Approximating mutual information of high-dimensional variables using learned representations

Gokul Gowri[1, 2, ✉], Xiao-Kang Lun[1, 2], Allon M. Klein[2, ✉, †], and Peng Yin[1, 2, †]

[1]Wyss Institute for Biologically Inspired Engineering
[2]Department of Systems Biology, Harvard University
[✉]corresponding authors: ggowri@g.harvard.edu, allon_klein@hms.harvard.edu
[†]senior authors: A.M.K. and P.Y. co-supervised this work.

Abstract

Mutual information (MI) is a general measure of statistical dependence with widespread application across the sciences. However, estimating MI between multi-dimensional variables is challenging because the number of samples necessary to converge to an accurate estimate scales unfavorably with dimensionality. In practice, existing techniques can reliably estimate MI in up to tens of dimensions, but fail in higher dimensions, where sufficient sample sizes are infeasible. Here, we explore the idea that underlying low-dimensional structure in high-dimensional data can be exploited to faithfully approximate MI in high-dimensional settings with realistic sample sizes. We develop a method that we call latent MI (LMI) approximation, which applies a nonparametric MI estimator to low-dimensional representations learned by a simple, theoretically-motivated model architecture. Using several benchmarks, we show that unlike existing techniques, LMI can approximate MI well for variables with $> 10^3$ dimensions if their dependence structure is captured by low-dimensional representations. Finally, we showcase LMI on two open problems in biology. First, we approximate MI between protein language model (pLM) representations of interacting proteins, and find that pLMs encode non-trivial information about protein-protein interactions. Second, we quantify cell fate information contained in single-cell RNA-seq (scRNA-seq) measurements of hematopoietic stem cells, and find a sharp transition during neutrophil differentiation when fate information captured by scRNA-seq increases dramatically. An implementation of LMI is available at `latentmi.readthedocs.io`.

1 Introduction

Mutual information is a universal dependence measure which has been used to describe relationships between variables in a wide variety of complex systems: developing embryos [1], artificial neural networks [2], flocks of birds [3], and more. Its widespread use can be attributed to at least two of its appealing properties: equitability and interpretability.

Many dependence measures are inequitable, meaning that they are biased toward relationships of specific forms [4]. For example, Pearson correlations quantify the strength of linear relationships, and Spearman correlations quantify the strength of monotonic relationships. Inequitability can be particularly problematic for complex systems, where relationships can be nonlinear, non-monotonic, or involve higher-order interactions between multidimensional variables [5]. Mutual information (MI) stands out as an equitable measure that can capture relationships of any form, and generalizes across continuous, discrete, and multidimensional variables [6]. And when scaled consistently, MI provides a universal currency in interpretable units – which can be understood as the number of 'bits'

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-4223

of information shared between variables [7]. MI can also be interpreted through decomposition into pointwise mutual information (pMI) [8, 9], which attributes dependence to specific pairs of values.

MI can be defined as the Kullback-Leibler divergence, D_{KL}, of a joint distribution from the product of its marginals. For absolutely continuous random variables X, Y defined over $\mathcal{X}, \mathcal{Y}$, with joint distribution P_{XY} and marginal distributions P_X, P_Y

$$I(X;Y) = D_{\text{KL}}(P_{XY}||P_X \otimes P_Y) = \int_{\mathcal{X}} \int_{\mathcal{Y}} P_{XY}(x,y) \log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} dydx \tag{1}$$

In practice, P_{XY} is often unknown, and $I(X;Y)$ must be estimated from observations $\{(x_i, y_i)\}$ that sparsely sample P_{XY}. While nonparametric MI estimators have been remarkably successful for variables with a single dimension [10, 11, 12, 13], MI estimation for high-dimensional variables remains a significant challenge. Nonparametric MI estimation suffers from the curse of dimensionality – accurate estimation requires a number of samples that scales exponentially with the dimensionality of the variables [14].

An exciting recent approach to scaling MI estimates to high dimension is the use of variational bounds on KL divergence to reduce the MI estimation problem to a gradient descent optimization problem [15, 16]. MI estimators based on variational bounds indeed empirically perform well for data with ones to tens of dimensions [12], but still suffer from the curse of dimensionality [14], and can exhibit high variance [16, 17]. To our knowledge, no techniques have been shown to reliably estimate MI in practice for variables with hundreds or thousands of dimensions – a regime relevant to many fields, including genomics, neuroscience, ecology, and machine learning [18, 19, 5, 20].

More generally, it has been shown that no technique can accurately estimate MI from finite samples without making strong assumptions about the distribution from which samples are drawn [21], resulting in a fundamental tension between the theoretical appeal of MI and the practical difficulty of its estimation. One way to resolve this tension is to develop alternative measures of statistical dependence [14, 22], which retain desirable properties of MI, but are tractable to estimate. Sliced MI, which is the average of MI estimates on random low-dimensional linear projections ("slices") of high-dimensional data, is a notable example of such an approach [14, 23]. While sliced MI is useful for many problems [24, 25], it does not retain the interpretability (in bits) of classical MI, and is inequitable, as it quantifies information that can be extracted through linear projections [14].

Here, we take a complementary approach to sliced measures. Rather than considering alternatives to classical MI, we ask if it is possible to make strong, yet reasonable, assumptions about data which enable tractable MI estimation. In this work, we consider the empirically supported assumption that complex systems have underlying low-dimensional structure [5]. The usefulness of this assumption relies on our ability to identify low-dimensional structure in data. We will explore methods for learning low-dimensional representations which are useful for MI estimation and highlight examples where the methods can still fail.

Specifically, we propose latent mutual information (LMI) approximation, which applies a non-parametric MI estimator to mutually informative compressed representations of high-dimensional variables. To learn such representations, we design a simple neural network architecture motivated by information-theoretic principles. We demonstrate, using synthetic multivariate Gaussian data, that LMI approximation can be stable for variables reaching thousands of dimensions, provided their dependence has low-dimensional structure. We then introduce an approach for resampling real data to generate benchmark datasets of two high-dimensional variables where ground truth mutual information is known. Using this approach, we evaluate the ability of LMI to capture statistical dependence in two types of real data: images and protein sequence embeddings. Finally, we apply LMI to two open problems in biology: quantifying interaction information in protein language model embeddings, and quantifying cell fate information in the gene expression of stem cells.

2 Approach

Our goal is to approximate MI from high-dimensional data using low-dimensional representations which capture dependence structure. Our specific approach is to use neural networks to map variables X, Y to low-dimensional representations Z_x, Z_y. Then, we use the well-established nonparametric MI estimator introduced in [10] to estimate $\hat{I}(Z_x; Z_y)$.

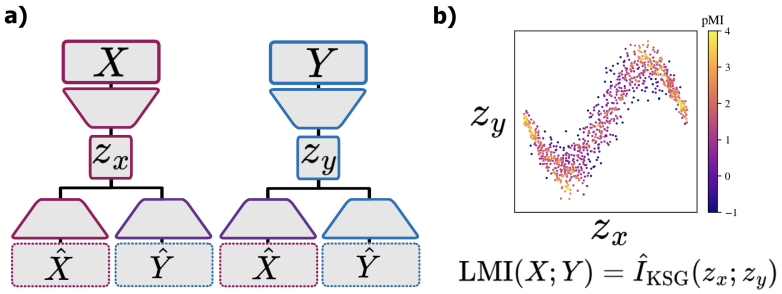


Figure 1: **Workflow of latent MI approximation a)** Embed high-dimensional data in low-dimensional space such that mutually informative structure is preserved. **b)** The KSG estimator [10] is used to estimate MI by averaging over pointwise MI (pMI) contributions.

The central challenge here is to learn Z_x, Z_y such that $\hat{I}(Z_x; Z_y) \approx I(X;Y)$. One sensible approach would be to use autoencoders [26] or other popular nonlinear dimensionality reduction techniques [27, 28] to compress each variable separately. While such an approach could yield a good approximation if compression is perfectly lossless, it can result in a poor approximation if compression is lossy. An illustrative example is two variables each with thousands of independent dimensions but a single pair of strongly dependent dimensions – two separate lossy compressions are unlikely to preserve the rare dependent components.

We can make this intuition precise using properties of entropy and MI [7]. For simplicity, let us consider an approximation using only one compressed variable, $Z_y = f(Y)$. By rewriting MI in terms of differential entropy, denoted h, we see that for absolutely continuous X, Y with finite differential entropy,

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

In the case of lossless compression, $h(X|Z_y) = h(X|Y)$, so

$$I(X;Y) - I(X;Z_y) = h(X|Z_y) - h(X|Y) = 0 \tag{3}$$

However, if compression does not perfectly preserve information, it is possible that $h(X|Z_y) \gg h(X|Y)$. Since $h(X|Y)$ is intrinsic to the data and independent of learned representations, minimizing $h(X|Z_y)$ is equivalent to minimizing $I(X;Y) - I(X;Z_y)$. This points to an approach to learn representations suitable for approximating $I(X;Y)$: regularizing a pair of autoencoders to learn compressed representations $Z_x = f(X)$ and $Z_y = g(Y)$ while minimizing $h(X|Z_y)$ and $h(Y|Z_x)$.

Because directly minimizing conditional entropies is intractable, we instead minimize a convenient proxy, which is the mean-squared error (MSE) loss of networks that predict one variable from another. The connection between conditional entropy and reconstruction loss has long been appreciated as a way to interpret autoencoders through an information-theoretic lens [29, 30]. Here, we observe that this connection can be applied to learn representations which lend themselves to MI estimation. We explicitly show that minimizing cross-prediction loss from Z_x to Y is equivalent to minimizing an upper bound on the conditional entropy $h(Y|Z_x)$ in Appendix A.1.1, Theorem 1.

Applying cross-predictive regularization to a pair of autoencoders results in a network architecture (Fig. 1) with one encoder for each variable and four decoders which reconstruct each variable from each latent code. We train the networks by minimizing the sum of the MSE reconstruction loss for each decoder. More precisely, for variables X, Y with dimensionality d_X, d_Y, we optimize encoders E_X, E_Y, and decoders $D_{XX}, D_{XY}, D_{YY}, D_{YX}$ to minimize $\mathcal{L}_{AEC} = \mathcal{L}_{AE} + \mathcal{L}_C$, where

$$\mathcal{L}_{AE} = \frac{1}{d_X}\mathbb{E}[||X - D_{XX}(E_X(X))||_2^2] + \frac{1}{d_Y}\mathbb{E}[||Y - D_{YY}(E_Y(Y))||_2^2] \tag{4}$$

$$\mathcal{L}_C = \frac{1}{d_X}\mathbb{E}[||X - D_{YX}(E_Y(Y))||_2^2] + \frac{1}{d_Y}\mathbb{E}[||Y - D_{XY}(E_X(X))||_2^2] \tag{5}$$

https://doi.org/10.52202/079017-4223

This is not the only way one could regularize autoencoders to preserve mutually informative structure. We design and empirically characterize some alternatives in Appendix A.2. We find that multiple regularization approaches can be effective, but cross-prediction comes with unique benefits. For example cross-predictive networks can be dissected to attribute high-dimensional MI estimates to specific dimensions, as demonstrated in Appendix A.2.3.

While the specific architecture of encoders and decoders could be carefully chosen for each estimation problem (e.g. convolutional layers for image data), here we use multilayer perceptrons with *a priori* determined hidden layer sizes for all problems. This is intentional: a useful MI estimator should not need extensive parameter selection. Every LMI estimate shown in this paper (excluding Appendix and Section 3.3) uses the default parameters of our library, equivalent to running `lmi.estimate(X_samples,Y_samples)`.

To ensure that optimizing based on cross-reconstruction does not introduce spurious dependence due to overfitting, we learn representations and estimate MI on different subsets of the data. That is, for N joint samples, we train the network using a subset of $N/2$ samples, then estimate MI by applying the estimator of [10] to latent representations of the remaining $N/2$ samples. A high-level overview of an MI estimate using LMI approximation is given in Algorithm 1.

We also state and prove some basic properties of LMI approximation in Appendix 1.3, Theorems 2 and 3, namely that $I(Z_x; Z_y) \leq I(X; Y)$, and that $I(Z_x; Z_y) = 0$ if $I(X; Y) = 0$.

Algorithm 1 Estimating MI using LMI Approximation

Require: N joint samples $\{(x_i, y_i)\}_{i=1}^{N}$ of random variables X, Y
Require: Encoders E_X, E_Y, decoders $D_{XX}, \ldots, D_{YY}$ parameterized by $\theta_1, \ldots, \theta_6$
randomly **split** into two subsets of $N/2$ samples, $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{est}}$
optimize $\theta_1, \ldots, \theta_6$ to minimize $\mathcal{L}_{AEC}$ on $\mathcal{D}_{\text{train}}$
encode $\mathcal{D}_{\text{est}}$ using $E_1, E_2, \theta_1, \theta_2$, yielding $\{(Z_i^x, Z_i^y)\}_{i=1}^{N/2}$
return $\hat{I}_{KSG}(\{(Z_i^x, Z_i^y)\}_{i=1}^{N/2})$

3 Empirical evaluation

Next, we empirically study the effectiveness of LMI approximation with 16 dimensional latent space (8 dimensions per variable), in comparison with three popular estimators: the nonparametric estimator from [10], referred to as KSG, and the variational bound estimators from [15, 31] referred to as MINE and InfoNCE, respectively (implementation details in Appendix A4).

3.1 Evaluating mutual information estimators on synthetic data

We first consider the problem of MI estimation between multivariate Gaussian distributions, because ground truth MI can be analytically computed, and dimensionality can be easily tuned. We consider the scalability of MI estimators with increasing dimensionality of two kinds: the ambient dimensionality of the data, denoted d, and the intrinsic dimensionality of the dependence structure, denoted k. We benchmark the performance of estimators in the regime of high ambient dimensionality and low intrinsic dimensionality. Specifically, we consider variables with $d = 10$ to $d = 5 \cdot 10^3$ ambient dimensions and $k = 1$ to $k = 9$ dimensional dependence structure.

To generate samples from two d-dimensional random variables X, Y with k-dimensional dependence structure, we sample multivariate Gaussians with a covariance matrix of a prescribed from. In particular, we enforce pairwise correlation ρ between k components of each variable, with redundant duplicate components as well as independent components constituting the remain $d - k$ dimensions. $I(X; Y)$ can then be easily computed analytically from the covariance matrix. The exact sampling procedure we use for these experiments is given in Appendix A.3.1, Algorithm 4.

Results of benchmarking MI estimators using these synthetic datasets are given in Fig. 2. For estimates from $N = 2 \cdot 10^3$ samples, we find that, as expected, the performance of existing estimators degrades with d, with near complete failure for $d > 100$ (Fig. 2a-c, 2f-h.). In contrast, applying LMI approximation results in stable estimates up to $d = 5 \cdot 10^3$ ambient dimensions (Fig. 2d, 2i). The faithfulness of LMI approximation instead degrades with increasing k. Nonetheless, LMI

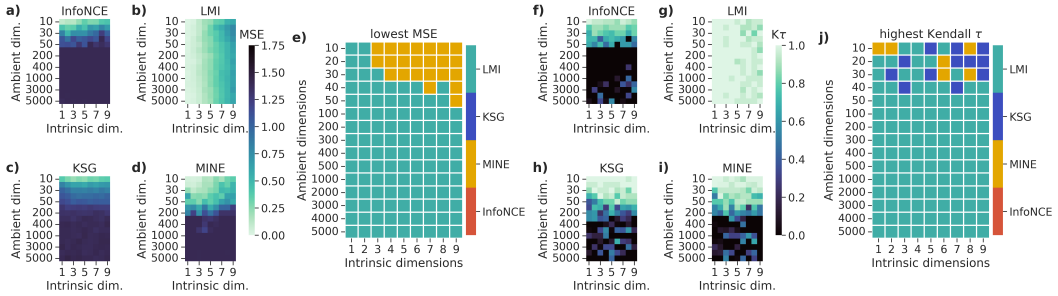


Figure 2: **MI estimator performance scaling with increasing dimensionality. a) - d)** Absolute accuracy measured by mean-squared error over 10 estimates per setting, with ground truth MI between 0 and 2 bits, and $2 \cdot 10^3$ samples per estimate (1:1 training split for neural network-based estimators). **e)** Estimator with highest absolute accuracy in each setting. Ties broken randomly. **f) - i)** Relative accuracy measured by Kendall τ rank correlation of estimates with ground truth. **j)** Estimator with highest relative accuracy in each setting. Ties broken randomly.

approximation gives more absolutely and relatively accurate MI estimates than alternatives for 83% and 87% of tested settings respectively (Fig. 2e, 2j). We also find that these benchmarking results remain qualitatively similar for multivariate Gaussian data after various MI-preserving transformations (Appendix A.3.4), similar to the benchmarking approach of [12].

3.1.1 Empirically quantifying convergence rates of MI estimators on synthetic data

The principle enabling the scalability of LMI approximation is that the number of samples it requires to converge is limited by k rather than d when $k \ll d$. We empirically demonstrate this by quantifying the convergence rates of MI estimators on the synthetic Gaussian datasets described above. We generate datasets with sample numbers in $N \in [10^2, 10^4]$, and ambient dimensionalities in $d \in [1, 50]$, each with a single correlated dimension between variables ($k = 1$), and 1 bit MI. For each estimator and each ambient dimensionality d, we empirically determine the number of samples required to achieve $|I(X, Y) - \hat{I}(X, Y)| < \epsilon$ bits, with linear interpolation between tested sample numbers.

As expected, methods that do not explicitly learn low-dimensional representations (InfoNCE, MINE, KSG) require increasing numbers of samples to estimate MI with error below $\epsilon = 0.1$ (Fig. 3a). KSG fails to estimate MI for $d \geq 13$ for any N, while MINE and InfoNCE scale slightly better, failing for $d \geq 25$ and $d \geq 37$ respectively. In contrast, the sample requirements of LMI remain qualitatively stable – no more than $4 \cdot 10^3$ samples are necessary for an accurate estimate.

While the convergence behavior of LMI is mostly unaffected by varying d, it is sensitive to varying k. When the same experiment is performed with increasing numbers of correlated dimensions at the limit where $k = d$, the convergence behavior of LMI is no longer favored over other estimators (Fig. 3c). The performance of all estimators dramatically decreases with k, such that a larger error tolerance must be chosen for informative convergence estimates. The dependence of variational bound estimator convergence on k is, to our knowledge, not explained by existing theory [16, 32]. In the intermediate case of $k = \lfloor 0.1 \cdot d \rfloor$ (Fig. 3b), we find that LMI convergence is fast with low k, but becomes slow as k grows, nonetheless remaining favorable compared to other estimators.

3.2 Evaluating mutual information estimators on resampled real-world data

While the empirical results on multivariate Gaussians are reassuring, they are not representative of performance on real data, where low intrinsic dimensionality is not known *a priori*, and distributions can be non-Gaussian. To better understand the behavior of LMI in more realistic settings, we introduce a technique for creating benchmark datasets by resampling real-world data. Briefly, we use correspondences between discrete labels and complex data (i.e. digit labels and digit images in MNIST) to transform simple discrete distributions into realistic high-dimensional distributions. A similar approach is explored in a concurrent benchmarking study [33].

Specifically, we draw samples from a bivariate Bernoulli vector, $\mathbf{L} = [L_x, L_y] \in \{0, 1\}^2$ with prescribed pairwise correlation $\text{Cor}(L_x, L_y) = \rho$, where each value corresponds to a discrete label

https://doi.org/10.52202/079017-4223

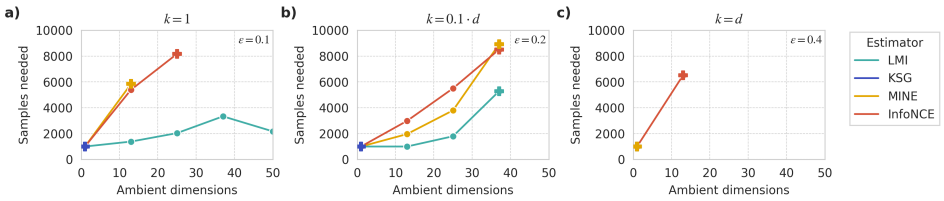


Figure 3: **Number of samples required to achieve** $|I(X,Y) - \hat{I}(X,Y)| < \epsilon$**. a)** Data with low-rank dependence structure, with $\epsilon = 0.1$. **b)** Moderate-rank dependence structure, with $\epsilon = 0.2$. **c)** Full-rank dependence structure, with $\epsilon = 0.4$. "+" marker indicates that $N > 10^4$ samples are required for accurate estimates for all larger d. Ground truth MI is 1 bit in all cases.

of a set of samples in a high-dimensional dataset (e.g. 0 and 1 correspond to images of 0s and 1s in MNIST). For each sample of $\mathbf{L}$, we replace each component with a random (without replacement) high-dimensional sample matching the label. For the example of MNIST, this transforms samples from $\mathbf{L}$ into pairs of images of 0s and 1s, represented as samples of random vectors $X, Y \in \mathbb{R}^{784}$.

Under the assumption that discrete labels can be uniquely identified by high-dimensional vectors, high-dimensional MI is identical to the discrete label MI. That is, assuming $H(L_x|X) = H(L_y|Y) = 0$ then $I(X;Y) = I(L_x;L_y)$ (shown in Appendix A.3.2, Theorem 4). And using our knowledge of ρ, $I(L_x;L_y)$ can be analytically computed.

We resample two different source datasets: (1) "binary" subset of MNIST, containing only images of 0s and 1s, with 5000 samples and 784 dimensions and (2) embeddings of a subset of protein sequences from *E. coli* and *A. thaliana* proteins, with 4402 samples and 1024 dimensions. For both source datasets, we validate the $I(L_x;L_y) \approx I(X;Y)$ approximation in Appendix A.3.3.

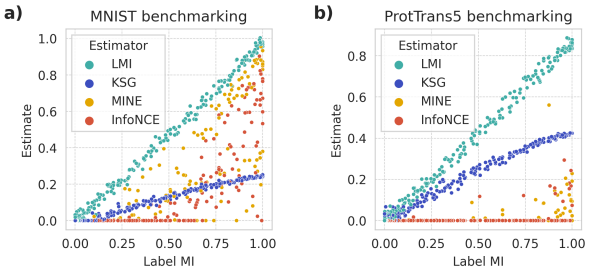
For each source dataset, we generate 200 benchmark datasets with true MI ranging from 0 to 1 bits. We estimate MI on each dataset with each estimator (Fig. 4a, 4b), and quantify absolute accuracy (MSE), relative accuracy (rank correlation with ground truth), and runtime for each estimator (Fig. 4c). For both types of source data, we find that variational bound estimators have high variance, in line with previous observations [16]. On protein embedding datasets, variational estimators nearly always fail to estimate nonzero values – resulting in a rank correlation below 0 for InfoNCE. The KSG estimator, while achieving high relative accuracy, systematically underestimates MI, resulting in low absolute accuracy. Furthermore, the amount by which it underestimates true MI is different between the two datasets – indicating inequitability. In contrast, LMI approximation yields estimates consistently close to the ground truth, with high relative and absolute accuracy.

3.3 Constructing and studying problems where LMI fails

While LMI performs well on a diverse set of benchmarks, it is not infallible. LMI can fail when (1) its representations do not capture dependence structure, or (2) when KSG fails to accurately estimate MI in latent space. Because the limitations of KSG are well-studied [34], here we focus on failure mode (1) and identify problems where LMI learns representations which result in poor MI estimates.

LMI trivially fails to learn useful representations when the dimensionality of the latent space is too small to capture the dependence structure of the variables (e.g. in Fig. 3c). This limitation can be partially overcome by evaluating LMI with a latent space large enough to capture dependence structure, however *a priori* knowledge of the appropriate embedding dimension is rarely possible. One heuristic approach (Appendix A.5.2) is to make estimates with several latent space sizes, and choose the size which maximizes the estimate.

LMI accuracy can also degrade when MSE cross-prediction loss fails to constrain latent representations, reducing the LMI model to a pair of independent autoencoders. This can happen when certain symmetries are present in the data, such as variables X, Y for which $\mathbb{E}[X|Y] = \mathbb{E}[X]$ and $I(X;Y) \geq 0$, two examples of which are shown in Fig. 5a. For such variables, the MSE-minimizing predictor of X becomes independent of Y, and cross-prediction loss no longer has a useful regularization effect. In these cases, if independent autoencoders are insufficient to capture dependence structure, LMI will fail to accurately estimate MI.



c)

	LMI	KSG	MINE	InfoNCE
MSE (MNIST)	**0.0**	0.19	0.11	0.14
Rank corr. (MNIST)	**0.95**	0.92	0.61	0.59
Runtime (MNIST)	**7.9**	39.7	12.18	12.81
MSE (PT5)	**0.01**	0.1	0.32	0.33
Rank corr. (PT5)	**0.94**	0.92	0.26	-0.06
Runtime (PT5)	6.49	37.72	**5.6**	11.09

Figure 4: **Performance of MI estimators on resampled real datasets. a)** Estimates on resampled pairs of MNIST digits, with $5 \cdot 10^3$ samples and 784 dimensions. **b)** Estimates on resampled pairs of ProtTrans5 sequence embeddings, with $4.4 \cdot 10^3$ samples and 1024 dimensions. **c)** Statistics of estimator accuracy and runtime (in seconds), for each dataset type.

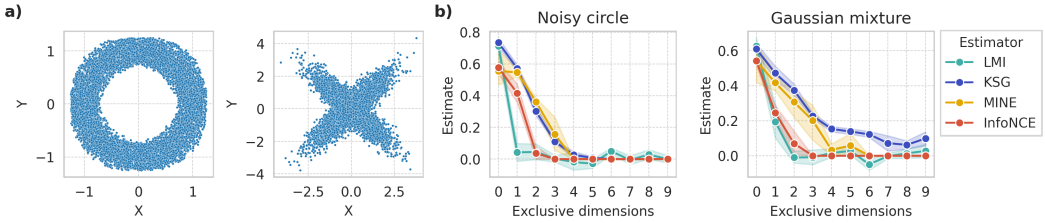


Figure 5: **Constructing problems where LMI fails. a)** Examples of symmetric distributions where $\mathbb{E}[X|Y] = \mathbb{E}[X]$. **b)** MI estimates from 10^3 samples of symmetric variables as exclusive (independent normal) dimensions are added. Ideal estimators are invariant. LMI latent space is 1D per variable.

We construct a benchmark to illustrate this limitation and understand if other estimators suffer from the same limitation in practice. In brief, we generate samples from variables with a single pair of symmetric dimensions and increasing numbers of independently normally distributed dimensions. The exact sampling procedure is given in Appendix A.3.5. An ideal estimator should not vary with the number of independent dimensions as true MI is invariant. In the case with no independent dimensions, we expect LMI to be accurate up to the limitations of KSG because independent autoencoders are sufficient to learn mutually informative representations. As the number of independent dimensions increases, LMI estimates should quickly degrade.

As anticipated, LMI estimates implemented with an MSE cross-prediction regularization decay quickly (Fig. 5b). However, this behavior is not unique to LMI: all studied estimators decay to near 0 with fewer than 10 dimensions. In this dimensionality, MINE and InfoNCE typically perform well [12], so their failure is not due merely to the dimensionality of the problem but also due to the nature of the distributions. The estimators do not agree even in the 1D case without independent dimensions, suggesting that symmetric distributions may be generally problematic for MI estimation, similar to long-tailed distributions [12].

Because this failure of LMI specifically arises from an artifact of the MSE loss, one might conclude avoiding MSE cross-prediction regularization may be sufficient to improve LMI estimation. We explore this possibility in Appendix A.2.4.

4 Applications

4.1 Quantifying interaction information in protein language model embeddings

Pretrained protein language models (pLMs) have recently seen widespread use, largely due to their convenient representations of protein sequences (vectors in $\mathbb{R}^N$, typically with $N \approx 10^3$) which can be used for transfer learning on downstream tasks [35, 36, 37]. While it is known that pLM sequence embeddings contain significant information about protein structure [38], it is not clear how well existing pLMs encode functional information. Recent work has shown that pLMs fail to capture

https://doi.org/10.52202/079017-4223

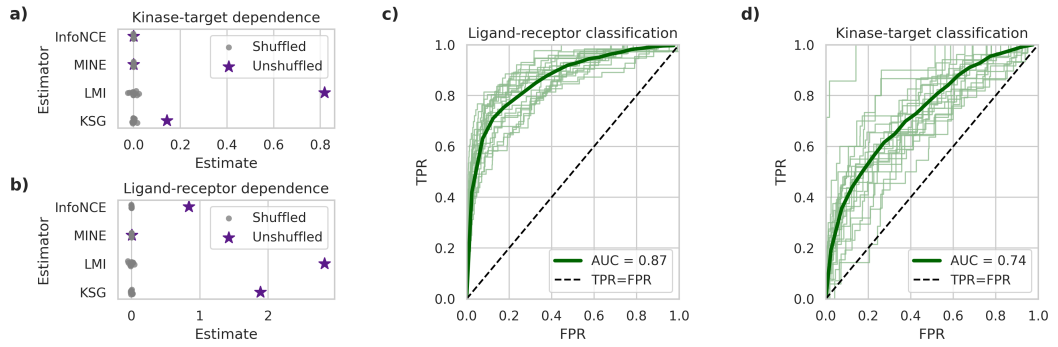


Figure 6: **Quantifying dependence between participants of protein interactions. a) - b)** MI estimates between interaction partners, compared to randomly permuted data. **c) - d)** ROC curves of density ratio classifier distinguishing annotated interacting pairs from unannotated "negative" samples, for all pairs of 170 held-out proteins. Averages over 20 random hold-out splits.

some important functional properties, such as thermostability [39]. It remains unclear the extent to which pLM embeddings contain information about protein-protein interactions, which are essential to protein function. Here, we use an information-theoretic approach to quantify interaction information contained in 1024-dimensional sequence embeddings from the ProtTrans5 model [36].

We study two types of protein-protein interactions: kinase-target and ligand-receptor interactions. For both, the OmniPath database [40] contains lists of thousands of annotated pairs of interacting proteins. We consider each annotated pair to be a sample from a joint "interaction" distribution over pairs of sequence embeddings. For example, for kinase-target interactions, we consider kinase and target sequence embeddings as random variables $K \in \mathbb{R}^{1024}, T \in \mathbb{R}^{1024}$, with joint distribution $P_{KT}(k,t)$. Then, using joint samples $\{(k_1,t_1),\ldots,(k_N,t_N)\}$ we estimate mutual information between interaction partners, $I(K;T)$. We analogously estimate $I(L;R)$ for ligand-receptor interactions.

If pLM embeddings capture interaction information, MI between interaction partners should be significantly above 0 bits. Applying LMI approximation, we estimate $I(L;R) \approx 2.8$ bits and $I(K;T) \approx 0.8$ bits. To test the significance of these values, we estimated MI from shuffled data, and found $I_{\text{shuff}}(K;T) \approx I_{\text{shuff}}(L;R) \approx 0$ (both mean and standard deviation < 0.05), across 20 random shuffles. These results indicate that pLM embeddings contain information about both types of interactions, and contain more information about ligand-receptor interactions than kinase-target interactions. In contrast, existing estimators yield far lower estimates, with MINE estimates indicating independence for both types of interactions (Fig. 6a, 6b). To validate the LMI estimates of dependence, we next operationally verify the presence of interaction information.

If protein-protein interactions can be predicted for a set of held-out proteins based on sequence embeddings, then sequence embeddings must contain interaction information. To see if this is the case, we extend LMI to predict protein interactions from sequence embeddings. For ligand-receptor prediction, our goal is to predict whether a held-out pair of sequence embeddings (l,r) is an annotated ligand-receptor pair. One way to do this is estimating the log density ratio $\log \frac{P_{LR}(l,r)}{P_L(l)\cdot P_R(r)}$, and setting a threshold above which sequence pairs are predicted to be annotated interactions.

We make a simple modification to the KSG estimator to yield the desired density ratio estimates (given in Algorithm 2), and use these estimates (with latent approximation) to predict interaction annotations. For 20 different random splits of 170 held out proteins, we use density ratio estimates to classify all $2.89 \cdot 10^4$ pairs of held out proteins as interacting or non-interacting. The receiver operating characteristic (ROC) curves for predictions of both interaction types are shown in Fig. 6c, 6d, with mean AUC-ROC scores of 0.87 and 0.74 for ligand-receptor and kinase-target interactions respectively. These results demonstrate that protein interactions can be predicted better than random chance using ProtTrans5 embeddings, suggesting that pLM embeddings capture information about kinase-target and ligand-receptor interactions. And in line with the LMI estimates, ligand-receptor interactions are better predicted than kinase-target interactions.

Algorithm 2 k-nearest neighbor log density ratio estimator

Require: joint samples $\{(x_i, y_i)\}_{i=1}^N$
Require: query point (q_x, q_y)
  let $(r_x, r_y) \leftarrow$ k-th nearest neighbor sample of (q_x, q_y) in joint space (default $k = 3$)
  \\ compute Chebyshev distance
  let $d \leftarrow ||(q_x, q_y) - (r_x, r_y)||_\infty$
  let $n_x \leftarrow 0, n_y \leftarrow 0$
  \\ count neighbors within d in marginal spaces
  for each (x_i, y_i) **do**
    if $||q_x - x_i||_\infty < d$ **then**
      n_x += 1
    end if
    if $||q_y - y_i||_\infty < d$ **then**
      n_y += 1
    end if
  end for
  \\ return estimate of $\log \frac{p(q_x, q_y)}{p(q_x)p(q_y)}$
  return $\psi(k) + \psi(N) - \psi(n_x) - \psi(n_y)$, where ψ is Digamma function

4.2 Identifying cell fate information in hematopoietic stem cells

Single cell RNA sequencing (scRNA-seq) measures the expression of $g \approx 10^4$ genes in single cells, which can be thought of samples of a gene expression state variable $X \in \mathbb{R}^g$. These samples can be used to infer a probability distribution over gene expression states, P_X. To study the dynamics of gene expression, one approach is to make measurements at multiple timepoints $t_1, ...t_N$, which can be thought of as samples of random variables $X_i \in \mathbb{R}^g$. Lineage tracing is a technique where clonally related cells can be labelled with barcodes, allowing sampled cells from different timepoints to be matched with their "twins" from another. When combined with scRNA-seq, lineage tracing can be thought to provide samples from the joint distribution $P_{X_1,...,X_N}$ [41].

One fundamental question about cellular dynamics is whether the time evolution of gene expression state is dependent entirely on the current gene expression state. That is, if the "fate" of a cell can be formally modeled as a Markov chain $X_i \to X_{i+1}$. In some cases, cell behavior may be a function of hidden variables resulting in non-Markovian dynamics. Using the data processing inequality (DPI) [7], we know that if gene expression dynamics are Markovian, $I(X_i; X_{i+1}) \geq I(X_i; X_{i+2})$, and if the DPI does not hold, then gene expression states are non-Markovian. This can, in principle, be tested using samples from the joint distribution $P_{X_1,..,X_N}$. Due to the difficulty of high-dimensional MI estimation, previous work has used heuristic alternatives to indirectly test the Markovian assumption for lineage-traced scRNA-seq (LT-seq) data [42]. Here, we will use LMI to explicitly estimate high-dimensional MI from LT-seq data, and test the Markovian assumption for gene expression states.

We study a previously published LT-seq data set of *in vitro* differentiating mouse hematopoietic stem cells [42]. The dataset includes sister cells which are separated at day 2 of the experiment and allowed to differentiate in separate wells until day 6, with cell states sampled on both days 2 and 6. Under the Markovian assumption, this can be modeled as $X_2 \to X_6$, $X_2 \to X_{6'}$, where $X_{6'}$ is the state of the cells on day 6 in the second well. By the DPI we should have $I(X_2; X_6) \geq I(X_6; X_{6'})$. Using LMI, we estimate that $I(X_2; X_6) \approx 0.31 \pm 0.02$ bits and $I(X_6; X_{6'}) \approx 0.98 \pm 0.01$ bits (mean $\pm$ SEM), over 20 random pairings of clonally related cells. This indicates that gene expression states are non-Markovian, in line with prior findings [42, 43].

Cell fate information manifests in transcriptomes sometime between days 2 and 6. By decomposing LMI estimates into pointwise contributions (Appendix A.4.3, Algorithm 5), we can determine precisely when this hidden information emerges. Generally, we see that pMI increases along differentiation trajectories (Fig. 7b). Along the neutrophil trajectory, we quantatively compare cell fate information with neutrophil pseudotime computed by graph smoothing [42], and find that pMI begins to rapidly increase around pseudotime value of $35 \cdot 10^3$, which is roughly aligned with the transition from granulocyte-myeloid progenitor to promyelocyte, as defined in [42].

 https://doi.org/10.52202/079017-4223

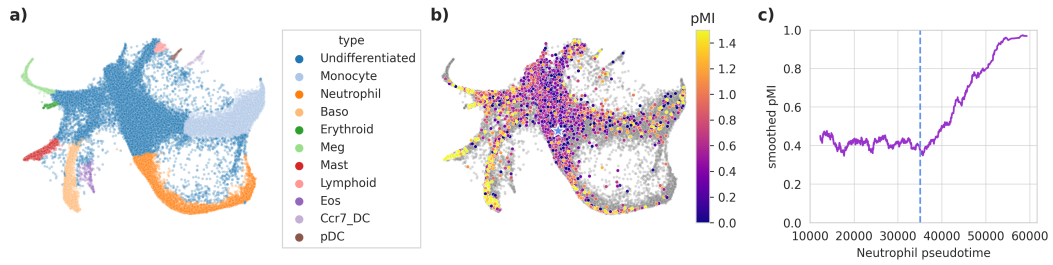


Figure 7: **Quantifying cell fate information in single cell transcriptomes. a)** 2D SPRING embedding [44] of lineage-traced single cell RNA-sequencing data from [42]. **b)** Pointwise decomposition of MI between sister cells across timepoints, as estimated using LMI. Hue applied to early timepoint cells, late timepoint and unbarcoded cells in grey. Star indicates neutrophil pseudotime value $35 \cdot 10^3$. **c)** Smoothed pMI (rolling average over 100 cells) across neutrophil differentiation trajectory. Vertical line denotes neutrophil pseudotime value $35 \cdot 10^3$.

5 Discussion

In this paper, we tested the hypothesis that low-dimensional structure can enable scalable estimation of MI from high-dimensional data. We introduced LMI approximation, which applies a nonparametric MI estimator to low-dimensional representations learned by neural networks. We quantified the effectiveness of LMI approximation, using multiple approaches spanning over 3000 benchmark datasets. Our results suggest that, unlike existing techniques, LMI approximation is generally effective for high-dimensional data with low-dimensional structure, even if the number of available samples remains relatively low – a regime where many real datasets reside [5].

We used LMI to study two open problems in biology. We show one example where LMI enables the use of information-theoretic ideas to study the dynamics of gene expression. In the original study [44], the authors had indeed wished to estimate MI but were unable to do so and resorted to heuristic approaches. LMI may similarly help identify dependence in cellular dynamics in other systems [43, 45, 46]. In a different subfield of biology, we showed an example of using LMI to quantify functional information learned by pLMs. Our results suggest that nontrivial protein-protein interaction information is learned by ProtTrans5, motivating the development of interaction prediction tools based on pLMs. As the number of large pLMs grows [35, 37, 47, 36], information-theoretic approaches using LMI could help benchmark models.

Limitations The most prominent limitation of LMI follows directly from its motivating assumption: it will fail to quantify dependence not captured by the learned representations. As a result, it is easy to design synthetic datasets on which LMI will fail (see Fig. 3c). However, it is likely that many real datasets (beyond those explored in this paper) will be amenable to LMI approximation, as there is strong evidence that complex systems generally have low intrinsic dimensionality [5]. Our implementation of LMI approximation also inherits some limitations of the KSG estimator, notably that it fails for strongly dependent (near deterministically related) variables [32, 34]. To overcome this, it may be possible to apply previously developed corrections [34]. Finally, despite reassuring empirical results, few theoretical properties of LMI have been derived. This is an important line of future work, which could help precisely identify settings where LMI approximation is effective.

Broader impacts MI estimators have been used to quantify moral and legal fairness [25, 48, 49]. LMI approximation is not universally faithful, and more generally no MI estimator can be universally accurate [32]. MI estimates must be interpreted with great care when applied to human lives. The experiments (and pilot iterations) in this paper were performed on a single NVIDIA RTX 3090, and resulted in estimated 45.36 kg CO_2eq. Estimates made using [50].

Code availability The code necessary to reproduce all results from this paper are available at `https://github.com/ggdna/latent-mutual-information`. The `lmi` Python package can be found at `https://github.com/ggdna/latentmi`, and its documentation is hosted at `https://latentmi.readthedocs.io`.

Acknowledgments and Disclosure of Funding

We thank Caroline Holmes, Ninning Liu, Sean McGeary, and Pippa Richter for thoughtful discussions. This work is supported by funding from NIH Pioneer Award DP1GM133052, R01HG012926 to P.Y., and Molecular Robotics Initiative at the Wyss Institute.

Author contributions G.G. conceived of, designed the study, developed the software, analyzed the data, and wrote the paper. X.L. contributed to study design. A.M.K. contributed to the design of the study, analysis of the data, wrote the paper, and supervised the study. P.Y. supervised the study. All authors reviewed, edited, and approve the paper.

References

[1] Julien O Dubuis, Gasper Tkacik, Eric F Wieschaus, Thomas Gregor, and William Bialek. Positional information, in bits. *Proc. Natl. Acad. Sci. U. S. A.*, 110(41):16301–16308, October 2013.

[2] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv [cs.LG]*, March 2017.

[3] Matthijs Meijers, Sosuke Ito, and Pieter Rein Ten Wolde. Behavior of information flow near criticality. *Phys Rev E*, 103(1):L010102, January 2021.

[4] David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, December 2011.

[5] Vincent Thibeault, Antoine Allard, and Patrick Desrosiers. The low-rank hypothesis of complex systems. *Nat. Phys.*, pages 1–9, January 2024.

[6] Justin B Kinney and Gurinder S Atwal. Equitability, mutual information, and the maximal information coefficient. *Proc. Natl. Acad. Sci. U. S. A.*, 111(9):3354–3359, March 2014.

[7] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. Wiley & Sons, Incorporated, John, 2006.

[8] Yao-Hung Hubert Tsai, Han Zhao, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Neural methods for point-wise dependency estimation. *arXiv [cs.LG]*, June 2020.

[9] Xianghao Kong, Ollie Liu, Han Li, Dani Yogatama, and Greg Ver Steeg. Interpretable diffusion via information decomposition. *arXiv [cs.LG]*, October 2023.

[10] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 69(6 Pt 2):066138, June 2004.

[11] Caroline M Holmes and Ilya Nemenman. Estimation of mutual information for real-valued data with error bars and controlled bias. *Phys Rev E*, 100(2-1):022404, August 2019.

[12] Paweł Czyż, Frederic Grabowski, Julia E Vogt, Niko Beerenwinkel, and Alexander Marx. Beyond normal: On the evaluation of mutual information estimators. *arXiv [stat.ML]*, June 2023.

[13] Thalia E Chan, Michael P H Stumpf, and Ann C Babtie. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell Syst*, 5(3):251–267.e3, September 2017.

[14] Ziv Goldfeld and Kristjan Greenewald. Sliced mutual information: A scalable measure of statistical dependence. *arXiv [cs.IT]*, October 2021.

[15] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. MINE: Mutual information neural estimation. *arXiv [cs.LG]*, January 2018.

 https://doi.org/10.52202/079017-4223

[16] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv [cs.LG]*, May 2019.

[17] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. *arXiv [cs.LG]*, October 2019.

[18] Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, May 2015.

[19] Surya Ganguli and Haim Sompolinsky. Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis. *Annu. Rev. Neurosci.*, 35:485–508, April 2012.

[20] Ziv Goldfeld, Ewout van den Berg, Kristjan Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. Estimating information flow in deep neural networks. *arXiv [cs.LG]*, October 2018.

[21] 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, June 2020.

[22] Danqi Liao, Chen Liu, Benjamin W Christensen, Alexander Tong, Guillaume Huguet, Guy Wolf, Maximilian Nickel, Ian Adelstein, and Smita Krishnaswamy. Assessing neural network representations during training using noise-resilient diffusion spectral entropy. In *2024 58th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, March 2024.

[23] Ziv Goldfeld, Kristjan Greenewald, Theshani Nuradha, and Galen Reeves. k-sliced mutual information: A quantitative study of scalability with dimension. *arXiv [cs.IT]*, June 2022.

[24] Dor Tsur, Ziv Goldfeld, and Kristjan Greenewald. Max-sliced mutual information. *arXiv [cs.LG]*, September 2023.

[25] Yanzhi Chen, Wei-Der Sun, Yingzhen Li, and Adrian Weller. Scalable infomin learning. *Adv. Neural Inf. Process. Syst.*, abs/2302.10701, February 2023.

[26] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[27] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W H Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.*, December 2018.

[28] Kevin R Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, Natalia B Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions in high-dimensional biological data. *Nat. Biotechnol.*, 37(12):1482–1492, December 2019.

[29] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 1096–1103, New York, NY, USA, July 2008. Association for Computing Machinery.

[30] 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 [stat.ML]*, August 2018.

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv [cs.LG]*, July 2018.

[32] David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. *arXiv [cs.IT]*, November 2018.

[33] Kyungeun Lee and Wonjong Rhee. A benchmark suite for evaluating neural mutual information estimators on unstructured datasets. *arXiv [stat.ML]*, October 2024.

[34] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. *arXiv [cs.IT]*, November 2014.

[35] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U. S. A.*, 118(15), April 2021.

[36] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *arXiv [cs.LG]*, July 2020.

[37] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, 16(12):1315–1322, December 2019.

[38] Zhidian Zhang, Hannah K Wayment-Steele, Garyk Brixi, Haobo Wang, Matteo Dal Peraro, Dorothee Kern, and Sergey Ovchinnikov. Protein language models learn evolutionary statistics of interacting sequence motifs. *bioRxiv*, page 2024.01.30.577970, January 2024.

[39] Francesca-Zhoufan Li, Ava P Amini, Yisong Yue, Kevin K Yang, and Alex X Lu. Feature reuse and scaling: Understanding transfer learning with protein language models. *bioRxiv*, page 2024.02.05.578959, February 2024.

[40] Dénes Türei, Alberto Valdeolivas, Lejla Gul, Nicolàs Palacio-Escat, Michal Klein, Olga Ivanova, Márton Ölbei, Attila Gábor, Fabian Theis, Dezső Módos, Tamás Korcsmáros, and Julio Saez-Rodriguez. Integrated intra- and intercellular signaling knowledge for multicellular omics analysis. *Mol. Syst. Biol.*, 17(3):e9923, March 2021.

[41] Shou-Wen Wang, Michael J Herriges, Kilian Hurley, Darrell N Kotton, and Allon M Klein. CoSpar identifies early cell fate biases from single-cell transcriptomic and lineage information. *Nat. Biotechnol.*, February 2022.

[42] Caleb Weinreb, Alejo Rodriguez-Fraticelli, Fernando D Camargo, and Allon M Klein. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367(6479), February 2020.

[43] Kunal Jindal, Mohd Tayyab Adil, Naoto Yamaguchi, Xue Yang, Helen C Wang, Kenji Kamimoto, Guillermo C Rivera-Gonzalez, and Samantha A Morris. Single-cell lineage capture across genomic modalities with CellTag-multi reveals fate-specific gene regulatory changes. *Nat. Biotechnol.*, September 2023.

[44] Caleb Weinreb, Samuel Wolock, and Allon M Klein. SPRING: a kinetic interface for visualizing high dimensional single-cell expression data. *Bioinformatics*, 34(7):1246–1248, April 2018.

[45] Duncan M Chadly, Kirsten L Frieda, Chen Gui, Leslie Klock, Martin Tran, Margaret Y Sui, Yodai Takei, Remco Bouckaert, Carlos Lois, Long Cai, and Michael B Elowitz. Reconstructing cell histories in space with image-readable base editor recording. *bioRxiv*, page 2024.01.03.573434, January 2024.

[46] Daniel E Wagner and Allon M Klein. Lineage tracing meets single-cell omics: opportunities and challenges. *Nat. Rev. Genet.*, 21(7):410–427, July 2020.

[47] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan Dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, March 2023.

[48] Jaewoong Cho, Gyeongjo Hwang, and Changho Suh. A fair classifier using mutual information. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2521–2526. IEEE, June 2020.

 https://doi.org/10.52202/079017-4223

[49] Jian Kang, Tiankai Xie, Xintao Wu, Ross Maciejewski, and Hanghang Tong. InfoFair: Information-theoretic intersectional fairness. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1455–1464. IEEE, December 2022.

[50] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv [cs.CY]*, October 2019.

[51] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 2010. PMLR.

[52] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. *arXiv [cs.LG]*, December 2019.

[53] Lukas Heumos, Anna C Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D Lücken, Daniel C Strobl, Juan Henao, Fabiola Curion, Single-cell Best Practices Consortium, Herbert B Schiller, and Fabian J Theis. Best practices for single-cell analysis across modalities. *Nat. Rev. Genet.*, pages 1–23, March 2023.

A Appendix / supplemental material

In this appendix, we first elaborate on the theory and implementation details of the LMI approximation. Then, we will discuss some alternative approaches one could use to implement the LMI approximation. Then, in Appendix A.3., we describe the details, assumptions, and motivations of our empirical evaluation benchmarks. In Appendix A.4., we provide details relevant to reproducibility – specifically, our implementations of existing estimators and data preprocessing methods (all of which are also included in our code supplement). Finally, we discuss the problem of choosing an appropriate latent space size for LMI models.

A.1 Theory and implementation of LMI

A.1.1 Theoretically motivating the cross-predictive representation learning architecture

The core theoretical underpinning of the network architecture used in the LMI approximation is that cross-predictive mean-squared loss is a proxy of conditional entropy. Here, we explicitly show this.

Theorem 1. *Let $X = [X_1, \ldots, X_d]$ and $Z = [Z_1, \ldots, Z_k]$ be absolutely continuous random vectors in $\mathbb{R}^d$ and $\mathbb{R}^k$ respectively with finite differential entropy. Let $f_\theta : \mathbb{R}^k \to \mathbb{R}^d$ be a function (a neural network parameterized by θ) to estimate $\hat{X} = f_\theta(Z)$. For any θ,*

$$h(X|Z) \leq \alpha + \frac{1}{2} \log \textit{MSE}(\hat{X}, X) \tag{6}$$

where α is a positive constant and $\textit{MSE}(\hat{X}, X) = \frac{1}{d} \sum_i \mathbb{E}[(X_i - \hat{X}_i)^2]$

Proof. From the chain rule for differential entropy, we can bound

$$h(X|Z) \leq \sum_i h(X_i|Z) \tag{7}$$

Because the maximum entropy distribution with fixed variance is Gaussian [7], we can bound

$$\sum_i h(X_i|Z) \leq \sum_i \frac{1}{2} \log(2\pi e \operatorname{Var}(X_i|Z)) \tag{8}$$

$$= \sum_i \frac{1}{2} \log(2\pi e \, \mathbb{E}\big[(X_i - \mathbb{E}[X_i|Z])^2\big]) \tag{9}$$

Because the expectation of a random variable is its best estimator, [7]

$$\sum_i \frac{1}{2} \log(2\pi e \, \mathbb{E}\big[(X_i - \mathbb{E}[X_i|Z])^2\big]) \leq \sum_i \frac{1}{2} \log(2\pi e \, \mathbb{E}\big[(X_i - \hat{X}_i)^2\big]) \tag{10}$$

So with positive constant α,

$$h(X|Z) \leq \alpha + \frac{1}{2} \log \operatorname{MSE}(\hat{X}, X) \tag{11}$$

□

In LMI, Z corresponds to the latent representation of one input variable (Y), and f corresponds to the decoder which aims to reconstruct the other variable X from the latent code Z.This result is very similar to some used in information-theoretic interpretations of autoencoders [29, 30], and can be thought of as a continuous analog of Fano's inequality [7]. The $d = k = 1$ case of this bound is given as a Corollary to Theorem 8.6.6 in [7].

 https://doi.org/10.52202/079017-4223

A.1.2 Implementation of the representation learning architecture

There are many ways one could implement the high-level network architecture suggested by Theorem 1, with different encoder and decoder architectures, and choices of hyperparameters. Here, we will describe our design choices. Our motivating philosophy was that the implementation details should not need to be tuned for specific estimation problems.

All LMI estimates presented in the main text use programatically determined parameters, requiring no user input beyond joint samples. By default, all latent representations have 16 dimensions (8 for each variable). Each encoder and decoder is a multilayer perceptron (MLP) with three hidden layers, whose sizes are determined as a function of the dimensionality of the input variable. For a variable with dimensionality d, the encoder has hidden layer sizes $L, L/2, L/4$ with $L = \max(2^{\lfloor log_2(d) \rfloor}, 1024)$. Decoders have the same structure inverted, with hidden layer sizes $L/4, L/2, L$.

All MLP activations used are Leaky ReLUs, with negative slope 0.2, except the last layers of decoders, which have no activation. Cross-decoders are trained with 50% dropout after each activation layer. All weights are initialized using Xavier uniform initialization [51], and optimized using Adam, with hyperparameters listed in Table 1. They are trained with batch size of 512, with $1:1$ train-validation splits, and a maximum of 300 epochs using early stopping procedure provided in Algorithm 3.

All models are implemented in Pytorch [52]. All experiments in this paper were done using a single NVIDIA RTX 3090.

Parameter	Value
Learning rate (α)	10^{-4}
β_1	0.9
β_2	0.999
Epsilon (ϵ)	10^{-7}

Table 1: Adam optimizer parameters used in LMI.

Algorithm 3 Early Stopping

```
procedure EarlyStopping(model, validation_losses, patience = 30)
    best_loss ← ∞
    patience_counter ← 0
    best_model ← None
    while patience_counter < patience do
        current_loss ← validation_loss(model)
        if current_loss < best_loss then
            best_loss ← current_loss
            patience_counter ← 0
            best_model ← model
        else
            patience_counter ← patience_counter + 1
        end if
    end while
    return best_loss, best_model
end procedure
```

A.1.3 Theoretical properties of LMI approximation

The error of MI estimates using LMI approximation can be broadly attributed to two sources: error due to the representation approximation (i.e. $|I(X;Y) - I(Z_x;Z_y)|$), and classical MI estimation error (i.e. $|\hat{I}_{KSG}(Z_x;Z_y) - I(Z_x;Z_y)|$). Here, we will explore the first source by deriving some basic properties of the representation approximation $I(Z_x;Z_y)$.

Theorem 2. *Let X, Y be random vectors in $\mathbb{R}^d$ and $Z_x = f_\theta(X)$, $Z_y = g_\phi(Y)$ where f, g are neural networks parameterized by θ, ϕ. For any θ, ϕ,*

$$I(Z_x; Z_y) \leq I(X; Y) \tag{12}$$

Proof. Because $Z_y = f_\theta(Y)$, we have that $X \to Y \to Z_y$ form a Markov chain. From the data processing inequality, [7]

$$I(X; Y) \geq I(X; Z_y) \tag{13}$$

Similarly, we have $Z_y \to X \to Z_x$, and

$$I(X; Y) \geq I(X; Z_y) \geq I(Z_x; Z_y) \tag{14}$$

□

Theorem 3. *Let X, Y be independent random vectors in $\mathbb{R}^d$, such that $I(X; Y) = 0$. Let $Z_x = f_\theta(X)$, $Z_y = g_\phi(Y)$ where f, g are neural networks parameterized by θ, ϕ. For any θ, ϕ,*

$$I(Z_x; Z_y) = 0 \tag{15}$$

Proof. Due to the nonnegativity of mutual information, we know

$$I(Z_x; Z_y) \geq 0 \tag{16}$$

From Theorem 2, we have

$$I(Z_x; Z_y) \leq I(X; Y) = 0 \tag{17}$$

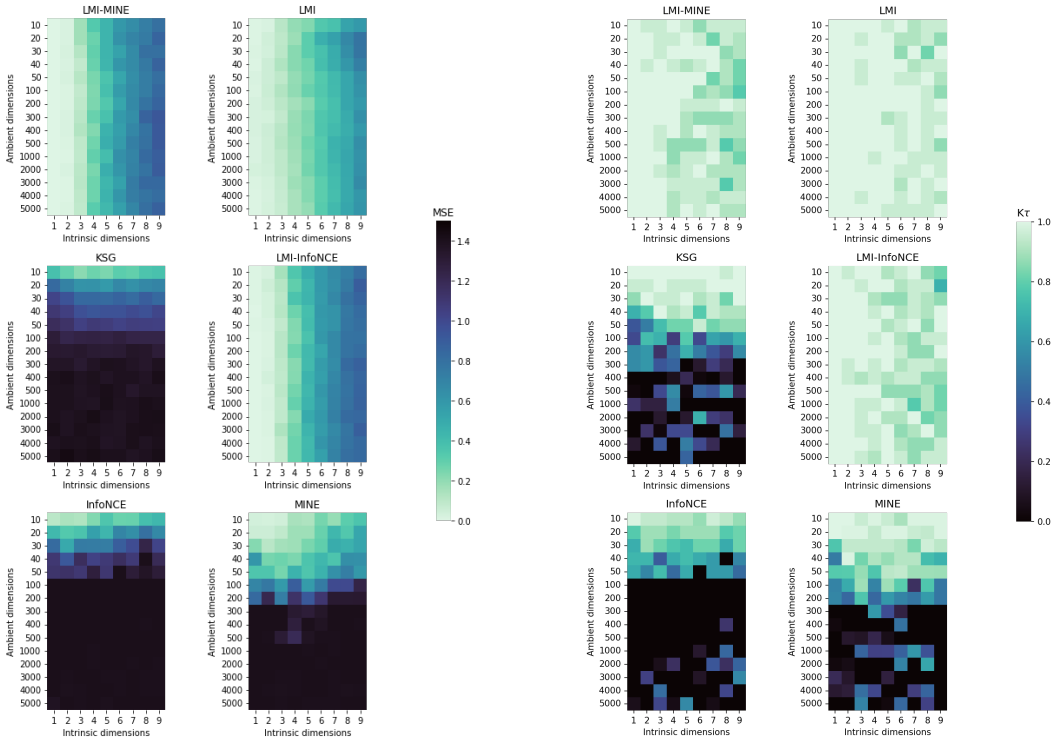
So,

$$0 \leq I(Z_x; Z_y) \leq 0 \tag{18}$$

$$I(Z_x; Z_y) = 0 \tag{19}$$

□

 https://doi.org/10.52202/079017-4223



(a) Absolute error, measured by MSE.

(b) Relative error, measured by Kendall τ.

Figure 8: Experiment from Figure 2, with alternate regularization approaches.

A.2 Alternate approaches to latent MI approximation

The broad goal of LMI, to estimate $I(X;Y)$ using $I(Z_y;Z_x)$ where Z_x, Z_y are low-dimensional representations, is quite general and could be approached in many ways beyond cross-predictive regularization. Next, we will discuss some alternative approaches, empirically explore their performance, and finally, show one unique advantage of the cross-predictive representation learning architectures.

A.2.1 Alternate methods of regularizing autoencoders for MI estimation

Another approach to learn Z_x, Z_y such that $\hat{I}(Z_x;Z_y) \approx I(X;Y)$ is to regularize autoencoders to maximize $I(Z_x;Z_y)$. This approach is sensible because the data processing inequality ensures that $I(Z_x;Z_y) \leq I(X;Y)$. So maximizing $I(Z_x;Z_y)$ is equivalent to minimizing the approximation error $I(X;Y) - I(Z_x;Z_y)$.

While maximizing directly $I(Z_x;Z_y)$ is intractable, we can build on the variational bounds explored in [15, 31]. We can add loss term $\mathcal{L}_{\text{MINE}}(Z_x;Z_y)$ or $\mathcal{L}_{\text{InfoNCE}}(Z_x;Z_y)$ to our autoencoder loss functions to regularize latent codes to preserve mutually informative structure.

We implement both of these approaches, and benchmark them using the approach described in Figure 2 and Section 3.1. We find that the estimates from these regularization approaches perform similarly, but slightly more poorly, than the cross-predictive regularization.

A.2.2 Comparing latent nonparametric and latent variational MI estimation

After learning low-dimensional representations, there are multiple estimators that could be used for latent MI approximation. This appendix evaluates two methods: the KSG nonparametric estimator or the InfoNCE variational estimator. Nonparametric nearest-neighbor estimators have several advantages in low-dimensional settings: they generally require far fewer samples for accurate estimation [12], and yield pointwise mutual information decompositions (see Algorithm 5).

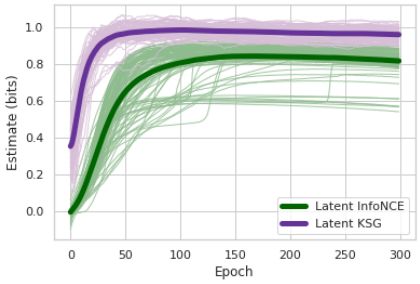


Figure 9: Convergence of multiple latent estimation approaches during training. Bold lines indicate averages over 100 trials. Ground truth is 1 bit MI.

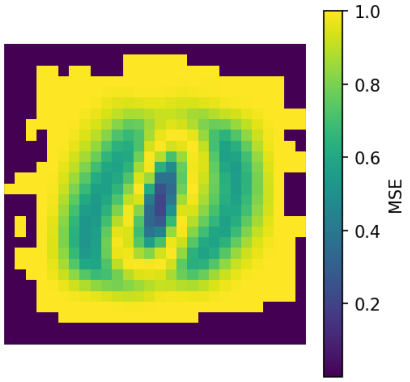


Figure 10: Pixel-wise reconstruction error of cross-decoders in paired binary MNIST dataset where $L_x = L_y$.

For completeness, we empirically compare both options here on one particular estimate. For a Gaussian dataset generated by Algorithm 4 with $d = 100$, $k = 1$, $N = 5 \cdot 10^3$ and 1 bit MI, we train autoencoders regularized by InfoNCE loss to maximize $I(Z_x; Z_y)$. After each training epoch, we measure LMI as estimated using latent KSG, and using latent InfoNCE. Both are plotted for 100 trials in Fig. 9. The latent KSG estimation converges quickly to the true value, while the latent InfoNCE estimate converges somewhat slowly to a value below the ground truth.

A.2.3 Interpreting decoders with element-wise reconstruction error

Beyond performance differences, one benefit of using cross-predictive networks to regularize latent representations is that the cross-decoders themselves are useful. For example, by inspecting the dimension-wise reconstruction error of decoders, we can attribute an MI estimate to the predictability of certain dimensions. For a network trained on the "binary" MNIST dataset with $L_x = L_y$, visualizing the dimension-wise reconstruction error of a cross-predictive decoder reveals the pixels that contain information about digit identity (Fig. 10).

Pixels with low reconstruction error are likely to be "well-explained" by the other high-dimensional variable, while pixels with high reconstruction error are poorly explained. However, this reasoning is not universally applicable. Dimensions with no variation do not contribute to MI, but have very low reconstruction error. In Fig. 10, the outermost pixels are examples of this.

A.2.4 Effect of regularization on failure for symmetric and exclusive variables

Because the failure to preserve MI specifically arises from distributions with symmetry under MSE loss, one might conclude avoiding cross-prediction loss may be sufficient to improve LMI estimation. Indeed, we demonstrate this for two models, LMI-MINE and LMI-InfoNCE, described in Appendix A.2.1, which are regularized by maximizing a variational lower bound on $I(Z_x; Z_y)$ rather than cross-prediction. In our Gaussian benchmark, we found these regularizations to be less effective than

https://doi.org/10.52202/079017-4223

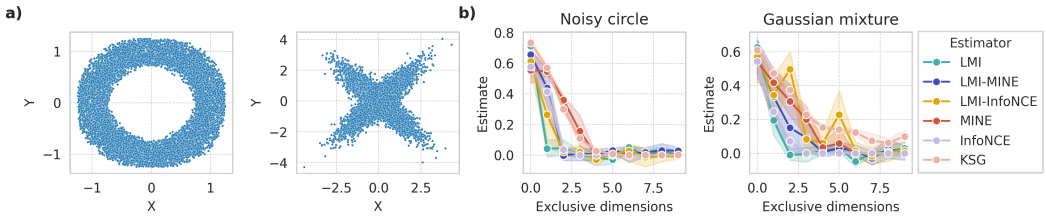


Figure 11: **a)** Examples of symmetric distributions where $\mathbb{E}[X|Y] = \mathbb{E}[X]$. **b)** MI estimates from 10^3 samples of symmetric variables as exclusive dimensions are added (increasing non-mutual information). Ideal estimators are invariant. All LMI latent spaces are 1 dimension per variable. LMI-MINE and LMI-InfoNCE denote alternately regularized LMI models.

cross-prediction: performance decayed more quickly with increasing intrinsic dimensions (Fig. 11). Here, however, these alternate models perform better than LMI, although ultimately they still fail with 8 exclusive dimensions. Many practical data sets may not suffer from the symmetries considered here, but these failure modes are nonetheless instructive in clarifying how choices made in learning latent representations may drive loss of MI. In practice, if one suspects that their data may have structure not captured by MSE cross-prediction loss (e.g. symmetries), the benchmarking results suggest that it may be advisable to use one of the alternate regularization methods. In our software library, this is simple to do, e.g. `lmi(X, Y, regularizer="models.AEMINE")`.

A.3 Details of experimental evaluation benchmarks

In this section, we will describe the details of our experimental evaluation benchmarks from Section 3. First, we will describe how we generate multivariate Gaussian datasets. Then, we will provide theoretical and empirical validation for the cluster-based benchmarking approach.

A.3.1 Generating multivariate Gaussian datasets with low-dimensional dependence structure

The algorithm used to sample multivariate Gaussians in Figure 2 is given in Algorithm 4. Briefly, we generate samples of k correlated dimensions by sampling k bivariate Gaussians. We then complete the remaining $d-k$ ambient dimensions using half "redundant" dimensions (copies of the k correlated dimensions) and half "nuisance" dimensions (independent univariate Gaussians).

Algorithm 4 Generating multivariate Gaussian datasets with low-dimensional dependence structure

Require: ambient dimensionality d
Require: dependence structure dimensionality k
Require: number of nuisance dimensions n (default $(d-k)/2$)
Require: number of samples N
Require: ground truth MI b
\\ Compute cov matrix to yield dependence b
let $\rho \leftarrow \sqrt{6 * 3.5} * \sqrt{1 - 2^{-2b/k}}$
let $\Sigma \leftarrow [[6, \rho], [\rho, 3/5]]$
\\ Sample bivariate Gaussians for k dependent dimensions
for i in $1..k$ **do**
  let $X_i, Y_i \leftarrow N$ samples from $\mathcal{N}(0, \Sigma)$
end for
\\ Duplicate random dimensions for redundant dimensions
for i in $1..(d-(k+n))$ **do**
  let $r \leftarrow$ unif$([1..k])$
  let $X_{i+k}, Y_{i+k} \leftarrow X_r, Y_r$
end for
\\ Sample univariate Gaussians for nuisance dimensions
for i in $0..(n-1)$ **do** let $X_{d-i} \leftarrow N$ samples from $\mathcal{N}(0,1)$ let $Y_{d-i} \leftarrow N$ samples from $\mathcal{N}(0,1)$
end for
return [X_1,...X_d], [Y_1,...Y_d]

A.3.2 Theoretical justification for label MI approximation of high-dimensional MI

For the benchmarking setup described in Section 3.2, we have $L_y \to L_x \to X$ and $L_x \to L_y \to Y$. We will show that $I(X;Y) = I(L_x; L_y)$ under the condition that $H(L_x|X) = H(L_y|Y) = 0$.

Theorem 4. *Let L_x, L_y be Bernoulli random variables. Let X, Y be absolutely continuous random vectors in $\mathbb{R}^N$ such that $I(L_x; Y|L_y) = I(L_y; X|L_x) = 0$ and $H(L_x|X) = H(L_y|Y) = 0$. Then,*

$$I(X;Y) = I(L_x; L_y) \tag{20}$$

Proof. Due to conditional independence,

$$I(L_x; L_y) = I(L_x; L_y) + I(L_x; Y|L_y) \tag{21}$$

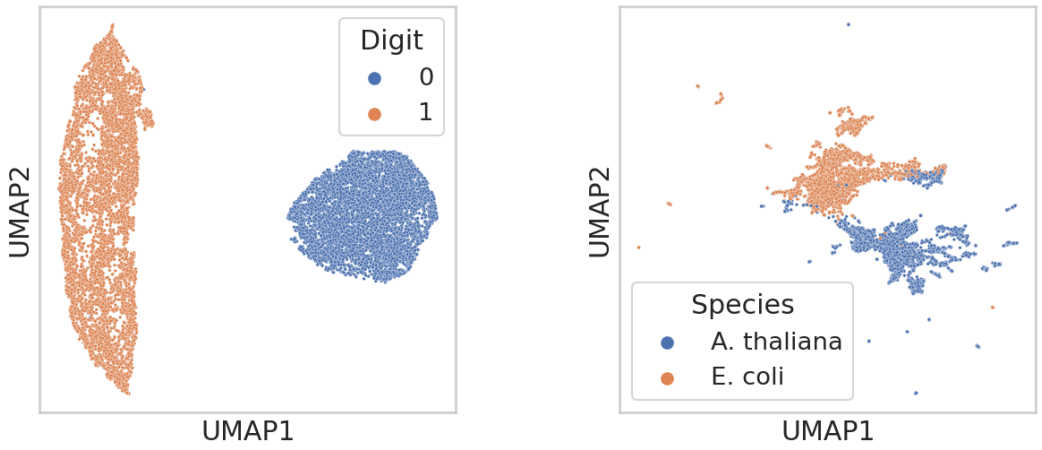
Using the chain rule for mutual information [7],

$$I(L_x; L_y) = I(L_x; L_y) + I(L_x; Y|L_y) = I(L_x; Y) + I(L_x; L_y|Y) \tag{22}$$

Applying the chain rule again,

$$I(L_x; L_y) = I(L_x; Y) + I(L_x; L_y|Y) = I(X; Y) + I(L_x; Y|X) + I(L_x; L_y|Y) \tag{23}$$

 https://doi.org/10.52202/079017-4223



(a) UMAP of binary MNIST subset.

(b) UMAP of proteome subsets.

Figure 12: Validating assumptions of cluster-based benchmarking, by visualizing separation of clusters.

Due to the non-negativity of mutual information we have

$$I(L_x; L_y) \geq I(X; Y) \tag{24}$$

Because L_x, L_y are discrete, we can bound

$$I(L_x; L_y) \leq I(X; Y) + H(L_x|X) + H(L_y|Y) \tag{25}$$

So if $H(L_x|X) = H(L_y|Y) = 0$, we have

$$I(L_x; L_y) \leq I(X; Y) \tag{26}$$

Combining this with (24), we have

$$I(X; Y) = I(L_x; L_y) \tag{27}$$

□

A.3.3 Validating the assumptions of cluster-based benchmarking setups

The effectiveness of the benchmarking setup in Section 3.2 relies on the assumption that $H(L_x|X) \approx H(L_y|Y) \approx 0$. We provide evidence that this is the case for MNIST 0s and 1s, and sequence embeddings of *E. Coli* and *A. Thaliana* proteins. First, we provide qualitative evidence by showing that label clusters (digits and species respectively) are well separated on UMAP visualizations of each dataset (Fig. 12). This indicates that the label of a sample can be reliably determined from its high-dimensional representation (such that $H(L_x|X) \approx 0$). We make this notion more quantitatively precise by showing that logistic regression can predict the labels of held-out samples with high accuracy. Over 100 random $1:1$ train-test splits, logistic regression achieves mean validation accuracy > 0.98 with standard error of mean $< 10^{-3}$ for both datasets. Logistic regression classifiers are trained with L_2 penalty and $\lambda = 1$.

A.3.4 Synthetic multivariate Gaussian benchmarking with nonlinear transformations

It is known that benchmarking MI estimators on Gaussian synthetic data has many shortcomings [12]. In addition to addressing this with the resampling-based benchmarking approch in Section 3.2, we also perform a subset of the multivariate Gaussian benchmarks after data has been nonlinearly transformed.

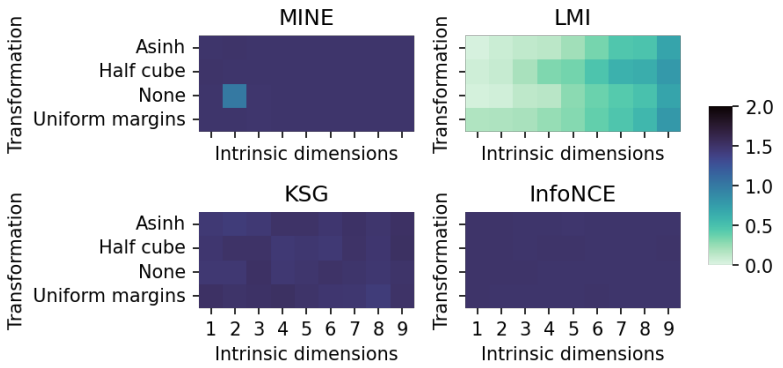


Figure 13: Performance (measured as MSE) on subset of multivariate Gaussian benchmark after non-linear transformations (defined in [12]). All estimation problems are with 1000 ambient dimensions.

Rather than using the exact tasks from [12], we have developed complementary high-dimensional benchmarks by applying some of the transforms proposed in [12] to our multivariate Gaussians from Fig. 2. This allows us to create versions of the existing Fig. 2 with "half-cube", "asinh", and "uniform marginal" distributions (as defined in [12]). We explore only the most challenging settings from Fig. 2: those with 1000 ambient dimensions and 1-9 intrinsic dimensions. We find that LMI performance is qualitatively similar, though not identical, for untransformed data and all three transformations (Fig. 13).

A.3.5 Sampling procedure for symmetric and exclusive variables

To sample from symmetric and exclusive variables (section 3.3, Fig. 5), we concatenate independent normally distributed dimensions to symmetric variables. Below, we give the procedure for sampling from symmetric variables.

Noisy circle For each sample, an angle θ is drawn uniformly from the interval $[0, 2\pi]$. For each angle, the corresponding x- and y-coordinates of each sample are $x = \cos(\theta) + \epsilon_x$ and $y = \sin(\theta) + \epsilon_y$, where ϵ_x and ϵ_y are independently sampled from a uniform distribution in the range $[-0.25, 0.25]$. This results in points approximately lying on the circumference of a unit circle, with random noise.

Gaussian mixture We sample from a two component Gaussian mixture, with each component having equal weight and covariance matrices $\begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$, and $\begin{bmatrix} 1 & -0.9 \\ -0.9 & 1 \end{bmatrix}$ and both with mean $\mu = 0$.

https://doi.org/10.52202/079017-4223

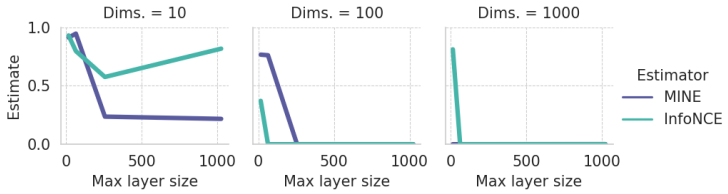


Figure 14: Variational bound estimators with increasing critic complexity, evaluated on multivariate Gaussians with 1-dimensional dependence structure. Each estimation problem has $5 \cdot 10^3$ samples and ground truth MI of 1 bit.

A.4 Experimental reproducibility details

Here, we will briefly summarize key details necessary for the reproduction of experiments in this paper. All of this information, and more details, can be found (albeit in a less easily readable form) in our code supplement.

A.4.1 Preprocessing ProtTrans5 embeddings

We downloaded ProtTrans5 embeddings of all *H. sapiens*, *A. thaliana*, and *E. coli* proteins directly from the UniProt database. Embeddings are from `prottrans_t5_xl_u50` [36]. All proteins longer than $12 \cdot 10^3$ residues are excluded. These embeddings are then unit variance normalized, and values are clipped at 10 and -10.

A.4.2 Preprocessing hematopoiesis lineage tracing scRNA-seq data

We first downloaded all data from Experiment 1 of the data repository from [42]. Then, we preprocessed using the Scanpy best practices [53], normalizing total reads per cell to 10^4, log transforming, filtering for the 10^3 most highly variable genes, and finally unit variance normalizing and clipping values at 10 and -10. We used the inferred diffusion pseudotime and SPRING embeddings computed in [42]. For pseudotime analysis, we omit cells with pseudotime value below 10^4, because many are Lymphoid-fated rather than Neutrophil-fated. To generate joint samples of clones between two conditions, we identified all clonal barcodes that appeared in both conditions, and randomly sampled a single cell with each barcode from each condition. Because there are often several cells with the same clonal barcode in the same sample, there are many possible random pairings of clonally related cells.

A.4.3 MINE and InfoNCE implementation details

We use the implementations of MINE and InfoNCE from [12], with their default parameter choices. To summarize, architectures have two hidden layers with sizes (16, 8) and are optimized using Adam.

Choosing critic architectures Because the benchmark tasks of [12] are dramatically different from those considered in this work (tens of dimensions as opposed to thousands), we consider the possibility that the parameters of MINE and InfoNCE used in [12] are not suitable for our use case.

To determine if other critic architectures could be more suitable, we first tested two layer architectures with layer sizes $(L, L/2)$ for various L from 16 to 1024 on multivariate Gaussian data sets. We considered variables with dimensions 10, 100, and 1000, with 1 bit MI, 1-dimensional dependence structure, and $5 \cdot 10^3$ samples. We find that increasing L does not improve estimation quality, and the architecture used by [12] indeed is optimal for all three tested settings (Fig. 14). We suspect that this is because increasing model complexity does not help when sample size remains small.

On the MNIST benchmarking setup, we verify that the effectiveness of LMI estimation over MINE and InfoNCE are not merely from model complexity, by using critics with the same complexity as the LMI encoders. In line with the hypothesis that large critic architectures are not suitable for the small sample size regime, we find that with high complexity critics, MINE and InfoNCE fail entirely on the MNIST benchmark (Fig. 15).

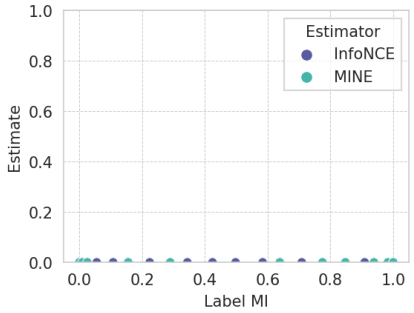


Figure 15: MNIST benchmarking for neural estimators with critic complexity equivalent to LMI encoders, over 20 datasets with true MI between 0 and 1.

A.4.4 KSG implementation details for pointwise decompositions

We implement KSG with $k = 3$ nearest neighbors. For protein interaction data, due to large sample numbers resulting in high computational cost, we obtain KSG estimates by averaging over estimates on batches of data containing 10^3 samples.

We slightly adjust the KSG estimator to yield pointwise mutual information estimates. While the original KSG estimator [10] takes a sample expectation over computed pointwise mutual information values, we simply return an array of pointwise estimates rather than the average. For completeness, the algorithm is given in Algorithm 5.

Algorithm 5 KSG estimator for pointwise estimates

Require: joint samples $\{(x_i, y_i)\}_{i=1}^{N}$
Require: parameter k (default $k = 3$)
  let `pmis` $\leftarrow$ [ ]
  for each (x_i, y_i) **do**
    find k-th nearest neighbor in joint space (x_k, y_k)
    compute Chebyshev distance $d = ||(x_k, y_k) - (x_i, y_i)||_\infty$
    let $n_x \leftarrow 0, n_y \leftarrow 0$
    for each (x_j, y_j) **do**
      if $||x_j - x_i||_\infty < d$ **then**
        n_x += 1
      end if
      if $||y_j - y_i||_\infty < d$ **then**
        n_y += 1
      end if
    end for
    `pmis.append(` $\psi(k) + \psi(N) - \psi(n_x) - \psi(n_y)$ `)`
  end for
  return `pmis`

https://doi.org/10.52202/079017-4223

A.5 Choosing appropriate latent space size

All estimates in the main text of this paper is made with 8 latent dimensions per variable. Without optimizing this parameter, we find that LMI performs reasonably well compared to other estimators on diverse benchmarks and real-world problems. However, the choice of 8 latent dimensions per variable is somewhat arbitrary, and certainly not optimal for most estimation problems. Here, we outline some principles for how one might choose a more optimal number of latent dimensions.

A.5.1 General principles for choosing latent space size

Intuitively, there is a tradeoff which arises when changing latent space size. As the latent space gets larger, the capacity of the compressed representation increases, and we might expect that representation quality increases (with the caveat that representation quality can be limited by sample sparsity). However, as the latent space gets larger, MI estimation in latent space becomes more difficult, due to the curse of dimensionality for nonparametric MI estimators [14]. As such, the ideal choice is the smallest possible latent space size which captures the dependence structure of the variables. In practice, this size can be difficult to determine in a rigorous way. Instead, we suggest a heuristic approach.

A.5.2 Heuristic approach to choosing latent space size

From Theorem 2, a simple extension of the data processing inequality, we know that $I(Z_x; Z_y) \leq I(X; Y)$. With the caveat that the inequality is not guaranteed to hold for the estimated $\hat{I}(Z_x; Z_y)$, we can reason that the parameter choices that maximize $\hat{I}(Z_x; Z_y)$ are likely ideal. As such, one sensible way to choose a latent space size is to try several, and use that which yields the largest estimate.

As an example, let us consider multivariate Gaussian data with 4-dimensional dependence structure, in 1000D ambient space, with ground truth MI of 1 bit, and $5 \cdot 10^3$ samples (generated analogously to Figure 2). The most accurate estimate comes from LMI with optimal choice of 4 latent dimensions, but all tested choices (including the 2D latent space which cannot fully capture dependence) improve over MINE and InfoNCE. If we use the heuristic approach, we would choose the optimal latent space size. If we had arbitrarily chosen 8, we would be within 5% of the "optimal" 4 latent dimension estimate.

	LMI-2	LMI-4	LMI-6	LMI-8	MINE	InfoNCE
Estimate	0.295732	0.719762	0.670632	0.686974	-0.000001	-0.001417

Table 2: LMI estimates with varying latent space size, with k dimensions per variable denoted as LMI-k. Data is multivariate Gaussian, generated as in Figure 2, with $d = 10^3$, $k = 4$, and $N = 5 \cdot 10^3$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The results stated in the abstract and introduction are summaries of the experimental results shown in Sections 3 and 4 (figures 2-6).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: There is a subsection of the Discussion dedicated to limitations of the LMI approach. These limitations are considered throughout the paper. For instance, in the abstract, we highlight the necessity of low dimensional intrinsic dependence structure.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-4223

Answer: [Yes]

Justification: Assumptions about distributions (such as absolute continuity, finite entropies) are specified in theoretical analysis (Theorems 1-4).

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Key details necessary to reproduce the experimental results are given in the appendix. All experiments can be reproduced using the supplementary code. It is carefully annotated to match each result in the paper to a specific Jupyter notebook.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: As stated before, all experiments can be reproduced using the supplementary code. It is carefully annotated to match each result in the paper to a specific Jupyter notebook. The code reproducibility "best practices" were followed.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and test details are provided in the Appendix sections on implementation, and can be found in the supplementary code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report SEMs and standard deviations in the text when relevant and feasible. No plots contain error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/079017-4223

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the hardware used for all experiments, and the projected overall environmental impact of all experiments (failed and pilot) reported in the paper. We do not provide further granularity because the experiments have rather modest compute requirements – the entirety of the experiments in this paper can be reproduced using a commercial NVIDIA GPU (RTX 3090) in about one day (by running every Jupyter notebook in the code supplement).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed and adhered to the guidelines. We mention potential social and environmental impacts in the Discussion section.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We discuss the potential social impact of MI estimators.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release data or models with high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the library used to implement the LMI approximation (PyTorch).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

https://doi.org/10.52202/079017-4223

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code developed for latent MI approximation (the `lmi` library provided in the supplement) is well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve human subjects or crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve human subjects or crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-4223