.475	**0.315**
							spearman	**0.923**	**0.923**	0.914	0.898	0.878	0.886	0.929	**0.931**	**0.940**
							bias	-0.049	-0.051	-0.048	-0.042	**-0.005**	-0.114	**-0.089**	-0.192	-0.108
	✓		✓		200	2	rmse-norm	0.305	0.332	0.281	0.324	**0.233**	0.850	0.416	0.544	**0.405**
							spearman	0.794	0.798	**0.911**	**0.916**	0.867	0.586	0.889	0.907	**0.944**
							bias	-0.124	-0.146	-0.117	-0.125	**-0.077**	-0.282	**-0.149**	-0.227	**-0.156**
	✓	✓	✓		1000	2	rmse-norm	0.129	0.151	0.072	0.094	**0.060**	0.278	**0.135**	0.195	0.155
							spearman	**0.961**	**0.958**	0.941	**0.960**	**0.952**	0.951	0.942	**0.969**	**0.967**
							bias	-0.046	-0.064	-0.019	-0.023	**0.000**	-0.106	**-0.029**	-0.057	**-0.038**
		✓	✓		1000	2	rmse-norm	0.090	0.103	**0.046**	**0.052**	**0.049**	0.274	**0.130**	0.169	0.144
							spearman	0.947	0.950	**0.954**	0.943	**0.961**	0.957	0.941	**0.973**	**0.969**
							bias	-0.031	-0.041	**-0.004**	**-0.003**	0.017	-0.104	**-0.025**	-0.042	**-0.032**
	✓	✓			1000	2	rmse-norm	0.126	0.147	**0.065**	**0.060**	**0.060**	0.489	**0.281**	0.358	**0.290**
							spearman	0.940	0.935	0.939	**0.949**	**0.950**	0.879	**0.950**	**0.946**	**0.950**
							bias	-0.042	-0.060	-0.020	**0.000**	**0.000**	-0.213	**-0.105**	-0.145	**-0.112**
		✓			200	5	rmse-norm	0.962	1.088	0.481	**0.470**	**0.469**	1.014	0.949	0.982	**0.934**
							spearman	0.736	0.408	**0.964**	**0.966**	**0.969**	0.715	0.792	**0.852**	0.789
							bias	-0.664	-0.822	-0.322	**-0.293**	**-0.294**	-0.703	-0.641	-0.673	**-0.629**
✓					200	5	rmse-norm	0.738	0.783	0.688	**0.628**	**0.625**	0.974	**0.956**	0.993	**0.952**
							spearman	0.713	0.569	0.893	0.895	**0.905**	0.606	0.804	0.765	**0.833**
							bias	-0.442	-0.498	-0.406	**-0.349**	**-0.346**	-0.672	**-0.648**	-0.681	**-0.649**
	✓				200	5	rmse-norm	0.778	0.819	0.726	**0.695**	**0.693**	1.084	0.980	0.999	**0.967**
							spearman	0.933	0.931	0.932	0.937	0.939	0.605	0.773	**0.805**	0.711
							bias	-0.539	-0.589	-0.498	**-0.462**	**-0.460**	-0.784	-0.670	-0.688	**-0.660**
				✓	1000	5	rmse-norm	**0.258**	**0.259**	0.793	**0.253**	**0.253**	0.514	0.489	0.680	**0.285**
							spearman	**0.990**	**0.989**	0.812	**0.985**	**0.986**	**0.974**	0.847	**0.968**	0.954
							bias	-0.149	-0.152	-0.486	**-0.134**	**-0.134**	-0.304	0.171	-0.441	**0.076**
✓	✓				1000	5	rmse-norm	0.898	0.981	0.792	0.767	**0.744**	0.952	**0.767**	0.819	**0.766**
							spearman	0.582	0.433	0.943	**0.963**	**0.965**	0.511	**0.923**	**0.924**	0.902
							bias	-0.523	-0.615	-0.472	-0.456	**-0.440**	-0.680	**-0.526**	-0.567	**-0.533**
		✓			200	10	rmse-norm	1.381	1.473	1.008	0.999	1.000	1.441	1.357	1.424	1.352
							spearman	0.131	0.170	**0.956**	**0.962**	**0.963**	0.512	0.790	0.806	**0.856**
							bias	-1.351	-1.522	-0.985	**-0.957**	**-0.960**	-1.412	**-1.288**	-1.388	**-1.290**
	✓	✓		✓	200	10	rmse-norm	1.424	1.514	1.384	1.122	1.121	1.446	1.063	1.411	1.096
							spearman	-0.127	-0.083	0.751	0.935	**0.941**	0.590	0.005	0.771	**0.786**
							bias	-1.212	-1.383	-1.175	**-0.910**	**-0.909**	-1.418	**-0.451**	-1.362	-0.784
	✓		✓		200	10	rmse-norm	1.356	1.412	1.229	1.321	1.213	1.287	1.163	1.382	1.162
							spearman	0.831	0.823	0.872	0.885	**0.900**	0.652	0.933	**0.942**	0.916
							bias	-1.164	-1.263	**-1.043**	**-1.111**	**-1.011**	-1.182	-0.990	-1.325	**-0.976**
✓			✓		1000	10	rmse-norm	1.217	1.260	1.158	1.353	1.063	0.937	**0.886**	1.077	0.912
							spearman	0.815	0.641	**0.977**	0.947	**0.986**	0.956	0.954	**0.982**	0.969
							bias	-1.033	-1.120	-0.976	-1.145	**-0.890**	-0.869	**-0.844**	-1.030	-0.861
		✓		✓	1000	10	rmse-norm	1.424	1.516	1.386	**0.875**	**0.877**	1.185	0.808	1.090	**0.723**
							spearman	-0.065	-0.450	0.845	**0.992**	**0.992**	0.748	0.875	**0.951**	0.898
							bias	-1.212	-1.386	-1.177	**-0.721**	**-0.723**	-1.145	**0.444**	-0.981	0.560
✓				✓	1000	2	rmse-norm	0.141	0.159	0.093	0.078	**0.061**	0.280	**0.157**	0.235	**0.150**
							spearman	**0.968**	0.943	**0.967**	**0.964**	**0.967**	0.793	**0.953**	**0.960**	**0.950**
							bias	-0.040	-0.050	-0.031	-0.014	**0.000**	-0.105	**-0.017**	-0.085	**-0.027**

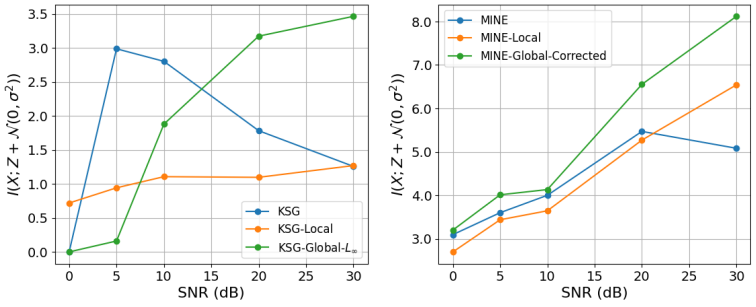
6.3.2 Experiment Summary and Takeaways

We summarize the empirical process for the results in Tables 2 and 3, as follows. For each experiment, we consider a specific set of transformations to be applied to X, which is shown in the first column of the tables. Once chosen, we then undergo 40 trials of data generation and MI estimation. In each trial, we generate N samples of $X, T \sim P(X, T)$ according to the base distribution, and then transform X according to the list of transformations in the corresponding row. For Table 2, we set $N = 1000$. The data dimensionality of X and T is represented via d and is shown in the tables. After generating N samples, we then obtain MI estimates from all estimators. Over the course of 40 trails, we then estimate the three different performance measures outlined in the previous section for all estimators. The only difference between Tables 2 and 3 is that Table 3 considers the correlated Gaussian base distribution, whereas Table 2 considers the additive Gaussian noise base distribution. Lastly, the red entries in the Tables refer to the case where the estimator error exceeds the base RMSE, yielding a normalized RMSE of greater than one. These results are thus not significant in terms of RMSE. However, even in these cases, we find that the MI estimates are often significantly correlated with the true MI (with Spearman correlation).

Takeaways: The main observations from the results are as follows:

- Overall, global and local normalization variants fare significantly better than the baseline measures.
- Our global normalization variants (MINE-global-corrected and KSG-Global-L_∞) overall fare better than other normalization strategies. In fact when the base distribution is additive Gaussian noise, we find that in most cases MINE-global-corrected and KSG-global-L_∞ outperform compared to the other normalization approaches.
- KSG-Global-L_∞ has very consistent performance, and across both settings, it seems to have the best performance in most cases. Even when the normalized RMSE estimates are insignificant (red entries), KSG-Global shows significant correlation with true MI in many of the cases.
- As discussed in our motivation, we find that overall the global normalization variants (MINE-Global-Corrected and KSG-Global-L_∞) perform better than their vanilla global normalization counterparts. This is much more apparent in the case of MINE.

7 Application of MI estimations in Deep Learning



(a) KSG (b) MINE

Figure 8: $I(X; Z + \mathcal{N}(0, \sigma))$ results with varying σ.

The analysis of neural networks' mechanisms remains a pivotal area of interest in deep learning research. MI as an important measure of the dependence between two variables, has been widely utilized to analyze the behavior of neural networks during training. In this section, we present a comprehensive analysis of MI measures on datasets, including IB dataset (Shwartz-Ziv & Tishby, 2017), MNIST (Deng, 2012) and CIFAR-10 (Krizhevsky & Hinton, 2009), estimating the MI during neural network training with original, local-normalized and global-normalized MINE and KSG estimators. Note that we implement a 20-neuron single-layer MINE estimator for both the MNIST and CIFAR-10 datasets, due to their complexity and the need for a stable estimator to track the trend of the mutual information. For the IB dataset, we employ a

two-layer MINE estimator with 30 neurons in each layer, as the relative simplicity of the IB dataset allows for the use of a more complex estimator to achieve stable results.

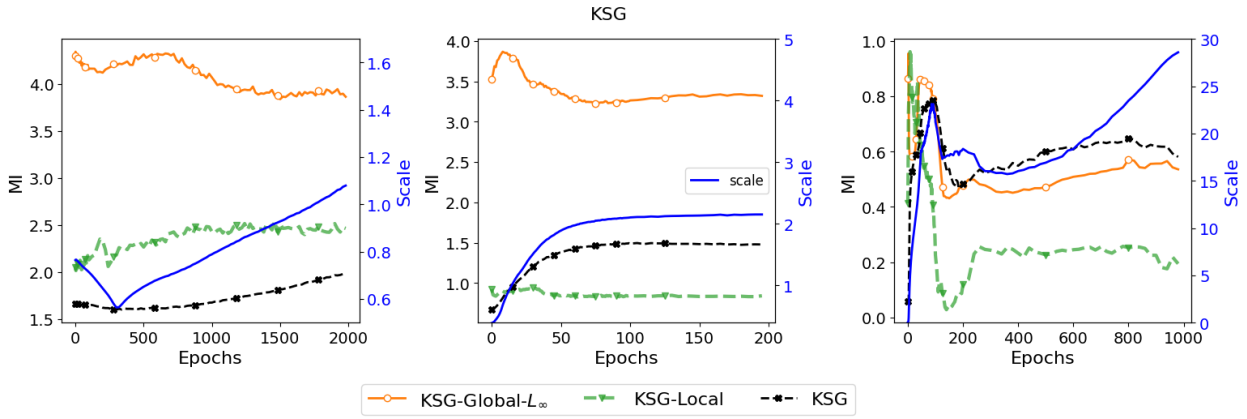
To build upon this analysis, we train three different networks for IB, MNIST, and CIFAR-10 datasets. For the MNIST and IB datasets, we replicate the network architectures from Saxe et al. (2018)'s work, using the widely-adopted *ReLU* activation function for the hidden layers. Specifically, for the IB dataset, we utilize a neural network with 7 hidden layers of dimensions 12-10-7-5-4-3-2. For the MNIST dataset, the neural network consists of 6 fully connected layers with dimensions 784-1024-20-20-20-10. For the CIFAR-10 dataset, we adopt a neural network with 4 convolutional layers, 3 fully connected layers, and batch normalization layers. The tasks for the MNIST and CIFAR-10 datasets involve classifying image inputs into their respective classes, while the task for the IB dataset involves training a binary decision rule based on 12 randomly distributed points. The networks are trained using SGD and cress-entropy loss. We train 2000 epochs for the IB dataset, 200 epochs for the MNIST dataset, and 1000 epochs for the CIFAR-10 dataset. Detailed architectures are provided in Appendix A.

Our analyses are summarized as follows. First, we train networks with the aforementioned configurations and extract the outputs of the selected intermediate layer, denoted as Z. For the IB dataset, the Z is extracted from the output of the third layer of the network, with a dimension of 7. For the MNIST dataset, the Z is extracted from the output of the third layer of the network, with a dimension of 20. For the CIFAR-10 dataset, the Z is extracted from the output of the Global Average Pooling layer of the network, with a dimension of 192. We analyze $I(X;Z)$ and $I(Z;Y)$ during neural network training from three perspectives:

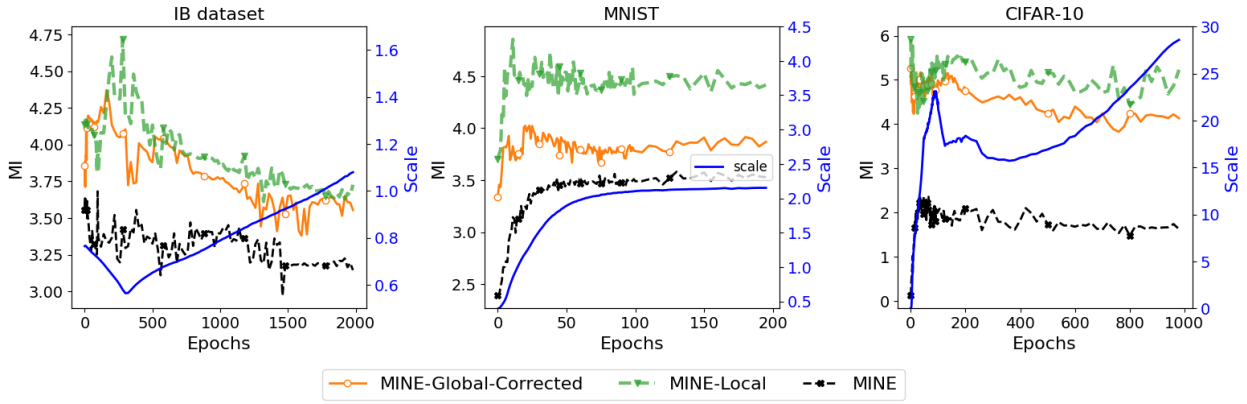
- **Impact of noise**: We examine how the MI changes when additive Gaussian noise is introduced in the intermediate layers of the network.
- **Training dynamics**: We investigate the changes of MI estimates $I(X;Z)$ over the course of training epochs.
- **Information plane visualization**: We plot $I(X;Z)$ against $I(Z;Y)$ to visualize the information plane, providing insights into the trade-off between the information preserved about the input X and the information relevant to the label Y.

Impact of noise in neural network training: We plot $I(X; Z + \mathcal{N}(0, \sigma^2))$ for trained networks, where a noise $N \sim \mathcal{N}(0, \sigma^2)$ is added before the Z layer. In these experiments, the signal-noise-ratio (SNR) quantifies the level of a signal relative to the level of background noise. Specifically, an SNR of a dB implies that with unit power, the noise variance is $10^{-a/10}$. Thus, as SNR increases, the noise level decreases, and intuitively the MI $I(X; Z + \mathcal{N}(0, \sigma^2))$ will increase due to the reduction in noise. As shown in figure 8, when SNR increases, the noise variance σ^2 decreases, increasing the dependence between X and Z. Consequently, the MI $I(X; Z + \mathcal{N}(0, \sigma^2))$ should increase. In figure 8, we observe that as the SNR increases, the mutual information does not change as initially anticipated for the original KSG and MINE estimators. Instead, their results initially increase and then decline as SNR continues to rise. Notably, both estimators with global normalization exhibit the most consistent trend, reflecting the expected increase in dependence between X and Z as noise is reduced.

$I(X;Z)$ versus the number of epochs: As we have already established that the KSG estimator is sensitive to data scale, we wanted to see to what extent this is the case during the training of neural networks. In figure 9, we present the change of $I(X;Z)$ during training, comparing the original KSG estimator, its local-normalized and global-normalized variants, and the MINE estimator and its variants. The displayed results represent the averages from 10 trials. We also simultaneously plot the scale of the features Z (i.e., $|Z|$), and its y-scale is placed on the right side of the figures. We find that in MNIST and CIFAR-10 datasets, $I(X;Z)$ from the original KSG estimator shows a significantly high correlation with the scale of the features during training, which hints that it may fundamentally capture the changes in feature scale. In contrast, the global-normalized estimate does not follow the scale curve and yields interesting trends. For the IB and CIFAR-10 datasets, the global-normalized estimate first increases and then decreases after training a certain number of epochs, thus being more adherent to the original fitting followed by the compression trend proposed by (Shwartz-Ziv & Tishby, 2017). In CIFAR-10, the decrease in $I(X;Z)$ happens right after 3



(a) KSG



(b) MINE

Figure 9: $I(X;Z)$ measures estimated after every epoch of training on IB, MNIST and CIFAR-10 datasets. Z represents the output of 3^{rd} layer for IB dataset and MNIST dataset, and 7^{th} layer for CIFAR-10 dataset.

epochs, which demonstrates a completely different trend than the baseline measures. Overall, for KSG, we note that in two of the three cases we see a clear fitting and compression phase as described in (Shwartz-Ziv & Tishby, 2017) using KSG-global-L_∞, which is not the case for other variants.

In figure 9, we find that the MI estimates obtained by MINE estimators are noisier. This may imply that MINE estimators may face challenges when dealing with high-dimensional variables that do not follow a standard distribution, particularly when the sample size is relatively small (N = 5000). However, we do observe fitting and compression for both local and global variants in IB and MNIST. In (Poole et al., 2019), authors highlighted that MINE often shows higher variance due to neural network training instability.

Information plane analysis: In figure 10, we plot the information plane for the IB, MNIST, and CIFAR-10 datasets using the KSG and MINE estimators to reveal MI changes in neural network training. The displayed results represent the averages from 10 trials. For IB and MNIST datasets, both $I(X;Z)$ and $I(Y;Z)$ obtained by the original KSG estimator generally increase with the number of training epochs. In contrast, the KSG-local and KSG-global-L_∞ estimators demonstrate a more refined information bottleneck trend. Specifically, these estimators show a clear fitting phase where $I(X;Z)$ initially increases and then stabilizes, followed by a compression phase where $I(X;Z)$ decreases while $I(Y;Z)$ remains monotonically increasing. Among these results, the KSG-global estimator yields the most consistent trends, exhibiting the most distinct fitting and compression phases in two of the three datasets, thus effectively capturing the information bottleneck phenomenon.

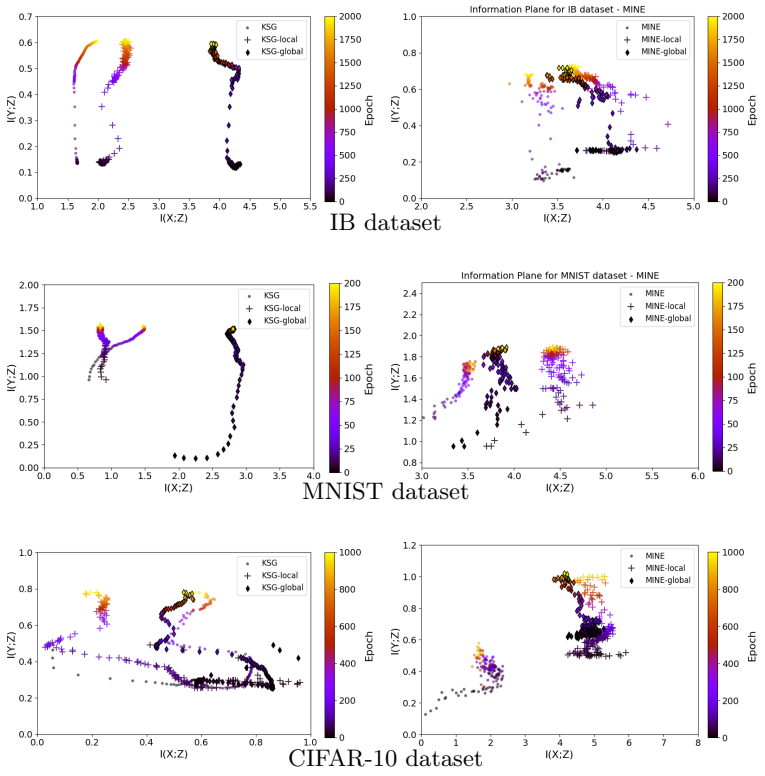


Figure 10: Information plane ($I(X;Z)$ against $I(Y;Z)$) for IB, MNIST and CIFAR-10 datasets. Z is the output of 3^{rd} layer for IB dataset and MNIST dataset, and 7^{th} layer for CIFAR-10 dataset.

The information plane plots using the MINE estimator for the IB, MNIST, and CIFAR-10 datasets reveal different levels of performance. The original MINE estimator fails to effectively capture the information bottleneck phenomenon in all three datasets. However, the MINE-global-corrected variant demonstrates an ability to observe the information bottleneck trend in all three datasets. The fitting and compression trends can also be observed for MINE-local, but the compression trends are harder to decipher clearly on MNIST and CIFAR-10. In general, we find that the MINE estimates are significantly noisier than the KSG estimates.

8 Conclusion

We presented a comprehensive study of scale invariance in MI estimators, and its impact on estimation accuracy, trends, and on MI based analysis of neural network training. We outlined multiple normalization approaches to combat scale changes, centered around KSG and MINE, and discussed the pros and cons of each approach. Specifically targeting the high-dimensional and low-data regime, intuitive and empirical arguments were given for each normalization approach and the final choice of estimators. Overall we found that while both local normalization and global normalization have their own strengths, in most practical scenarios, global normalization variants fare better. Both normalization strategies lead to desirable behaviour in response to input scale changes. Extensive experiments across two broad settings were conducted to measure the overall performance of each estimator. In almost all cases, the local and global normalization approaches fare much better than their unnormalized counterparts, while global normalization variants have the best performance overall. Lastly, on three real datasets, we studied the information plane dynamics w.r.t the hidden layer feature representations during training, for the unnormalized and normalized estimator variants. More clear trends of fitting and compression were observed with global normalization approaches in two out of the three datasets, with KSG-Global variants showing clearer trends than MINE-Global variants. Our work highlights the importance of scale-awareness in the problem of MI estimation, and its potential impact on MI estimates.

References

Milton Abramowitz. *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*. Dover Publications, Inc., USA, 1974. ISBN 0-486-61272-4.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In Jennifer Dy and Andreas Krause (eds.), *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 531–540. PMLR, July 2018.

Yuheng Bu, Shaofeng Zou, and Venugopal V. Veeravalli. Tightening mutual information based bounds on generalization error. In *IEEE International Symposium on Information Theory*, pp. 587–591, 2019.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, USA, 2006. ISBN 0-471-24195-4.

Paweł Czyż, Frederic Grabowski, Julia E Vogt, Niko Beerenwinkel, and Alexander Marx. Beyond normal: On the evaluation of mutual information estimators. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. Publisher: IEEE.

M. D. Donsker and S. R.S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time. IV. *Communications on Pure and Applied Mathematics*, 36(2):183–212, March 1983. ISSN 0010-3640. doi: 10.1002/cpa.3160360204. Publisher: Wiley-Liss Inc.

Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient Estimation of Mutual Information for Strongly Dependent Variables. In Guy Lebanon and S. V. N. Vishwanathan (eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pp. 277–286, San Diego, California, USA, May 2015. PMLR.

Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k-nearest neighbor information estimators. In *IEEE International Symposium on Information Theory*, pp. 1267–1271, 2017. doi: 10.1109/ISIT.2017.8006732.

Rohan Ghosh, Anupam K Gupta, and Mehul Motani. Investigating convolutional neural networks using spatial orderness. In *International Conference on Computer Vision Workshops*, pp. 2053–2056, 2019.

Ziv Goldfeld and Kristjan Greenewald. Sliced mutual information: A scalable measure of statistical dependence. *Advances in Neural Information Processing Systems*, 34:17567–17578, 2021.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

L. F. Kozachenko and Nikolai N. Leonenk. Sample estimate of the entropy of a random vector. *Problems of Information Transmission*, 23(2):9–16, 1987.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, June 2004. Publisher: American Physical Society.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In Silvia Chiappa and Roberto Calandra (eds.), *International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 875–884, August 2020.

Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003. Publisher: MIT Press.

Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5171–5180, 2019.

Dávid Pál, Barnabás Póczos, and Csaba Szepesvári. Estimation of Rényi entropy and mutual information based on generalized nearest-neighbor graphs. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. Fr-train: A mutual information-based approach to fair and robust training. In *International Conference on Machine Learning*, 2020.

Daniel Russo and James Zou. How much does your data exploration overfit? controlling bias via information usage. *IEEE Transactions on Information Theory*, 66(1):302–323, 2020.

Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Janett Walters-Williams and Yan Li. Estimation of Mutual Information: A Survey. In Peng Wen, Yuefeng Li, Lech Polkowski, Yiyu Yao, Shusaku Tsumoto, and Guoyin Wang (eds.), *Rough Sets and Knowledge Technology*, pp. 389–396, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-02962-2.

Rui Wang, Pengyu Cheng, and Ricardo Henao. Toward fairness in text generation via mutual information minimization based on importance sampling. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 4473–4485. PMLR, 25–27 Apr 2023.

Yaochen Xie, Ziqian Xie, Sheikh Muhammad Saiful Islam, Degui Zhi, and Shuiwang Ji. Genetic InfoMax: Exploring Mutual Information Maximization in High-Dimensional Imaging Genetics Studies. *Trans. Mach. Learn. Res.*, 2024, 2024.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020.

Jerrold H Zar. Spearman rank correlation. *Encyclopedia of biostatistics*, 7, 2005. Publisher: Wiley Online Library.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

A Appendix: Network Achitecture for Neural Network Analysis

Table 4: Model Architecture for IB Dataset

Layer	Dimension	Activation Function
Input	28×28	-
Flatten	12	-
Dense	10	ReLU
Dense	**7**	**ReLU**
Dense	5	ReLU
Dense	4	ReLU
Dense	4	ReLU
Dense	2	SoftMax

Table 5: Model Architecture for MNIST Dataset

Layer	Dimension	Activation Function
Input	28×28	-
Flatten	784	-
Dense	1024	ReLU
Dense	**20**	**ReLU**
Dense	20	ReLU
Dense	20	ReLU
Dense	10	SoftMax

Table 6: Model Architecture for CIFAR-10 Dataset

Layer	Dimension	Activation Function
Input	$32 \times 32 \times 3$	-
Conv2D	$32 \times 32 \times 96$	ReLU
Conv2D	$32 \times 32 \times 96$	ReLU
MaxPooling	$16 \times 16 \times 96$	-
Dropout	(0.5)	-
Conv2D	$16 \times 16 \times 192$	ReLU
Conv2D	$16 \times 16 \times 192$	ReLU
Global AveragePooling	**192**	**-**
Dense	512	ReLU
Dense	256	ReLU
Dense	20	SoftMax

In Table 5, Table 4 and Table 6, we present the network architecture and output dimensions for each layer of the neural networks used in our study. The layers with bold text are the layers for extracted Z.

For the IB dataset, we trained for 2000 epochs with an SGD optimizer and a learning rate of 5×10^{-3}. For the MNIST dataset, we trained for 200 epochs with an SGD optimizer and a learning rate of 5×10^{-4}. For the CIFAR-10 dataset, we trained for 1000 epochs with an SGD optimizer and a learning rate of 1×10^{-3}. The batch sizes were 256 for the IB dataset, 128 for the MNIST dataset, and 512 for the CIFAR-10 dataset.