NEURAL MUTUAL INFORMATION ESTIMATION WITH REFERENCE DISTRIBUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Estimating mutual information (MI) from data is a fundamental task in machine learning and data science, yet it remains highly challenging even with state-of-the-art estimators. This work proposes a new distribution-free MI estimator based on reference distributions. Unlike existing works that only discern between the joint distribution and the marginal distribution, which can easily overfit in high-MI settings, our method compares them with extra reference distributions. These artificial distributions share the same marginals as the original distributions but have known dependence structures, providing additional signals for more accurate dependency modelling. Experiments on synthetic tasks with non-Gaussian, high-dimensional data and real-world applications including Bayesian experimental design and self-supervised learning demonstrate the potential of our approach.

1 INTRODUCTION

Mutual information (MI), defined as the Kullback-Leibler divergence between the joint probability and the product of marginals, is a fundamental measure of the dependence between random variables. Unlike correlation, MI captures non-linear statistical dependencies between high-dimensional variables, making it a robust measure of dependence in diverse cases [1]. Consequently, MI has been widely applied across diverse fields, such as representation learning [2; 3; 4], Bayesian experimental design [5; 6], information bottleneck [7; 8], domain generalization [9; 10], and causality [11; 12].

Despite its widespread applications, estimating MI in high-dimensional spaces from raw data is highly challenging. Early methods work in a non-parametric fashion, utilizing techniques like binning or kernel density estimation [13; 14; 15; 16; 17]. As non-parameteric methods typically do not scale well with sample size or data dimensionality [18], recent studies have focused on utilizing powerful parametric models such as deep neural networks in MI estimation [19; 2; 20; 21; 22; 23; 24; 25]. In these estimators, a neural network (often known as the critic) is trained to discern or classify samples from the joint distribution and the marginal distribution. Upon convergence, it can be shown that the learned network corresponds to an estimate of the density ratio between the joint distribution and the product of marginals [19; 2; 20; 21; 22], or that the loss function of these estimators directly optimize a variational lower bound of MI [19; 2], thereby offering an estimate of MI. Compared to non-parametric methods, estimating MI through neural network training not only handles large sample sizes more effectively but also scales better with high-dimensional data. Thereby, it is increasingly gaining popularity in both machine learning and statistics communities.

Although neural network-based MI estimators have achieved success in numerous tasks, they are recently found to suffer from the so-called *high-discrepancy issue*, particularly when the underlying mutual information is high or the data dimensionality is high. In such scenarios, distinguishing samples from the joint and the marginal distributions becomes considerably easier, leading the network to easily overfit with limited data. This curse of high-discrepancy has been empirically observed in multiple applications [26; 22; 27] and has also been formally analyzed in theoretical studies [28; 29].

In this work, we propose Mutual Information Multinomial Estimation (MIME), a novel approach tailored for estimating mutual information accurately in high-discrepancy settings, drawing upon recent advances in multi-classes ratio estimation [22] and vector copula theory [30]. Unlike many existing methods that solely focus on discerning samples between the joint and the marginal distributions, which can easily overfit in high-discrepancy scenarios, MIME further compares them with

a set of reference distributions whose marginal distributions are the same as the original data distribution but exhibit various joint dependency structures. These additional comparisons encourage the network to learn a useful representation that captures the small differences in dependence structure, thereby effectively alleviating overfitting. To sum up, our main contributions are as follows:

- We propose a new mutual information estimator tailored for high-MI cases, combining the strengths of existing generative and distribution-free methods while avoiding their weaknesses;
- We systematically compare our estimator against state-of-the-art neural estimators on not only high-dimensional, non-Gaussian synthetic dataset with known MI but also real-world applications.

2 BACKGROUND

The mutual information (MI) between variables X and Y is defined as the Kullback-Leibler (KL) divergence between the joint distribution $p(x, y)$ and the product of the marginal $p(x)p(y)$

$$I(X;Y) = KL[p(x,y)\|p(x)p(y)] = \mathbb{E}\left[\log \frac{p(x,y)}{p(x)p(y)}\right]. \tag{1}$$

It is obvious that stronger dependence between X and Y leads to larger divergence between $p(x, y)$ and $p(x)p(y)$. Thereby, MI can measure the dependence between two random variables. Given a dataset $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^n$ drawn from the joint distribution $p(x, y)$, our goal in this work is to estimate $I(X;Y)$ from the empirical samples $\mathcal{D}$. Before delving into our approach, we first revisit some existing MI estimators and their limitations.

Distribution-free MI estimator. Many methods have been developed to estimate (1) from $\mathcal{D}$ without making any assumptions about the underlying probability distributions. One popular method is to utilize the Donsker-Varadhan (DV) representation of KL divergence [31]:

$$I(X;Y) \geq \sup_{f\in\mathcal{F}} \mathbb{E}_{p(x,y)}[f(x,y)] - \log \mathbb{E}_{p(x)p(y)}[e^{f(x,y)}] =: \hat{I}_{\text{DV}}(X;Y), \tag{2}$$

where $\mathcal{F}$ can be any class of function $\mathcal{F} : \mathcal{X} \to \mathbb{R}$ satisfying the integrability constraints of the theorem. A typical choice of $\mathcal{F}$ is the class of deep neural network. In such case, MI estimate reduces to training a neural network f to maximize the objective (2). Similarly, the NWJ estimator [32] makes use of the f-divergence lower bound to estimate MI: $I(X;Y) \geq \sup_{f\in\mathcal{F}} \mathbb{E}_{p(x,y)}[f(x,y)] - \mathbb{E}_{p(x)p(y)}[e^{f(x,y)-1}] =: \hat{I}_{\text{NWJ}}(X;Y)$. Both DV and NWJ estimators leads to a lower bound of MI. Note that the equalities in the two estimators hold if $\mathcal{F}$ is wide enough.

Alternatively, one can estimate MI by employing techniques from density ratio estimate [21; 33; 22]:

$$I(X;Y) \approx \mathbb{E}_{p(x,y)}[\log \hat{r}(x,y)] =: \hat{I}_{\text{ratio}}(X;Y) \tag{3}$$

where $\hat{r}(x, y) \approx p(x, y)/p(x)p(y)$ is the estimated density ratio. To learn r, one can adopt a process similar to GAN training [34; 35], where a classifier is trained to distinguish samples from the joint distribution $p(x, y)$ and samples from the marginal distribution $p(x)p(y)$. It can be shown that upon convergence, the output of the classifier estimates the log density ratio. This type of estimator is found more robust than the above DV-like estimators [3] and is widely used in information-theoretic representation learning [3; 36]. However, the MI estimated in this way is not a lower bound.

It is worth noting that these two lines of methods are closely related to each other. On one hand, it can be shown that the optimal f in DV and NWJ estimators is, up to some constant, the log density ratio $\log p(x, y)/p(x)p(y)$. Conversely, the estimated ratio $\hat{r}$ in ratio-estimating method can in turn be used in the DV representation to obtain a lower bound estimate of MI. Regarding the learning process, ratio-estimating method can be seen as optimizing a lower bound of the Jensen-Shannon divergence between $p(x, y)$ and $p(x)p(y)$, whereas DV and NWJ methods optimize KL divergence.

Curse of high discrepancy. Despite their conceptual advantages compared to non-parametric methods, distribution-free estimators often suffer from the so-called high-discrepancy issue in practical use. High-discrepancy issue typically happens when the dimensionality of data is high and/or the underlying MI is high, which results in a large KL between $p(x, y)$ and $p(x)p(y)$. However, it is shown that asymptotically, any distribution-free estimator of the KL divergence between two distributions p and q will inevitably suffer from an estimation variance that is exponential to the ground

truth value [29; 28]:

$$\mathbb{V}[\hat{KL}[p\|q]] \approx O(e^{KL[p\|q]}) \tag{4}$$

which implies that methods based on KL divergence estimation (e.g. DV, NWJ) will be impractical if the underlying MI is large or the intrinsic dimensionality of the data is high, both of which can lead to a large KL. The situation becomes even worse when we learn estimators with mini-batches. This issue raises concerns about the reliability of DV- and NWJ-like estimators in high-MI settings.

We highlight that the impact of the high discrepancy issue is beyond KL divergence-based estimators. Recent works have confirmed that high-discrepancy issue also exists in ratio-based estimator (3), where the classifier in the estimator tends to overfit when samples $x, y \sim p(x, y)$ and samples $x, y \sim p(x)p(y)$ are overly easy to discern [21; 37; 27; 22; 38]. One intuitive explanation is that when the gap between $p(x, y)$ and $p(x)p(y)$ is large, there exist many classifiers that can achieve almost perfect accuracy but have differently estimated ratios, leading to inaccurate estimate of MI.

3 Method

In this work, we propose a new MI estimator aiming to address the aforementioned high-discrepancy issue in distribution-free MI estimation. The key idea is, unlike existing works that solely compare the joint distribution $p(x, y)$ with the product of the marginal distributions $p(x)p(y)$, which easily causes overfitting, we also compare them with specifically designed reference distributions. These additional comparisons provide more fine-grained signals for accurately modelling the dependence structures of $p(x, y)$ and $p(x)p(y)$, allowing us to better model the dependence between X and Y.

Reference-based MI estimation. The core of our method is to formulate MI estimation as a multinomial classification problem, where we not only distinguish samples from $p(x, y)$, $p(x)p(y)$ but also samples from specific-designed reference distributions. Specifically, consider classifying samples x, y from four distributions $p_1(x, y), p_2(x, y), p_3(x, y), p_4(x, y)$ with a classifier h. This can be done by training h with the following objective function:

$$\begin{gathered}\max_h \ \mathcal{L}(h) = \mathbb{E}_{p(c)p_c(x,y)}\Big[\log \frac{e^{h_c(x,y)}}{\sum_{k=1}^4 e^{h_k(x,y)}}\Big], \\ p_1(x, y) = p(x, y), \quad p_2(x, y) = q(x, y), \quad p_3(x, y) = q(x)q(y), \quad p_4(x, y) = p(x)p(y).\end{gathered} \tag{5}$$

where h_c denotes the logit (i.e. the unnormalized class probability) computed for class c and $q(x, y)$ is some reference distribution with marginals $q(x), q(y)$. This reference distribution is a generative model learned from the data, which can be as simple as a Gaussian distribution that approximates $p(x, y)$ or can be as flexible as a diffusion model trained on the data [39]. A good choice of reference should make use of the inductive bias about the problem (MI estimation), which is discussed later.

Upon convergence, it can be shown that when the prior $p(c)$ is uniform, the optimal h satisfies [22]:

$$\log \frac{p_i(x, y)}{p_j(x, y)} = h_i(x, y) - h_j(x, y), \tag{6}$$

thus allowing the mutual information to be estimated as:

$$I(X; Y) \approx \hat{I}_h(X; Y) := \mathbb{E}_{p(x,y)}\Big[h_1(x, y) - h_4(x, y)\Big], \tag{7}$$

where the expectation on the RHS can be readily approximated by Monte Carlo integration. Note that this estimator is neither a lower or upper bound estimate[1].

The whole procedure for our reference-based MI estimator therefore include two independent steps:

- *Reference construction*. In this first step, we learn the reference $q(x, y)$ with data $(x, y) \sim p(x, y)$.
- *Classifier training*. This corresponds to training the classifier h using objective (5) given $q(x, y)$.

[1] Alternatively, one can use the DV representation (2) to estimate MI by setting $f(x, y)$ in (2) to be the estimated ratio: $f(x, y) = h_1(x, y) - h_4(x, y) \approx \log p(x, y)/p(x)p(y)$. This guarantees that the estimated MI is a lower bound of $I(X; Y)$, however it needs to obtain samples from $p(x)p(y)$, which is unneeded in (7).

which is highly modular and can be implemented with any reference distributions and any classifiers. In this procedure, the introduction of $p_2(x, y)$ and $p_3(x, y)$ does not affect the theoretical correctness of the ratio estimator (6); instead, they function as implicit regularization to alleviate overfitting.

This estimation strategy can be seen as a new paradigm for combining discriminative (distribution-free) methods [40; 21; 33; 22] and generative modelling [41; 30] for accurate MI estimation.

Choices of reference distributions. We next discuss the design of reference distributions. While any distributions can be used in principle, we focus on those that are *marginal preserving*, i.e. the marginals $q(x)$ and $q(y)$ of the reference distribution $q(x, y)$ should closely approximate those of the original distribution: $q(x) \approx p(x), q(y) \approx p(y)$. This property is motivated by the fact that MI is irrelevant to marginals, so a good reference should only differ from $p(x, y)$ in dependence structure; When asking the network h to distinguish among the various distributions in the above framework, these marginal-preserving references enforce the network to learning a useful representation to distinguish the small changes in dependence structures, leading to accurate MI estimate.

To this end, we design the reference distributions to be *vector Gaussian copula* (VGC) models [30]. It can be seen as either a degenerated case of a standard flow model [41; 42] or a generalization of the classic Gaussian copula model. Specifically, a VGC $q(x, y)$ is implicitly defined by its data generation process, where data $x, y \sim q_(x, y)$ can be seen as generating from the following process:

$$\begin{aligned} x = f(\epsilon_{\leq d}), \quad y = g(\epsilon_{>d}) \\ \epsilon \sim \mathcal{N}(\epsilon; 0, \Sigma) \end{aligned} \tag{8}$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ and $g : \mathbb{R}^{d'} \to \mathbb{R}^{d'}$ are two bijective functions and $\Sigma \in \mathbb{R}^{(d+d')\times(d+d')}$ is a p.s.d matrix where $\Sigma_{ii} = 1, \forall i$. Here $\epsilon_{\leq d}$ and $\epsilon_{>d}$ are the first d and the last d' dimensions of ϵ. Note that this model degenerates to a Gaussian copula when f and g are element-wise functions.

We implement the two bijective functions f and g by flow-based models e.g. [41; 42]. The parameters $\{f, g, \Sigma\}$ of VGC can then be learned by first training the flows f and g with data $x \sim p(x)$ and $y \sim p(y)$ respectively, then learning Σ as $\Sigma = \mathbb{E}[\epsilon\epsilon^\top]$ where $\epsilon_{\leq d} = f^{-1}(x)$ and $\epsilon_{>d} = g^{-1}(y)$.

We take the two reference distributions in (5) as two VGCs that approximates $p(x, y)$ and $p(x)p(y)$ respectively. We highlight the following two useful properties of VGC as reference distributions:

(a) *Reasonably close to $p(x, y)$ or $p(x)p(y)$*. The marginal distributions of a well-trained VGC are almost the same as those of $p(x, y)$ and $p(x)p(y)$, with its dependence structure a Gaussian approximation to that of $p(x, y)$ or $p(x)p(y)$, thereby is reasonably close to $p(x, y)$ or $p(x)p(y)$[2];

(b) *Easy to sample*. Sampling from a VGC can readily be done by (8). This allows us to generate arbitrarily many samples from the reference distributions when learning the network with (5).

In the theoretical analysis below, we analyze how these two properties contribute to an accurate estimate of MI. In the appendix, we further compare this reference with other choices of references.

4 Theoretic Analysis

In practice, we replace the expectation in (5) with N samples to learn our MIME estimator via: $\hat{h} = \arg\max_h \mathcal{L}^N(h)$. After training, we estimate $I(X; Y)$ via Monte Carlo estimation: $\hat{I}^N_{\hat{h}}(X; Y) := \frac{1}{N}\sum_{i=1}^N [\hat{h}_1(x_i, y_i) - \hat{h}_4(x_i, y_i)]$. In Proposition 1, we prove that our MIME estimator is consistent.

Proposition 1 (Consistency of reference-based MI estimate)**.** *Assuming that the classifier $h_c : X \times Y \to \mathbb{R}$ is uniformly bounded. For every $\varepsilon > 0$, there exists $N(\varepsilon) \in \mathbb{N}$, such that*

$$\left| \hat{I}^N_{\hat{h}}(X; Y) - I(X; Y) \right| < \varepsilon, \forall N \geq N(\varepsilon), a.s..$$

Proof. See Appendix A. □

This confirms that the probability that the estimated MI of MIME becomes arbitrarily close to the true MI converges to one, as the amount of data used in the proposed estimator approaches infinity.

[2] According to recent copula theory [30], the difference between two distributions $p(x, y)$ and $q(x, y)$ can be factorized into the difference in the (multivariate) marginals and the difference in the dependence structure.

Furthermore, we have the following formal result about the advantages of our proposed estimator.

Proposition 2 (Controlled error in reference-based MI estimate)**.** *Define* $\log \hat{r}_{i,j}(x,y) := h_i(x,y) - h_j(x,y)$ *with* h *being the classifier defined as above. Upon convergence, we have that:*

$$\left| \log \hat{r}_{1,4}(x,y) - \log \frac{p(x,y)}{p(x)p(y)} \right| \leq 3 \sup_i \left| \log \hat{r}_{i,i+1}(x,y) - \log \frac{p_i(x,y)}{p_{i+1}(x,y)} \right|,$$

Proof. See Appendix A. □

Proposition 2 tells that if the ratios $p(x,y)/q(x,y)$, $p(x)p(y)/q(x)q(y)$ and $q(x,y)/q(x)q(y)$ can all be estimated accurately in h, then the ratio $p(x,y)/p(x)p(y)$ estimated by the same network will also be accurate. This is exactly the case in our method: the first two ratios can be estimated accurately due to the reduced distributional discrepancies (see property (a)). For the ratio $q(x,y)/q(x)q(y)$, the discrepancy between these two reference distributions may be high, however we can generate infinite samples from them in learning (see property (b)), so this ratio can also be learned accurately.

We note that other choice of reference distributions, such as those in [22], may not enjoy the same benefit as our references. The fundamental reason is that it is difficult to find references that are respectively close enough to $p(x,y)$ and $p(x)p(y)$, but are also close to each other. Our generative references sidestep this challenge by not requiring the two references to have a small discrepancy.

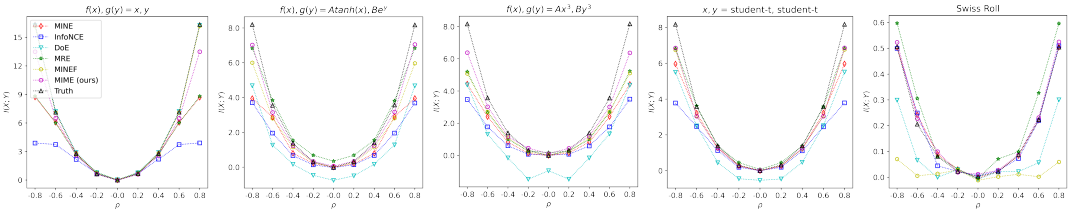
5 Related Works

Alleviating high-discrepancy issue in statistical divergence estimation. Several methods have been developed to address the high discrepancy issues in the estimation of statistical divergence the estimation of density ratio. One method is to reduce the estimation variance by clipping the output of the estimator [29], which can be seen as trading low variance with additional bias. Another line of methods solves the problem by the divide-and-conquer principle [21; 43], where the statistical divergence/density ratio between some intermediate distributions are first estimated, followed by an aggregation step to recover the original statistical divergence/density ratio from these intermediate estimations. Due to the reduced distributional discrepancy, each of these intermediate estimates can be expected to be accurate. However, divide-and-conquer methods are found to suffer from the *distribution-shift* issue during the aggregation step, which can lead to estimation inaccuracy. Unlike these methods, our method does not incur extra estimation bias and is free from distribution shift.

Multi-classes ratio/MI estimate. Similar to our work, some recent works also utilize a set of reference distributions q to improve density ratio estimate [44; 22], and the use of reference distributions in MI estimation have also been explored in [45; 46]. A crucial difference between our method and these approaches is that the reference distributions in our method are *marginal-preserving* i.e. their marginal distributions $q(x)$ and $q(y)$ are close to the original marginal distributions. As aforementioned, such marginal-preserving property is important as it enforces the network to focus on modelling the changes in dependence structures rather than modelling the marginal distributions, which are irrelevant for MI estimation. This importance will be verified in the experiment section.

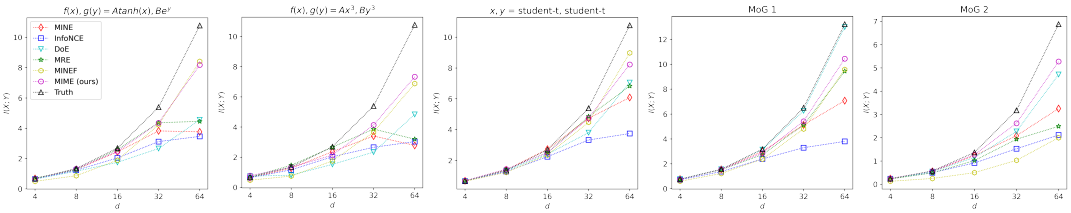
Generative methods for mutual information estimation. In addition to the above distribution-free methods for MI estimate, there also exist a set of MI estimators [47; 28; 48; 49; 39] based on density estimation or generative modelling. These methods work by learning a generative model (e.g. flow-based model [41; 42]) from data rather than comparing between different densities, thereby are free from the high-discrepancy issue. However, the performance of such estimators is dominated by the quality of the learned generative models, and generative modelling as done by e.g. flow-based model is well-known hard in high-dimensional cases [50; 51; 36; 42] and can suffer from model mis-specification. Our method also uses generative models, but only as informed reference distributions, which is less sensitive to model mis-specification and the quality of the learned generative models.

Vector Gaussian copula for MI estimate. We discuss a recent work [49] highly relevant to our method. This work first Gaussianizes the two marginals via two distint flow models, then approximates the joint distribution of the (marginally) Gaussianized data by a Gaussian distribution, from which mutual information is calculated. This process can be seen as modelling the original data distribution as a vector Gaussian copula model (8) and computing MI from this model. This way of MI calculation is accurate if the underlying data distribution can be well-approximated by a vector Gaussian copula, however it may be inaccurate in the case of model mis-specification. Our method on the contrary only use vector Gaussian copula as the reference, which is a more robust strategy.



(a) x, y (b) $\mathbf{A}\tanh(x), \mathbf{B}e^y$ (c) $\mathbf{A}x^3, \mathbf{B}y^3$ (d) Student-t (e) Swiss Roll

Figure 1: Comparison of different MI estimators under different ρ in four representative synthetic datasets. The dimensionalities of the data $X, Y \in \mathbb{R}^d$ in each case are 64, 32, 32, 32, 2 respectively.



(a) $\mathbf{A}\tanh(x), \mathbf{B}e^y$ (b) $\mathbf{A}x^3, \mathbf{B}y^3$ (c) Student-t (d) MoG 1 (e) MoG 2

Figure 2: Comparison of different MI estimators under $\rho = 0.7$ and various data dimensionality d.

6 Experiment

Baselines. We compare our new reference distribution-based MI estimator with four representative baselines in the field: MINE [19], InfoNCE [2], MRE [22], DoE [28] and MIENF [49]. Specifically:

- *Mutual Information Neural Estimate (MINE)*. This method estimates MI by the Donsker-Varadhan representation and a neural network f: $I(X;Y) = \sup_f \mathbb{E}_{p(x,y)}[f(x,y)] - \log \mathbb{E}_{p(x)p(y)}[e^{f(x,y)}]$.
- *InfoNCE*. Let $Y = \{y_1, \dots, y_{N-1}\}$ be $N-1$ negative samples from a proposal distribution, *InfoNCE* estimates MI by learning a neural network f to minimize the loss function $\mathcal{L} = -E_{p(x,y)}\left[\log \frac{f(x,y)}{\sum_{y_i \in Y \cup \{y\}} f(x,y_i)}\right]$, which is upper bounded by MI: $I(X,Y) \geq \log(N) - \mathcal{L}$.
- *Multinomial Ratio Estimate (MRE)*. A state-of-the-art method for density ratio estimation which also works by classifying samples from multiple distributions. The key difference between MRE and our method is the reference distributions in MRE are not marginal-preserving and generative.
- *Difference of Entropy (DoE)*. A generative method which works by first estimating the two entropies $H[Y]$ and $H[X|Y]$ with density estimators q, then calculate MI as $I(X;Y) = H[Y] - H[X|Y]$. Here, we study an improved variant of DoE where q is realized as a flow model [41]. This flow model is the same as the flow model in our method, which allows controlled comparison.
- *Mutual Information Estimate via Normalizing Flows (MIENF)*. A state-of-the-art generative method for MI estimation which works by first Gaussianizing the two marginals with two separately-trained flow models, then use a Gaussian distribution to approximate the (marginally) Gaussianized data, from which MI is analytically available. As discussed in Section 4, this way of MI estimation can be seen as assuming that the data distribution is a vector Gaussian copula (8).

Neural network architecture and optimizer. For fair comparison, we use the same network architecture for the critic $f(x,y)$ in all the MI estimators considered. This network takes the form of a MLP with 3 hidden layers, each of which has 500 neurons. A densenet architecture [52] is used for the network, where we concatenate the input of the first layer (i.e., x and y) and the representation of the penultimate layer before feeding them to the last layer. Leaky ReLU [53] is used as the activation function for all hidden layers. We train all networks by Adam [54] with its default settings, where the learning rate is set to be 5×10^{-4} and the batch size is set to be 512. Early stopping and a slight weight decay (1×10^{-6}) are applied to avoid overfitting. No dropout or batch normalization is used.

6.1 SYNTHETIC DATA

We first investigate the performance of our method on tractable simulated cases, where the mutual information is either analytically known or can be computed numerically up to very high precision.

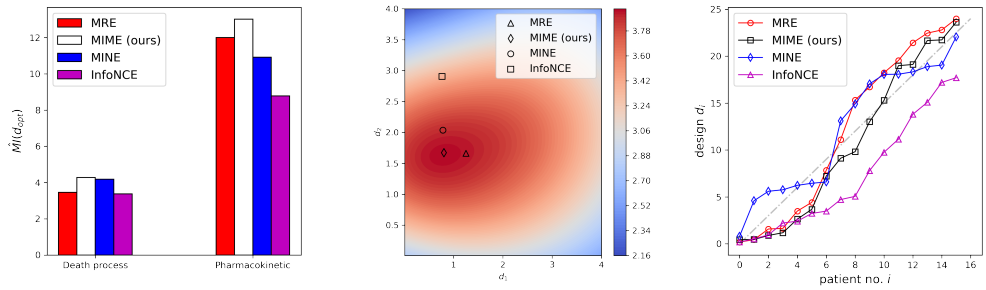
Setup. We consider estimating MI with data simulated from the following two classes of models. These benchmarks are similar to those in state-of-the-art benchmarks [55; 56] and can be seen as an instantiation of the so-called fine distributions [56] for principally evaluating MI estimators.

- *Nonlinear transformation of multivariate Gaussian*. The first model we consider is a generalized, high-dimensional version of the cubic Gaussian tasks widely used in literature [40; 29; 48]. Data $X \in \mathbb{R}^d, Y \in \mathbb{R}^d$ in this task is generated as follows:
$$x = f(\epsilon_{\leq d}), \qquad y = g(\epsilon_{>d}), \qquad \epsilon \sim \mathcal{N}(\epsilon; \mathbf{0}, \Sigma),$$
where Σ is a sparse convariance matrix satisfying $\Sigma_{i,i+d} = \rho, \Sigma_{i,i} = 1$ and $\Sigma_{i,j} = 0$ for any other $j \neq i + d, j \neq i$. $f : \mathbb{R}^d \to \mathbb{R}^d$ and $g : \mathbb{R}^d \to \mathbb{R}^d$ are some bijective functions. A difference between f, g in this work and those in [40; 29; 48; 55] is we further introduce two invertible matrices $\mathbf{A}, \mathbf{B}$ to entangle the dimensions of X and Y. Here ρ controls the dependence between X and Y. The ground truth MI in this model is given by $I(X; Y) = I(\epsilon_{\leq d}; \epsilon_{>d}) = -\frac{d}{2}\log(1 - \rho^2)$.
- *Mixture of multivariate Gaussians (MoG)*. The second case we consider is the mixture of M high-dimensional Gaussian distributions whose likelihood function being is as follows:
$$p(x, y) = \frac{1}{M}\sum_{k}^{M} \mathcal{N}([x, y]; \mu_k, \Sigma_k)$$
where $\Sigma_1 ... \Sigma_M$ is a set of sparse covariance matrices that are the same as in the non-linear Gaussian model. Note that the dependence factor ρ_k can be different for each Σ_k. Unlike the non-linear Gaussian model, the MI of this model is not analytically known; however, we can approximate it with Monte Carlo integration due to the availability of both the joint distribution $p(x, y)$ and the marginal distributions $p(x), p(y)$. The variance of MC integration vanishes given sufficient data.

We generate a total number of $n = 10^4$ data from each model and estimate MI from the generated data. Results for smaller sample size (e.g. $n = 2,500$ and $n = 1,250$ are shown in Appendix B). All experiment results are collected from 10 independent runs and we report their average values.

Varying the dependence level ρ. We first investigate whether our method can estimate MI accurately when we vary the dependence level ρ. Figure 1 shows the results for the non-linear Gaussian task with five typical choices of non-linear transformations f, g. These transformations are carefully chosen to represent a wide range of data patterns: Gaussian (case a), data with both bounded and exponential values (case b), data with long tails (case c, case d) and data confined to a low-dimensional manifold (case e); see Appendix B for more details. Estimators that plot closer to the ground truth MI curve are deemed more accurate. Overall, we see that there is no single method to rule in all cases. For example, while DoE performs well with Gaussian data (case a), it shows limitations in non-Gaussian cases (case b,c and e). Similarly, the InfoNCE method is effective when the data is low-dimensional (case e), yet it performs poorly in higher dimensional scenarios (case a,b,c,d). Unlike these methods, the proposed estimator consistently ranks within the top two methods across all scenarios, positioning it as a robust method that is widely applicable.

Varying the dimensionality d. In Figures 2, we further compare the performance of the various MI estimators under different data dimensionality. In this task, a higher d leads to a higher MI and thereby a more challenging task of MI estimation. From the figure, we see that the proposed MIME method clearly scales better w.r.t d than most methods, suggesting that it can better estimate MI in high MI settings. Although our method loses to DoE in one of the MoG task (case d), the margin is indeed small, and the MoG method performs unsatisfactorily in case a and case b. The poor performance of DoE may be due to the deficiency of the generative models used (i.e. the neural density estimator), that they can not accurately learn high-dimensional distributions in some cases. On the other hand, we also see that classic distribution-free methods like MINE and InfoNCE can not work reliably when the true MI is beyond 4nats, possibly due to the aforementioned high-discrepancy issue. Our method combines the strengths of both generative methods and distribution-free methods by using generative models as only the reference distributions in distribution-free methods, exhibiting as a strong method that are consistently reliable across diverse cases. Nonetheless, even with our powerful method, estimating MI in high-dimensional cases (e.g. $d = 64$) still presents a challenge.



(a) (b) (c)

Figure 3: Experiment results for BED. (a) Comparing the utility of the optimal design $\mathbf{d}^*$ found by different MI estimators. (b) Visualizing the contour of the underlying function between the utility and the design $\mathbf{d}$ in the death process task. (c) Visualizing the optimal designs found in the PK task. The results in (a) are collected from 5 independent runs and we report their average values.

6.2 Bayesian Experimental Design

We next consider the application of our method in Bayesian experimental design (BED), an important task in science and engineering. In BED, we wish to design experiments so that the data collected in the experiment will be maximally informative for understanding the underlying process of the experiment. For example, we may wish to find out the best policy design $\mathbf{d}$ for epidemic disease surveillance and intervention, so that we can infer the properties θ of the epidemic as accurate as possible with minimal observed data X. Such a task can be mathematically described as optimizing the conditional mutual information $I(X;\theta|\mathbf{d})$ between the experiment outcomes X (e.g. the observed dynamics of the infectious people in the population) and the properties of interest θ (e.g. the infectious rate of the epidemic) w.r.t the experiment design $\mathbf{d}$ (e.g. the intervention policy):

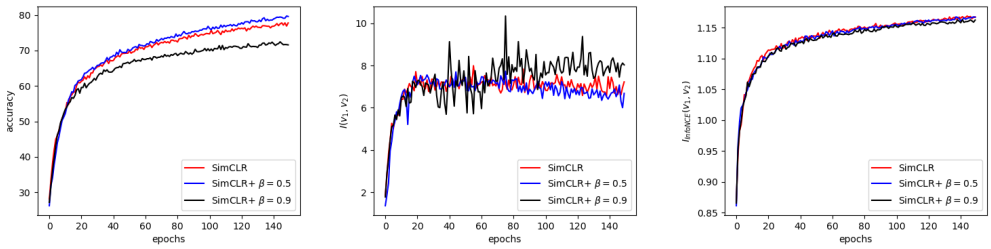
$$\max_{\mathbf{d}} I(X;\theta|\mathbf{d})$$

Setup. Many methods have been developed for finding the optimal design in BED [57; 6; 58; 5; 59; 60]. Here, we focus on the most general setup that the data generating process in the experiment is a black-box i.e. we know nothing about the details of the underlying data generation process but only the outcomes and the experiment design used. To find the optimal design $\mathbf{d}^*$ in such case, we use Bayesian optimization (BO) [61; 62] as in existing works to optimize $I(X;\theta|\mathbf{d})$ w.r.t the design $\mathbf{d}$. In this work, we use a total number of $T = 30$ BO steps to find the optimal design $\mathbf{d}^*$, where in each step we run $n = 3000$ experiments under design $\mathbf{d}$ to collect data $\mathcal{D} = \{x^{(i)}, \theta^{(i)}\}_{i=1}^n$ from the experiment. This data is then used to estimate $I(X;\theta|\mathbf{d})$ by a specific mutual information estimator.

We consider BED in two popular models that are widely used in epidemiology and medicine studies:

- *Death process model*. In this experiment, we model the dynamics of an epidemic in which healthy individuals become infected at rate θ. The design problem here is to choose observations times $\mathbf{d} \in \mathbb{R}^2$ at which to observe the number of infected individuals in two consecutive experiments, where in each experiment we observe the infection dynamics X of three independent populations.
- *Pharmacokinetic model*. In this experiment, we consider finding the optimal blood sampling time $\mathbf{d} \in \mathbb{R}^M, 0 \leq d_i \leq 24$ for a group of patients during a pharmacokinetic (PK) study. Each d_i corresponds to the sample time of each patient in the group. We would like to infer the properties of a specific drug demonstrated to the patients through the blood measurements. Here $M = 12$.

We are interested in seeing how the use of different MI estimators will affect the utility $\hat{I}(X;\boldsymbol{\theta}|\mathbf{d}^*)$ of the found design $\mathbf{d}^*$ as measured by MI. As the comparison on $\hat{I}(X;\boldsymbol{\theta}|\mathbf{d}^*)$ is only sensible when all methods estimate a lower bound of MI, we use the Donsker-Varadhan representation (2) rather than (7) to estimate MI in our method, where we set $f(x, y)$ in (2) as the log density ratio estimated by our method. This ensures that the MI estimated by our method is always a lower bound estimate.



(a) (b) (c)

Figure 4: Experimental results for SSL. (a) Classification accuracy on test set as evaluated by a linear classifier. (b) $I(v_1, v_2)$ as estimated by our method. (c) $I(v_1, v_2)$ as computed by InfoNCE.

Results. From Figure 3.(a), it is evident that our method consistently finds a better design $\mathbf{d}^*$ for both models, especially for the PK model where the underlying MI is high. In Figure 3.(b), we further visualize the underlying function between the utility and the design in the death process model. This function is modelled by a GP trained with 300 data. As one can see, the designs found by other methods are indeed suboptimal, being away from the optimum. In Figure 3.(c), we manually inspect the design strategies reported by different methods in the PK model. Compared to other methods, the blood sampling times suggested by our method is more uniformly distributed, which is sensible as a more diverse blood sampling time will allow us to collect richer information about the drug.

6.3 Self-supervised Learning

The third task we consider is the application of our MI estimator in self-supervised representation learning (SSL). SSL has been shown to be closely connected to infomax learning [63; 64; 65; 3; 4; 66]. For instance, [3; 4] showcase that maximizing the mutual information between different "views" of an image enables the acquisition of meaningful representations for downstream tasks. In this experiment, we aim to re-analyze SSL with our empowered estimator, investigating (a) how good can existing methods in SSL e.g. SimCLR [4] maximize the MI between different views of data; and (b) whether higher MI between views consistently results in better representations in SSL.

Setup. We focus on analyzing SimCLR, a classic method for SSL [4]. SimCLR learns representation by maximising the MI among various views of the same image by the InfoNCE loss [4; 2]

$$\max_{f,h} I_{\text{InfoNCE}}(h(f(v_1)), h(f(v_2)))$$

where f is an encoder (implemented as a ResNet-50 model [67]) used to compute the representation of an image. h is some projection function (often known as the projection head). Here the dimensionality of the representation is set to be $D = 1024$ and the batch size is set to be $B = 256$.

In addition to standard SimCLR where the estimate of MI is limited by the log batch size [68], we also consider a variant of SimCLR, SimCLR+, which can potentially attain a higher MI:

$$\max_{f,h} \beta \cdot I_{\text{MIME}}(f(v_1), f(v_2)) + (1-\beta) \cdot I_{\text{InfoNCE}}(h(f(v_1)), h(f(v_2)))$$

Here, we use the DV representation (2) to estimate MI in our method, where we set $f(x, y)$ in (2) as the log density ratio (3) estimated in our method. This ensures that our estimate is a lower bound. We adopt the common linear evaluation protocol to assess the quality of the learned representation.

High MI does not imply good representation. In Figure 4, we respectively plot (a) the test set accuracy; (b) the mutual information (MI) between different views of data as estimated by our method; (c) the MI estimated by InfoNCE. These figures are collected from a typical run on CIFAR10. By comparing SimCLR and SimCLR+ with $\beta = 0.9$, we see that a high MI may not necessarily be a good sign of better representation: SimCLR+ clearly attains a higher MI but has a significantly lower test accuracy. The weak correlation between high MI and representation quality is also evidenced by the trends of the curves in Figure 4.(a) and Figure 4.(b), where the test accuracy continues to grow even when there is no increase on MI. The results echo with previous studies [63] which stated that the success of SSL may only loosely relate to MI maximization under the linear evaluation protocol.

	MINE	InfoNCE	MRE	MIENF	MIME
$\hat{I}(K;T)$	0.0 ± 0.0	0.1 ± 0.0	0.6 ± 0.1	0.3 ± 0.1	0.8 ± 0.1
$\hat{I}_{\text{shuff}}(K;T)$	0.1 ± 0.1	0.0 ± 0.0	0.1 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

(a) **kinase-target interaction**

	MINE	InfoNCE	MRE	MIENF	MIME
$\hat{I}(L;R)$	1.3 ± 0.2	1.6 ± 0.2	1.9 ± 0.2	1.3 ± 0.1	2.8 ± 0.1
$\hat{I}_{\text{shuff}}(L;R)$	0.0 ± 0.1	0.0 ± 0.0	0.1 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

(b) **ligand-receptor interaction**

Table 1: Quantifying dependence between participants of protein interactions. 'shuff' corresponds to cases where the data is shuffled to break dependence. Results obtained from 5 independent runs.

InfoNCE is useful for MI maximization despite underestimation. Another interesting result is seen by comparing Figure 4.(b) and Figure 4.(c), where we observe a significant gap between the MI estimated by our method and that by InfoNCE. As our estimate here is a lower bound of the true MI, this suggests that InfoNCE actually largely underestimates the true MI between different views of data[3]. Interestingly, despite InfoNCE largely underestimates MI, optimizing InfoNCE loss does lead to an effective maximization of the true MI (at least at the early stage of learning). This suggests that to maximize MI, an accurate estimation of its exact value is indeed unnecessary.

6.4 Protein sequence analysis

Our last task involves quantifying dependence in protein language model (PLM) embeddings [69]. The goal is to investigate to what extent do the PLM embeddings contain information about protein-protein interactions, which are essential for the study of their functional property. This information is quantified by the mutual information between the embeddings of different protein sequences.

Setup. Following the setups in [70], we consider two types of protein-protein interactions: kinase-target and ligand-receptor interactions, using annotated data from the OmniPath database [36]. For kinase-target interaction, we consider kinase and target sequence embeddings as random variables $K \in R^{1024}$, $T \in R^{1024}$ with joint distribution $p(K,T)$, and are interested in evaluating $I(K;T)$. Similarly, we are interested in evaluating $I(L;R)$ for ligand-receptor interactions. All data are preprocessed using the auto-encoder approach in [70] to reduce their dimensionality to 32 before MI estimation. As in previous experiments, we ensure that all MI estimates are lower bound estimates. If PLM embeddings capture interaction information, MI between interaction partners should be significantly above 0 nats, and we investigate whether existing estimators can detect this information.

Results. For this task, our method estimates $I(K;T) \approx 0.8$nats and $I(L;R) \approx 2.8$nats. These results indicate that protein sequences as encoded by the PLM embeddings contain information about both interactions, with the information about ligand-receptor interactions much higher than kinase-target interactions — a result consistent with recent studies [70]. Methods like MINE largely underestimate this information and wrongly suggest independence for kinase-target interaction.

7 Conclusion

In this paper, we propose a new method for mutual information (MI) estimation in high-MI settings. A main challenge in this task is the high-discrepancy issue, which causes direct comparison between the joint distribution and the marginal distribution inaccurate. We address this issue by additionally comparing these distributions with their vector copula approximations, which share the same marginal distributions as the original data distribution and have Gaussian dependence structures. Experiments on diverse tasks demonstrate that such additional comparisons provide fine-grained signals for accurately modelling the true dependence in data, which leads to better MI estimate.

Despite our powerful method, we note that MI estimate is still challenging with high-dimensional non-Gaussian data, as can be seen from the simulation studies in Section 6.1. Furthermore, we find that to maximize MI, an accurate estimate of its value may indeed be unnecessary — a lesson learned from the SSL task. Together, these results highlight the values of developing proxies of MI for information-theoretic representation learning such as sliced mutual information [71; 72; 27; 73].

For future work, we consider exploring other choices of generative models, such as those in [19], as more powerful references, as well as extending our analysis on image benchmarks analysis [74; 75].

[3]We have included the constant $\log B$ when we compute the MI in InfoNCE (here B is the batch size).

REFERENCES

[1] Justin B Kinney and Gurinder S Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, 111(9):3354–3359, 2014.

[2] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

[3] 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. *arXiv preprint arXiv:1808.06670*, 2018.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[5] Steven Kleinegesse and Michael U Gutmann. Bayesian experimental design for implicit models by mutual information neural estimation. In *International conference on machine learning*, pages 5316–5326. PMLR, 2020.

[6] Adam Foster, Desi R Ivanova, Ilyas Malik, and Tom Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. In *International conference on machine learning*, pages 3384–3395. PMLR, 2021.

[7] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE information theory workshop (ITW)*, pages 1–5. IEEE, 2015.

[8] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[9] Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Dongsheng Li, Kurt Keutzer, and Han Zhao. Invariant information bottleneck for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7399–7407, 2022.

[10] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 29:3993–4002, 2020.

[11] Victor Solo. On causality and mutual information. In *2008 47th IEEE Conference on Decision and Control*, pages 4939–4944. IEEE, 2008.

[12] Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. *Advances in Neural Information Processing Systems*, 31, 2018.

[13] Andrew M Fraser and Harry L Swinney. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33(2):1134, 1986.

[14] Georges A Darbellay and Igor Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45(4):1315–1321, 1999.

[15] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[16] Young-Il Moon, Balaji Rajagopalan, and Upmanu Lall. Estimation of mutual information using kernel density estimators. *Physical Review E*, 52(3):2318, 1995.

[17] Nojun Kwak and Chong-Ho Choi. Input feature selection by mutual information based on parzen window. *IEEE transactions on pattern analysis and machine intelligence*, 24(12):1667–1671, 2002.

[18] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *Artificial intelligence and statistics*, pages 277–286, 2015.

[19] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

[20] Ruizhi Liao, Daniel Moyer, Polina Golland, and William M Wells. Demi: Discriminative estimator of mutual information. *arXiv preprint arXiv:2010.01766*, 2020.

[21] Benjamin Rhodes, Kai Xu, and Michael U Gutmann. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 33:4905–4916, 2020.

[22] Akash Srivastava, Seungwook Han, Kai Xu, Benjamin Rhodes, and Michael U Gutmann. Estimating the density ratio between distributions with high discrepancy using multinomial logistic regression. *Transactions on Machine Learning Research*, 2023(3):1–23, 2023.

[23] Qing Guo, Junya Chen, Dong Wang, Yuewei Yang, Xinwei Deng, Jing Huang, Larry Carin, Fan Li, and Chenyang Tao. Tight mutual information estimation with contrastive fenchel-legendre optimization. *Advances in Neural Information Processing Systems*, 35:28319–28334, 2022.

[24] Xu Wang, Ali Al-Bashabsheh, Chao Zhao, and Chung Chan. Smoothed infonce: Breaking the log n curse without overshooting. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 724–729. IEEE, 2022.

[25] Zhengyang Hu, Song Kang, Qunsong Zeng, Kaibin Huang, and Yanchao Yang. Infonet: Neural estimation of mutual information without test-time optimization. In *Forty-first International Conference on Machine Learning*.

[26] Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aaron Van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. In *Advances in Neural Information Processing Systems*, pages 15604–15614, 2019.

[27] Yanzhi Chen, Michael U Gutmann, and Adrian Weller. Is learning summary statistics necessary for likelihood-free inference? In *International Conference on Machine Learning*, pages 4529–4544. PMLR, 2023.

[28] David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR, 2020.

[29] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2019.

[30] Yanqin Fan and Marc Henry. Vector copulas. *Journal of Econometrics*, 234(1):128–150, 2023.

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

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

[33] Sudipto Mukherjee, Himanshu Asnani, and Sreeram Kannan. Ccmi: Classifier based conditional mutual information estimation. In *Uncertainty in artificial intelligence*, pages 1083–1093. PMLR, 2020.

[34] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[35] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.

[36] Yanzhi Chen, Dinghuai Zhang, Michael Gutmann, Aaron Courville, and Zhanxing Zhu. Neural approximate sufficient statistics for implicit models. *arXiv preprint arXiv:2010.10079*, 2020.

[37] Kristy Choi, Madeline Liao, and Stefano Ermon. Featurized density ratio estimation. In *Uncertainty in Artificial Intelligence*, pages 172–182. PMLR, 2021.

[38] Lukas Gruber, Markus Holzleitner, Johannes Lehner, Sepp Hochreiter, and Werner Zellinger. Overcoming saturation in density ratio estimation by iterated regularization. *arXiv preprint arXiv:2402.13891*, 2024.

[39] Giulio Franzese, Mustapha BOUNOUA, and Pietro Michiardi. Minde: Mutual information neural diffusion estimation. In *The Twelfth International Conference on Learning Representations*.

[40] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540. PMLR, 2018.

[41] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

[42] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2022.

[43] Kristy Choi, Chenlin Meng, Yang Song, and Stefano Ermon. Density ratio estimation via infinitesimal classification. In *International Conference on Artificial Intelligence and Statistics*, pages 2552–2573. PMLR, 2022.

[44] Lantao Yu, Yujia Jin, and Stefano Ermon. A unified framework for multi-distribution density ratio estimation. *arXiv preprint arXiv:2112.03440*, 2021.

[45] Chung Chan, Ali Al-Bashabsheh, Hing Pang Huang, Michael Lim, Da Sun Handason Tam, and Chao Zhao. Neural entropic estimation: A faster path to mutual information estimation. *arXiv preprint arXiv:1905.12957*, 2019.

[46] Dor Tsur, Ziv Aharoni, Ziv Goldfeld, and Haim Permuter. Neural estimation and optimization of directed information over continuous spaces. *IEEE Transactions on Information Theory*, 69(8):4777–4798, 2023.

[47] Bao Duong and Thin Nguyen. Diffeomorphic information neural estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7468–7475, 2023.

[48] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pages 1779–1788. PMLR, 2020.

[49] Ivan Butakov, Aleksander Tolmachev, Sofia Malanchuk, Anna Neopryatnaya, Alexey Frolov, and Kirill Andreev. Mutual information estimation via normalizing flows. *arXiv preprint arXiv:2403.02187*, 2024.

[50] Oren Rippel and Ryan Prescott Adams. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.

[51] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016.

[52] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[53] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models.

[54] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[55] 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.

[56] Paweł Czyż, Frederic Grabowski, Julia E Vogt, Niko Beerenwinkel, and Alexander Marx. The mixtures and the neural critics: On the pointwise mutual information profiles of fine distributions. *arXiv preprint arXiv:2310.10240*, 2023.

[57] Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. *Advances in Neural Information Processing Systems*, 32, 2019.

[58] Desi R Ivanova, Adam Foster, Steven Kleinegesse, Michael U Gutmann, and Thomas Rainforth. Implicit deep adaptive design: Policy-based experimental design without likelihoods. *Advances in Neural Information Processing Systems*, 34:25785–25798, 2021.

[59] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical science*, pages 273–304, 1995.

[60] Simon Valentin, Steven Kleinegesse, Neil R Bramley, Peggy Seriès, Michael U Gutmann, and Christopher G Lucas. Designing optimal behavioral experiments using machine learning. *Elife*, 13:e86224, 2024.

[61] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE transactions on information theory*, 58(5):3250–3265, 2012.

[62] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[63] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2019.

[64] Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*, 2019.

[65] Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel Yamins, and Noah Goodman. On mutual information in contrastive learning for visual representations. *arXiv preprint arXiv:2005.13149*, 2020.

[66] Amitis Shidani, R Devon Hjelm, Jason Ramapuram, Russell Webb, Eeshan Gunesh Dhekane, and Dan Busbridge. Poly-view contrastive learning. In *The Twelfth International Conference on Learning Representations*.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[68] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[69] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[70] Gokul Gowri, Xiao-Kang Lun, Allon M Klein, and Peng Yin. Approximating mutual information of high-dimensional variables using learned representations. *arXiv preprint arXiv:2409.02732*, 2024.

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

[72] Yanzhi Chen, Yingzhen Li, Adrian Weller, et al. Scalable infomin learning. *Advances in Neural Information Processing Systems*, 35:2226–2239, 2022.

[73] Dor Tsur, Ziv Goldfeld, and Kristjan Greenewald. Max-sliced mutual information. *Advances in Neural Information Processing Systems*, 36, 2024.

[74] Ivan Butakov, Alexander Tolmachev, Sofia Malanchuk, Anna Neopryatnaya, Alexey Frolov, and Kirill Andreev. Information bottleneck analysis of deep neural networks via lossy compression. In *The Twelfth International Conference on Learning Representations*.

[75] Kyungeun Lee and Wonjong Rhee. A benchmark suite for evaluating neural mutual information estimators on unstructured datasets. *arXiv preprint arXiv:2410.10924*, 2024.

A THEORETICAL PROOFS

In this section, we are going to prove Proposition 1 and Proposition 2. Before delving into that, we first revisit the proposed MIME estimator. Specifically, MIME proposes to estimate the mutual information by learning a multinomial classifier h with a set of delicate reference distributions. To train the classifier h, we approximate the learning objective by using N samples

$$\hat{h} = \underset{h}{\operatorname{argmax}}\, \mathcal{L}_N(h) := \frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{h_c(x_i,y_i)}}{\sum_{k=1}^{K} e^{h_k(x_i,y_i)}}, \quad x_i, y_i \sim p(c)p_c(x,y),$$

where we use a four-ways classifier (i.e., $K = 4$) with $p_1(x,y) = p(x,y)$ and $p_4(x,y) = p(x)p(y)$. After training, we approximate the true mutual information $I(X;Y)$ using

$$\hat{I}^N_{\hat{h}}(X;Y) := \frac{1}{N}\sum_{i=1}^{N} \hat{h}_1(x_i,y_i) - \hat{h}_4(x_i,y_i),$$

For brevity, we denote $\hat{r}(x,y) := \hat{h}_1(x,y) - \hat{h}_4(x,y)$ as the estimated ratio with the learned classifier $\hat{h}$ and $r^*(x,y) := \log \frac{p(x,y)}{p(x)p(y)}$ as the ground truth ratio. Now, we are ready to prove the consistency of the proposed MIME estimator.

A.1 PROOF OF PROPOSITION 1

Proposition 1 (Consistency of reference-based MI estimate)**.** *Assuming that the classifier $h_c : X \times Y \to \mathbb{R}$ is uniformly bounded. For every $\varepsilon > 0$, there exists $N(\varepsilon) \in \mathbb{N}$, such that*

$$\left| \hat{I}^N_{\hat{h}}(X;Y) - I(X;Y) \right| < \varepsilon, \forall N \geq N(\varepsilon), a.s..$$

Proof. Firstly, we rewrite the LHS according to our definition of $\hat{I}^N_{\hat{h}}(X;Y)$:

$$\left| \hat{I}^N_{\hat{h}}(X;Y) - I(X;Y) \right| = \left| \frac{1}{N}\sum_{i=1}^{N} \hat{r}(x_i,y_i) - I(X;Y) \right|.$$

Then, by applying the triangle inequality, we have

$$\left| \frac{1}{N}\sum_{i=1}^{N} \hat{r}(x_i,y_i) - I(X;Y) \right| \leq \left| \frac{1}{N}\sum_{i=1}^{N} \hat{r}(x_i,y_i) - \frac{1}{N}\sum_{i=1}^{N} r^*(x_i,y_i) \right| + \left| \frac{1}{N}\sum_{i=1}^{N} r^*(x_i,y_i) - I(X;Y) \right|.$$

Obviously, the second term in the RHS can be bounded by a sequence $\varepsilon_N \xrightarrow{a.s.} 0$ due to the normal strong law of large numbers under mild conditions. To bound the first term, we employ the result in [22, Appendix A], which proves that the multinomial ratio estimator is consistent. Specifically, let $\hat{h} = \operatorname{argmax}_h \mathcal{L}_N(h)$ and $\hat{r} = \hat{h}_1 - \hat{h}_4$. Then for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$, such that

$$|\hat{r}(x,y) - r^*(x,y)| < \varepsilon, \forall x, y, a.s..$$

Thereby, there exists a sequence $\varepsilon_i \xrightarrow{a.s.} 0$ for each $i = 1, 2, \ldots, N$ such that

$$\begin{aligned} \left| \frac{1}{N}\sum_{i=1}^{N} \hat{r}(x_i,y_i) - \frac{1}{N}\sum_{i=1}^{N} r^*(x_i,y_i) \right| &\leq \frac{1}{N}\sum_{i=1}^{N} |\hat{r}(x_i,y_i) - r^*(x,y)| \\ &\leq \frac{1}{N}\sum_{i=1}^{N} \varepsilon_i \leq \max(\varepsilon_1, \ldots, \varepsilon_N), \end{aligned}$$

where the first inequality holds due to the triangle inequality. Hence, for each $\epsilon > 0$, there exist an $N(\varepsilon) \in \mathbb{N}$, such that $\left| \hat{I}^N_{\hat{h}}(X;Y) - I(X;Y) \right| < \varepsilon$, almost surely. □

A.2 PROOF OF PROPOSITION 2

Proposition 2 (Controlled error in reference-based MI estimate)**.** *Define* $\log \hat{r}_{i,j}(x,y) \coloneqq h_i(x,y) - h_j(x,y)$ *with* h *being the classifier defined as above. Upon convergence, we have that:*

$$\Big|\log \hat{r}_{1,4}(x,y) - \log \frac{p(x,y)}{p(x)p(y)}\Big| \leq 3\sup_i \Big|\log \hat{r}_{i,i+1}(x,y) - \log \frac{p_i(x,y)}{p_{i+1}(x,y)}\Big|,$$

Proof. The key of the proof is the following identify in the learned network h:

$$h_1 - h_4 = \sum_{i=1}^{3}(h_i - h_{i+1})$$

which, by definition,

$$\log \hat{r}_{1,4}(x,y) = \sum_{i=1}^{3} \log \hat{r}_{i,i+1}(x,y)$$

Let $r_{i,j}(x,y) = \frac{p_i(x,y)}{p_j(x,y)}$ be the true density ratio between $p_i(x,y)$ and $p_j(x,y)$. We have the following important observation:

$$\log r_{1,4}(x,y) = \sum_{i=1}^{3} \log r_{i,i+1}(x,y)$$

which is due to the fact $\frac{p_1(x,y)}{p_4(x,y)} = \frac{p_1(x,y)}{p_2(x,y)} \cdot \frac{p_2(x,y)}{p_3(x,y)} \cdot \frac{p_3(x,y)}{p_4(x,y)}$. By triangular inequality

$$\Big|\sum_{i=1}^{3} \log \hat{r}_{i,i+1}(x,y) - \sum_{i=1}^{3} \log r_{i,i+1}(x,y)\Big| \leq \sum_{i=1}^{3} \Big|\log \hat{r}_{i,i+1}(x,y) - \log r_{i,i+1}(x,y)\Big|$$

which leads to

$$\Big|\log \hat{r}_{1,4}(x,y) - \log r_{1,4}(x,y)\Big| \leq 3 \sup_i \Big|\log \hat{r}_{i,i+1}(x,y) - \log r_{i,i+1}(x,y)\Big|$$

substituting $r_1(x,y) = \frac{p(x,y)}{p(x)p(y)}$ and setting $\lambda = 3$, we have

$$\Big|\log \hat{r}_{1,4}(x,y) - \log \frac{p(x,y)}{p(x)p(y)}\Big| \leq \lambda \Big|\log \hat{r}_{i,i+1}(x,y) - \log r_{i,i+1}(x,y)\Big|, \forall i$$

which completes the proof. Note that the constant λ is typically smaller than 3 in practice. □

B ADDITIONAL EXPERIMENTAL DETAILS AND RESULTS

B.1 DETAILS OF THE MODELS IN SYNTHETIC TASKS

We provide details of the Swiss Roll model, MoG 1 model and MoG 2 model in the synthetic task.

- *Swiss Roll*. In this model, data $x \in \mathbb{R}^2$, $y \in \mathbb{R}$ is generated by the following data generating process:
$$x_1 = \frac{t\cos(t)}{21}, \qquad x_2 = \frac{t\sin(t)}{21}, \qquad y = v,$$
where
$$t = \frac{3\pi}{2}(1 + 2\Phi(\epsilon_x)), \qquad v = \Phi(\epsilon_y)$$
$$\begin{bmatrix}\epsilon_x \\ \epsilon_y\end{bmatrix} \sim \mathcal{N}\Big(\begin{bmatrix}\epsilon_x \\ \epsilon_y\end{bmatrix}, \mathbf{0}, \begin{bmatrix}1 & \rho \\ \rho & 1\end{bmatrix}\Big)$$
, where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of standard normal distribution. It is easy to show that this model is still in the class of non-linear transformation of Gaussian.
- *MoG 1*. This model consists of 5 equally-weighted mixture of multivariate Gaussian where the mean for the kth component is $\mu_k = m_k\mathbf{1}$ and the dependence coefficient in the kth covariance matrix is ρ_k. Here $m_1, ...m_5 = [-0.4, -0.1, 0, 0.1, 0.4]$ and $\rho_1, ...\rho_5 = [0.5, 0.6, 0.7, 0.8, 0.9]$.
- *MoG 2*. This model consists of 5 equally-weighted mixture of multivariate Gaussian where the mean for the kth component is $\mu_k = m_k\mathbf{1}$ and the dependence coefficient in the kth covariance matrix is ρ_k. Here $m_1, ...m_5 = [-0.2, -0.1, 0, 0.3, 0.4]$ and $\rho_1, ...\rho_5 = [-0.2, -0.1, 0, 0.3, 0.4]$.

B.2 IMPLEMENTATION OF THE MAPPING f, g IN VECTOR GAUSSIAN COPULA

The detailed implementation of the bijective mappings f and g are task-dependent. Specifically,

- **Non-representation learning tasks**. For this type of tasks, we implement the two mappings f, g by a continuous flow model trained by flow matching [42]. This flow model is implemented as a 4 layer MLP with 1024 hidden units per each layer and softplus non-linearity. This MLP is trained by Adam [54] with a learning rate of 5×10^-4 and early stopping. We use this implementation for all the experiments in Section 6.1 (synthetic data) and Section 6.2 (Bayesian experiment design).
- **Representation learning tasks**. For this type of tasks, we implement f, g as element-wise functions. In such case, the model degenerates to a Gaussian copula model. Sampling in Gaussian copula can be done by first sampling $\epsilon \sim \mathcal{N}(\epsilon; 0, \Sigma)$, then computing the rank of ϵ_d in the population: $r_d = \lceil n\Phi(\epsilon_d) \rceil$. Here $\Phi(\cdot)$ is the CDF of standard normal distribution and n is the sample size of population. After that, we set x_d as the r_d-th smallest element in the population of X_d. The estimation of Σ can be done by inverting this process, which takes $\leq$ 30ms when $n = 10,000$.

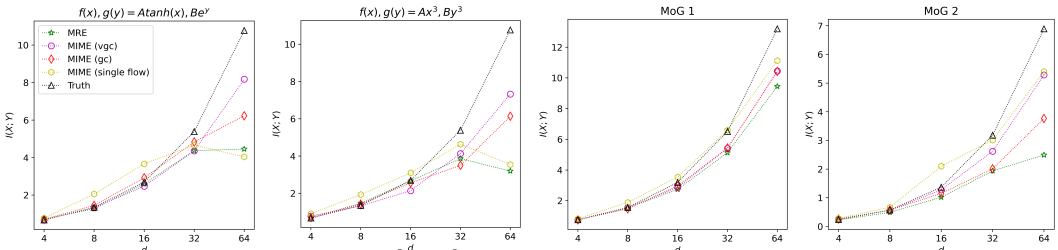
B.3 COMPUTATION TIME COMPARISONS

Table 2: Running time of a typical run of different MI estimators on the MoG task.

Method	InfoNCE	MINE	MRE	DoE	MIME
time (s)	390.24	140.71	40.28	321.45	160.21

B.4 COMPARING DIFFERENT CHOICE OF REFERENCE DISTRIBUTIONS

We conducted experiments with different choices of reference distributions $q(x, y)$. The results are shown in Figure 1. Here MIME (vgc) is our original method where $q(x, y)$ is modeled by a vector Gaussian copula and hence is marginal-preserving. MIME (gc) is the case where $q(x, y)$ is modeled by a Gaussian copula, which is fast to learn from data but is less marginal-preserving. MIME (single flow) is the case where $q(x, y)$ is modelled by a single flow model. This flow is the same as in MIME (vgc). Note that this flow is likely to be more difficult to learn than the two flows $q(x)$ and $q(y)$ in



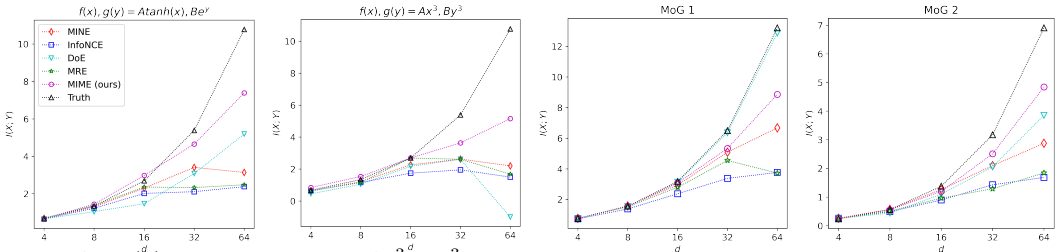
(a) $A\tanh(\overset{d}{x}), Be^y$ (b) Ax^3, By^3 (c) MoG 1 (d) MoG 2

Figure 5: Comparison of different choice of reference distribution q. The number of training data is $N = 10,000$.

VGC due to the increased dimensionality. MRE is a baseline where $q(x, y)$ is an implicit distribution that is not marginal-preserving. It is clear that MIME (vgc) achieves the best performance.

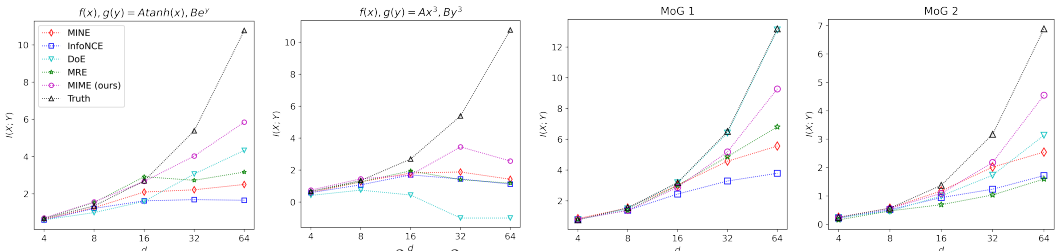
B.5 Experiments on small-data regime

we conducted experiments in small data regime with sample sizes $N = 2,500$ and $N = 1,250$ respectively. As shown in Figure 6 and Figure 7, the proposed estimator still demonstrates superior performance compared to the baselines in these small data regimes, particularly in high-dimensional settings.



(a) $A\tanh(\overset{d}{x}), Be^y$ (b) Ax^3, By^3 (c) MoG 1 (d) MoG 2

Figure 6: Comparison of different MI estimators when trained on $N = 2,500$ data.



(a) $A\tanh(\overset{d}{x}), Be^y$ (b) Ax^3, By^3 (c) MoG 1 (d) MoG 2

Figure 7: Comparison of different MI estimators when trained on $N = 1,250$ data.