

PAPER • OPEN ACCESS

A robust estimator of mutual information for deep learning interpretability

To cite this article: Davide Piras *et al* 2023 *Mach. Learn.: Sci. Technol.* **4** 025006

View the [article online](#) for updates and enhancements.

You may also like

- [CTformer: convolution-free Token2Token dilated vision transformer for low-dose CT denoising](#)
Dayang Wang, Fenglei Fan, Zhan Wu et al.
- [IDP-PGFE: an interpretable disruption predictor based on physics-guided feature extraction](#)
C. Shen, W. Zheng, Y. Ding et al.
- [MIDRC CRP10 AI interface—an integrated tool for exploring, testing and visualization of AI models](#)
Naveena Gorre, Eduardo Carranza, Jordan Fuhrman et al.



PAPER

OPEN ACCESS

RECEIVED

25 November 2022

REVISED

28 February 2023

ACCEPTED FOR PUBLICATION

14 March 2023

PUBLISHED

11 April 2023

Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



A robust estimator of mutual information for deep learning interpretability

Davide Piras^{1,2,*} , Hiranya V Peiris^{1,3}, Andrew Pontzen¹, Luisa Lucie-Smith⁴, Ningyuan Guo¹ and Brian Nord^{5,6,7}

¹ Department of Physics & Astronomy, University College London, Gower Street, London WC1E 6BT, United Kingdom

² Département de Physique Théorique, Université de Genève, 24 Quai Ernest Ansermet, 1211 Genève 4, Switzerland

³ The Oskar Klein Centre for Cosmoparticle Physics, Department of Physics, Stockholm University, AlbaNova, Stockholm SE-10691, Sweden

⁴ Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Str. 1, 85748 Garching, Germany

⁵ Fermi National Accelerator Laboratory, PO Box 500, Batavia, IL 60510, United States of America

⁶ Department of Astronomy & Astrophysics, University of Chicago, Chicago, IL 60637, United States of America


⁷ Kavli Institute for Cosmological Physics, University of Chicago, Chicago, IL 60637, United States of America

* Author to whom any correspondence should be addressed.

E-mail: d.piras@ucl.ac.uk

Keywords: deep learning, mutual information, interpretability, representation learning

Abstract

We develop the use of mutual information (MI), a well-established metric in information theory, to interpret the inner workings of deep learning (DL) models. To accurately estimate MI from a finite number of samples, we present GMM-MI (pronounced ‘Jimmie’), an algorithm based on Gaussian mixture models that can be applied to both discrete and continuous settings. GMM-MI is computationally efficient, robust to the choice of hyperparameters and provides the uncertainty on the MI estimate due to the finite sample size. We extensively validate GMM-MI on toy data for which the ground truth MI is known, comparing its performance against established MI estimators. We then demonstrate the use of our MI estimator in the context of representation learning, working with synthetic data and physical datasets describing highly non-linear processes. We train DL models to encode high-dimensional data within a meaningful compressed (latent) representation, and use GMM-MI to quantify both the level of disentanglement between the latent variables, and their association with relevant physical quantities, thus unlocking the interpretability of the latent representation. We make GMM-MI publicly available in this [GitHub repository](#). 

1. Introduction

The flexibility and expressiveness of deep learning (DL) models are attractive features, which have led to their application to a variety of scientific problems (see e.g. Raghu and Schmidt [1] for a recent review). Despite this recent progress, deep neural networks remain opaque models, and their power as universal approximators [2–4] comes at the expense of interpretability [5]. Many techniques have been developed to gain insight into such black-box models [6–13]. These solutions vary in their computational efficiency and in the range of tasks to which they can be applied; however, there is no consensus as to which method provides the most trustworthy interpretations, and a general framework to interpret deep neural networks is still an avenue of active investigation (see e.g. Li *et al* [14], Linardatos *et al* [15] for recent reviews).

DL models are also widely used in representation learning, where a high-dimensional dataset is compressed to a smaller set of variables; this latent representation should contain all the relevant information for downstream tasks such as reconstruction, classification or regression [16, 17]. Disentanglement of these compressed variables is also often imposed, in order to associate each latent to a physical quantity of domain interest [17–24]. However, how best to access the information captured by these latent vectors and connect it to the relevant factors remain open questions.

In this work, we focus on representation learning and link the latent variables to relevant physical quantities by estimating their mutual information (MI), a well-established information-theoretic measure of the relationship between two random variables. MI allows us to interpret what the DL model has learned about the domain-specific parameters relevant to the problem: by interrogating the model through MI, we aim to discover what information is used by the model in making predictions, thus achieving the interpretation of its inner workings. We also use MI to quantify the level of disentanglement of the latent variables.

MI has found applications in a variety of scientific fields, including astrophysics [25–33], biophysics [34–40], and dynamical complex systems [41–48], to name a few. However, estimating the MI $I(X, Y)$ between two random variables X and Y , given samples from their joint distribution $p_{(X,Y)}$, remains a long-standing challenge, since it requires $p_{(X,Y)}$ to be known or estimated accurately [49, 50]. When X and Y are continuous variables with values over $\mathcal{X} \times \mathcal{Y}$, $I(X, Y)$ is defined as:

$$I(X, Y) \equiv \int_{\mathcal{X} \times \mathcal{Y}} p_{(X,Y)}(x, y) \ln \frac{p_{(X,Y)}(x, y)}{p_X(x)p_Y(y)} dx dy, \quad (1)$$

where p_X and p_Y are the marginal distributions of X and Y , respectively, and \ln refers to the natural logarithm, so that MI is measured in natural units (nat). $I(X, Y)$ represents the amount of information one gains about Y by observing X (or vice versa): it captures the full dependence between two variables going beyond the Pearson correlation coefficient, since $I(X, Y) = 0$ if and only if X and Y are statistically independent [51]. A comprehensive summary of MI and its properties can be found in Vergara and Estévez [50].

The most straightforward estimator of $I(X, Y)$ given samples of $p_{(X,Y)}$ consists of binning the data and approximating equation (1) with a finite sum over the bins. This approach is heavily dependent on the binning scheme, and is prone to systematic errors [39, 52–59]. Kraskov *et al* [56] proposed an estimator (hereafter referred to as KSG), based on k -nearest neighbors, which rewrites $I(X, Y)$ in terms of the Shannon entropy, and then applies the Kozachenko–Leonenko entropy estimator [60] to calculate each term (see section 2.2 for more details). However, the KSG estimator only returns a point estimate, is strongly dependent on the number of chosen neighbors, and does not scale well with sample size [61]. Bayesian approaches to obtain the full distribution of MI have also been discussed [62–64], but they are not easily applicable to continuous data, and have been shown to be strongly dependent on the chosen prior [64].

More recently, MI estimators based on bounds approximated by neural networks have gained interest [19, 65–76]. In particular, Belghazi *et al* [69] proposed a neural estimator of $I(X, Y)$ (hereafter referred to as MINE) rewriting it as a Kullback–Leibler (KL) divergence [77], and considering its Donsker–Varadhan representation [78] (see section 2.2 for more details). While yielding differentiable MI estimates (essential e.g. for backpropagation when training DL models), such neural-network-based estimators do not necessarily return an accurate estimate of equation (1), are heavily dependent on the training hyperparameters, and have been shown to suffer from a poor variance-bias tradeoff [75]. The use of MI estimates for interpreting deep representation learning has recently been investigated as well [19, 32, 79, 80]; however, exploiting MI to interpret deep representation learning requires a robust density estimate of the joint probability distribution between latent variables and relevant physical parameters, and the uncertainties on the MI estimate to be quantified, ensuring that any trends in MI are statistically significant.

To address these requirements, we present Gaussian mixture model (GMM)-MI (pronounced ‘Jimmie’), an algorithm to estimate the full distribution of $I(X, Y)$ based on fitting samples drawn from the distribution with GMMs. While the use of GMMs to estimate MI is not new [81–86], these previous works only considered MI in the context of feature selection, and did not carry out uncertainty quantification on the relevant MI estimates, which is critical when using MI to interpret DL models. GMM-MI has been designed to be a robust and flexible tool that can be applied to multiple settings where MI estimation is required. Crucially, it also returns error estimates which we verified to be statistically correct on test datasets including bivariate distributions of various shapes and non-linear transformations of Gaussian samples. We first extensively validate GMM-MI on these toy data for which the ground truth MI is known, including comparisons to the KSG and MINE estimators in terms of both efficiency and accuracy, additionally showing that GMM-MI is unbiased and the MI uncertainty scales as expected with the sample size. We then train representation-learning models on high-dimensional datasets including simulations of dark matter halos formed through non-linear physical processes, real astrophysical spectra and synthetic shape images with known labels. We demonstrate the use of GMM-MI to achieve the interpretability of such models.

The paper is structured as follows. In section 2.1 we describe GMM-MI, and recall the essential details of the KSG and MINE estimators in section 2.2. In section 3, we present extensive experiments where we validate our MI estimator on toy data, and then in section 4 we use MI to interpret the latent space of DL

models trained on synthetic and real data. We conclude in section 5, including an outlook over planned extensions of our algorithm.

2. Method

2.1. Estimation procedure (GMM-MI)

Our algorithm uses a GMM with c components to obtain a fit of the joint distribution $p_{(X,Y)}$:

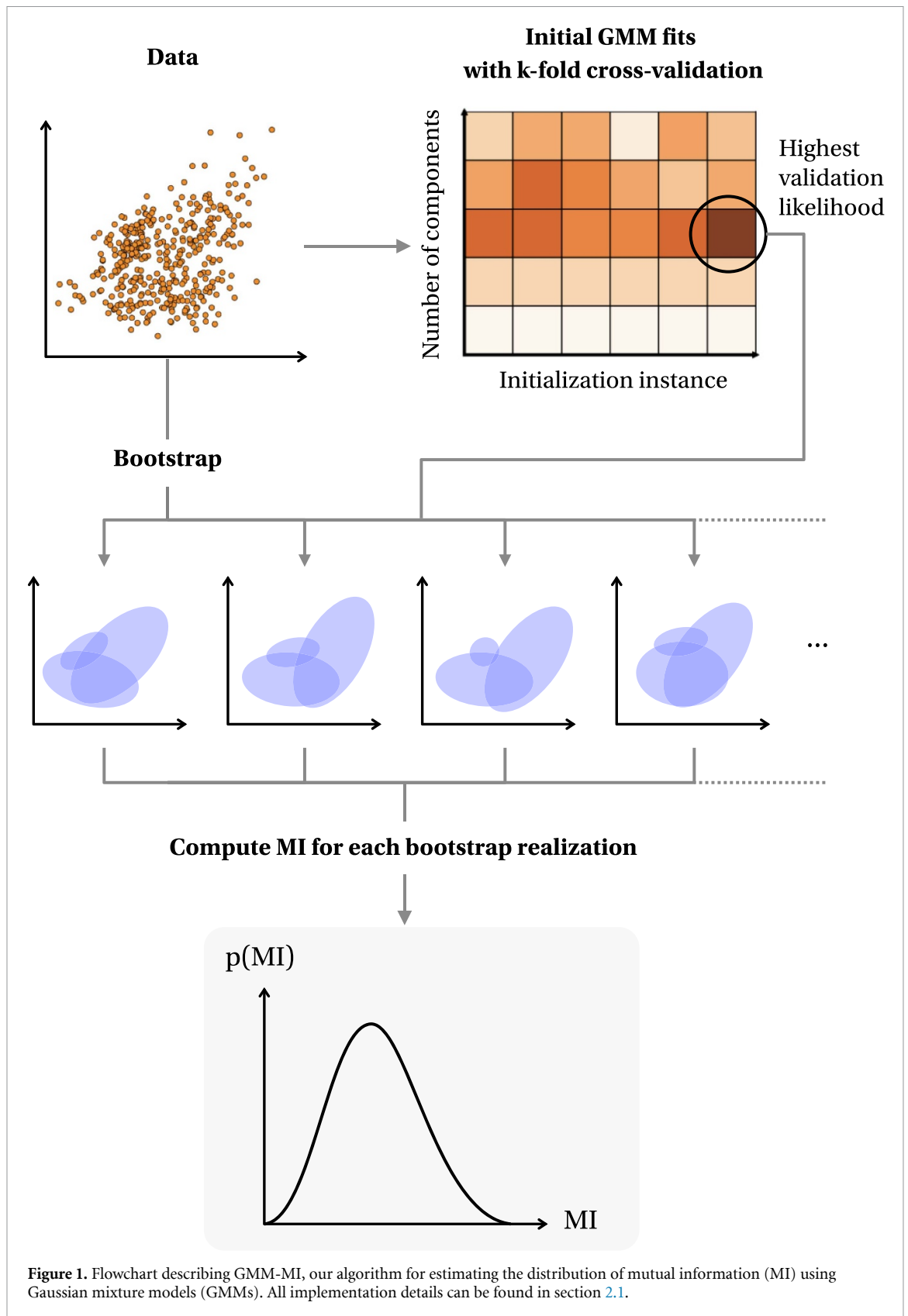
$$p_{(X,Y)}(x, y | \theta) = \sum_{i=1}^c w_i \mathcal{N}(x, y | \mu_i, \Sigma_i), \quad (2)$$

where θ is the set of weights $w_{1:c}$, means $\mu_{1:c}$ and covariance matrices $\Sigma_{1:c}$. With this choice, the marginals $p(x)$ and $p(y)$ are also GMMs, with parameters determined by θ . Our procedure for estimating MI and its associated uncertainty is as follows.

1. For a given number of GMM components c , we randomly initialize n_{init} different GMM models. Each set of initial GMM parameters is obtained by first randomly assigning the responsibilities, namely the probabilities that each point belongs to a component i , sampling from a uniform distribution. The starting values of each μ_i and Σ_i are calculated as the sample mean and covariance matrix of all points, weighted by the responsibilities, while each w_i is initialized as the average responsibility across all points. Having multiple initializations is crucial to reduce the risk of stopping at local optima during the optimization procedure [87–91].
2. We fit the data using k -fold cross-validation: this means that we train a GMM on $k - 1$ subsets of the data (or ‘folds’), and evaluate the trained model on the remaining validation fold. Each fit is performed with the expectation-maximization algorithm [92], and terminates when the change in log-likelihood on the training data is smaller than a chosen threshold. We also add a small regularization constant ω to the diagonal of each covariance matrix, as described e.g. in Melchior and Goulding [91], to avoid singular covariance matrices.
3. We select the model with the highest mean validation log-likelihood across folds $\hat{\ell}_c$, since it has the best generalization performance. Among the k models corresponding to $\hat{\ell}_c$, we also store the final GMM parameters with the highest validation log-likelihood on a single fold: these will be used to initialize each bootstrap fit in step 5, thus reducing the risk of stopping at local optima and significantly accelerating convergence.
4. We repeat steps 1–3 iteratively increasing the number of GMM components from $c = 1$. We stop when $\hat{\ell}_c - \hat{\ell}_{c-1}$ is smaller than a user-specified positive threshold, and select the value of $c - 1$ as the optimal number of GMM components to fit. In this way, we avoid overfitting the training data and adding too many components, which would considerably slow down the procedure while not significantly improving the density estimation.
5. We bootstrap the data n_b times, and fit a GMM to each bootstrapped realization. Each fit is initialized with the set of parameters selected in step 3, and with the number of components found in step 4. We use bootstrap to capture not just a point estimate of MI, but its full distribution.
6. For each fitted model, we calculate MI by solving the integral in equation (1) using Monte Carlo (MC) integration over M samples.
7. We return the sample mean and standard deviation of the distribution of MI values.

A flowchart summarizing the GMM-MI procedure is shown in figure 1. We choose the initialization procedure described in step 1 for its speed, but in our implementation of GMM-MI other initialization procedures are also available and could be alternatively used. For instance, it is possible that the random initialization we set as default returns overlapping components which inhibit the optimization procedure; in those cases, we recommend switching to an initialization based on k -means [93]. On the other hand, k -means itself is known to only guarantee convergence to local optima [94]; for this reason, we also provide the possibility to perturb the means by a user-specified scale after an initial call to k -means. We call this approach ‘randomized k -means’, and offer full flexibility to select the most appropriate initialization type based on the data being analyzed.

Our implementation also allows the user to set a higher patience, i.e. consider more than one additional component in step 4 after the validation loss has started to decrease; alternatively, it is possible to select the number of components yielding the lowest Akaike information criterion (AIC, [95]) or Bayesian information criterion (BIC, [96]), with details in appendix A. All three methods implemented are computationally efficient, and aim to prevent the model from overfitting the available samples; in figure 8 we further show that even in a case where the three metrics disagree on the number of GMM components to use, the final MI estimates agree with each other within the uncertainties, thus demonstrating that GMM-MI



is robust to the metric being used. The number of folds (k) should be set based on the number of available samples, so that each fold is representative of the data. The number of initializations (n_{init}), bootstrap realizations (n_b), and MC samples (M) should be chosen based on the available computational budget.

In many instances, the factors of variation that are used to generate the data are discrete variables [97]; in these cases, we will need to estimate MI between a continuous variable X and a categorical variable F which can take ν different values $f_{1:\nu}$. In this case, assuming the ν values have equal probability (as will be the case

when considering the 3D shapes dataset in section 4.1), the MI $I(X, F)$ can be expressed as:

$$I(X, F) = \frac{1}{v} \sum_{i=1}^v \int_{\mathcal{X}} dx p_{(X|F)}(x|f_i) \left[\ln p_{(X|F)}(x|f_i) - \ln \frac{1}{v} \sum_{j=1}^v p_{(X|F)}(x|f_j) \right], \quad (3)$$

where we use a GMM to fit each conditional probability $p_{(X|F)}(x|f_i)$. The full derivation of equation (3) can be found in appendix B.

2.2. Alternative estimators

In order to validate our algorithm, we compare it with two established estimators of MI. The KSG estimator, first proposed in Kraskov *et al* [56], rewrites MI as:

$$I(X, Y) = H(X) + H(Y) - H(X, Y), \quad (4)$$

where $H(\cdot)$ refers to the Shannon entropy, defined for a single variable as:

$$H(X) \equiv - \int_{\mathcal{X}} p_X(x) \ln p_X(x). \quad (5)$$

The Kozachenko–Leonenko estimator [60] is then used to evaluate the entropy in equation (5):

$$\widehat{H}(X) = -\psi(k) + \psi(N) + \ln c_d + \frac{d}{N} \sum_{i=1}^N \ln \epsilon^{(k)}(i), \quad (6)$$

where $\psi(\cdot)$ is the digamma function, k is the chosen number of nearest neighbors, N is the number of available samples, d is the dimensionality of X , c_d is the volume of the unit ball in d dimensions, and $\epsilon^{(k)}$ is twice the distance between the i th data point and its k th neighbor. Applying equation (6) to each term in equation (4) would lead to biased estimates of MI [39, 56]; for this reason, the KSG estimator actually considers a ball containing the k -nearest neighbors around each sample, and counts the number of points within it in both the x and y direction. The resulting estimator of MI then becomes [39, 56]:

$$\widehat{I}(X, Y) = \psi(k) + \psi(N) - \frac{1}{k} - \langle \psi(n_x^{(k)}) + \psi(n_y^{(k)}) \rangle, \quad (7)$$

where $n_x^{(k)}$ ($n_y^{(k)}$) represents the number of points in the x (y) direction, and $\langle \cdot \rangle$ indicates the mean over the available samples. In our experiments, we consider the implementation of the KSG estimator available from SKLEARN in this [https link](#).

We also compare our algorithm against the MINE estimator proposed in Belghazi *et al* [69]. MI as defined in equation (1) can be interpreted as the KL divergence D_{KL} between the joint distribution and the product of the marginals:

$$I(X, Y) = D_{\text{KL}} [p_{(X,Y)} || p_X p_Y], \quad (8)$$

where the KL divergence between two generic probability distributions p_X and q_X defined over \mathcal{X} is defined as:

$$D_{\text{KL}} [p_X || q_X] \equiv \int_{\mathcal{X}} dx p_X(x) \ln \frac{p_X(x)}{q_X(x)}. \quad (9)$$

The MINE estimator then considers the Donsker–Varadhan representation [78] of the KL divergence:

$$D_{\text{KL}} [p_X || q_X] = \sup_T \mathbb{E}_{p_X} [T] - \ln \mathbb{E}_{q_X} [e^T], \quad (10)$$

where the supremum is taken over all the functions T such that the expectations $\mathbb{E}[\cdot]$ are finite, and parameterizes T with a neural network. In our experiments, we consider the implementation available in this [https link](#), which includes the mitigation of the gradient bias through the use of an exponential moving average, as suggested in Belghazi *et al* [69].

2.3. Representation learning

We apply our MI estimator GMM-MI to interpret the latent space of representation-learning models. Specifically, we consider β -variational autoencoders (β -VAEs, [21, 98]), where one neural network is trained to encode high-dimensional data D into a distribution over disentangled latent variables \mathbf{z} , and a second network decodes samples of the latent distribution back into data points \tilde{D} . The two networks are trained together to minimize the following loss function:

$$\mathcal{L} = \text{MSE}(D, \tilde{D}) + \beta D_{\text{KL}} [p_{\phi}(\mathbf{z}|D) || p(\mathbf{z})] , \quad (11)$$

where MSE indicates the mean squared error, $p_{\phi}(\mathbf{z}|D)$ represents the encoder parameterized by a set of weights ϕ , $p(\mathbf{z})$ is the prior over the latent variables \mathbf{z} , and β is a regularization constant which controls the level of disentanglement of \mathbf{z} .

We will also reproduce the results of Lucie-Smith *et al* [32] in section 4.2, for which the architecture is slightly different: the latent samples are combined with a given query (the radius r) and fed through the decoder to predict dark matter halo density profiles at each given r . This model is referred to as the interpretable variational encoder (IVE), with an analogous loss function to equation (11).

3. Validation

In this section, we validate GMM-MI on toy data for which the MI can be computed analytically: we show that GMM-MI is in good agreement with the ground truth, as well as other MI estimators, while returning the full distribution of MI including its uncertainty. We run all the MI estimations on a single CPU node with 40 2.40 GHz Intel Xeon Gold 6148 cores using no more than 300 MB of RAM, reporting the speed performance in each case.

We first consider a bivariate Gaussian distribution with unit variance of each marginal and varying level of correlation $\rho \in [-1, 1]$, following Belghazi *et al* [69]. In this case, the true value of $I(X, Y)$ can be obtained analytically by solving the integral in equation (1), yielding:

$$I(X, Y)_{\text{true}} = -\frac{1}{2} \ln(1 - \rho^2) . \quad (12)$$

We consider two additional bivariate distributions, the gamma-exponential distribution [54, 56, 99, 100], with density ($\alpha > 0$ is a free parameter):

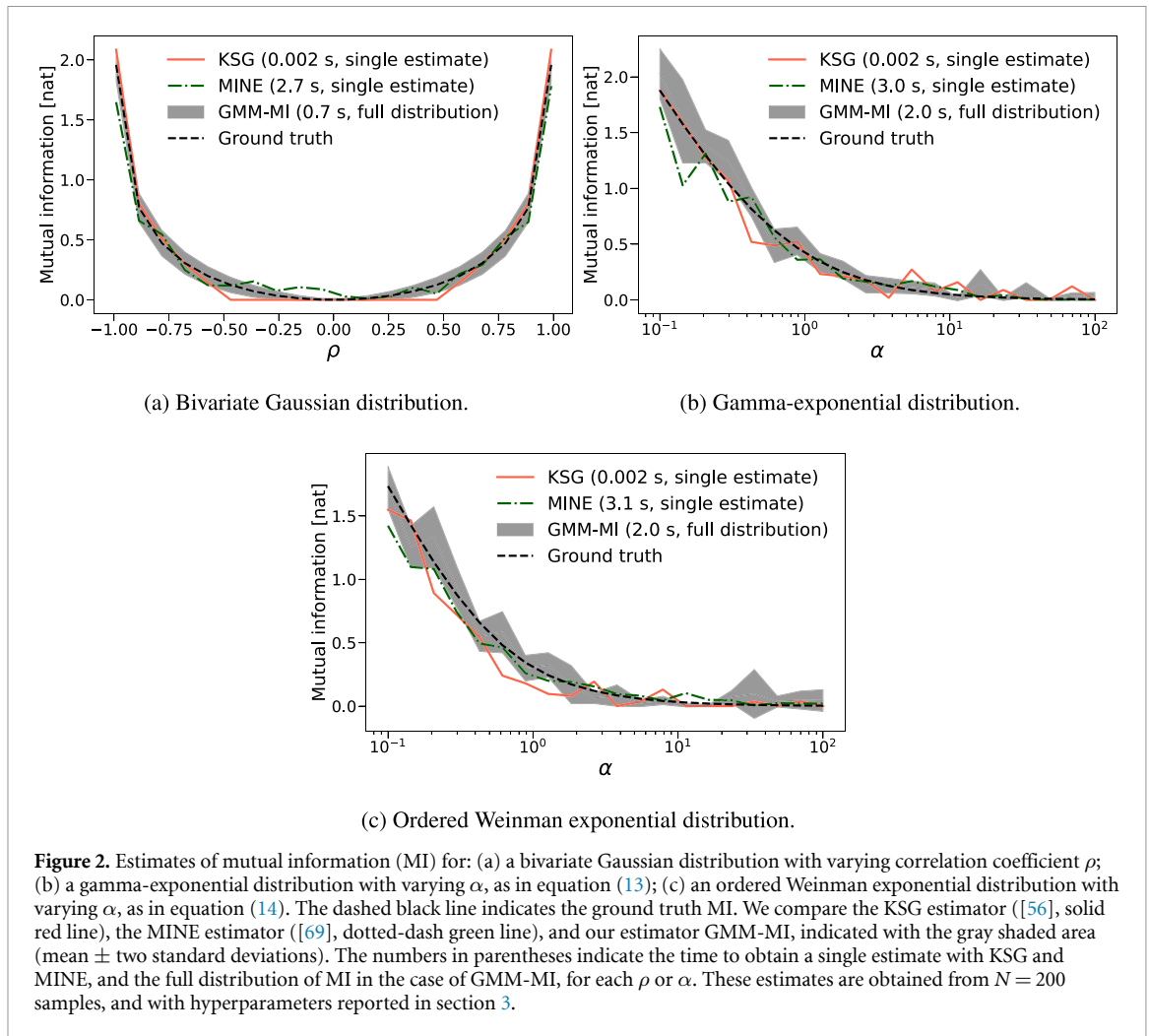
$$p_{(X,Y)}(x, y | \alpha) = \begin{cases} \frac{1}{\Gamma(\alpha)} x^{\alpha} e^{-x-xy} & x > 0, y > 0 \\ 0 & \text{otherwise} \end{cases} , \quad (13)$$

where Γ is the gamma function, and the ordered Weibull exponential distribution [54, 56, 99, 100], with density:

$$p_{(X,Y)}(x, y | \alpha) = \begin{cases} \frac{2}{\alpha} e^{-2x - \frac{y-x}{\alpha}} & y > x > 0 \\ 0 & \text{otherwise} \end{cases} . \quad (14)$$

The true value of $I(X, Y)$ for these distributions can be obtained analytically, and is reported in appendix C. Since $I(X, Y)$ is invariant under invertible transformations of each random variable [56], we consider $\ln(X)$ and $\ln(Y)$ when estimating MI in the case of the last two distributions [56]. To demonstrate the power of our estimator, we restrict ourselves to the case with only $N = 200$ samples. To estimate MI, we consider the KSG estimator with 1 neighbor (to minimize the bias, and following Kraskov *et al* [56]), the MINE estimator trained for 50 epochs with a learning rate of 10^{-3} and a batch size of 32, and our estimator GMM-MI with $k = 2$ folds, $n_{\text{init}} = 3$ different initializations, a log-likelihood threshold on each individual fit of 10^{-5} , a threshold on the mean validation log-likelihood to select the number of GMM components of 10^{-5} , $n_b = 100$ bootstrap realizations, $M = 10^4$ MC samples, and a regularization scale of $\omega = 10^{-12}$.

The results are reported in figure 2. The KSG estimator is the fastest, and yields MI values closely matching the ground truth, but returns biased estimates around e.g. $|\rho| = 0.4$ in the bivariate Gaussian case, and $\alpha \simeq 1$ in the ordered Weibull case. The MINE estimator is more computationally expensive and shows a relatively high variance, which is expected since MINE has been shown to be prone to variance overestimation due to the use of batches [72]. GMM-MI, on the other hand, returns a distribution of MI in good agreement with the ground truth in $\mathcal{O}(1)$ s, and provides an uncertainty estimate due to the finite sample size. We also found the results of GMM-MI to be robust to the choice of hyperparameters: changing the values of the likelihood threshold, MC samples, bootstrap realizations or regularization scale by one



order of magnitude, or doubling the number of folds and initializations, did not significantly change the results obtained with GMM-MI.

We further validate GMM-MI by testing that it is unbiased, and that the estimated MI variance scales as N^{-1} , when the number of available samples N increases. We additionally show that GMM-MI satisfies the MI property of invariance under invertible non-linear transformations [56]. We consider a bivariate Gaussian distribution with $\rho = 0.6$, and three different functions applied to one marginal variable Y : $f(y) = y$ (identity), $f(y) = y + 0.5y^3$ (cubic) and $f(y) = \ln(y + 5.5)$ (logarithmic). To deal with these datasets, we change the GMM-MI hyperparameters to $k = 3$, $n_{\text{init}} = 5$, and $M = 10^5$; however, we find no significant variations in the results even with different sets of hyperparameters. We repeat the estimation procedure of MI 500 times, drawing N samples with a different seed every time, and considering $N = 200$, $N = 2000$ and $N = 20000$. For each estimate, we calculate the bias, i.e. the difference between the estimated value of MI and the ground truth.

We report violin plots of the bias and of the MI standard deviation as returned by GMM-MI across the 500 trials in figure 3. The mean bias, indicated as a black cross, converges to 0 as N grows, and it is always well below the typical value of the standard deviation, thus demonstrating that GMM-MI is unbiased. This is true even when considering the cubic and the logarithmic transformations, further confirming that GMM-MI correctly captures the invariance property of MI. Moreover, in all cases the standard deviations returned by GMM-MI follow a power law $\propto \frac{1}{\sqrt{N}}$ as expected, represented as a gray line in the bottom plots. Remarkably, we found that even with very low numbers of samples ($N = 50$), GMM-MI returns MI values consistent with the ground truth, even when applying the non-linear transformations considered in this section.

3.1. A note on bootstrap

As reported in Holmes and Nemenman [39], using bootstrap to associate an error bar to MI estimates can lead to catastrophic failures: duplicate points can be interpreted as fine-scale features, introducing spurious

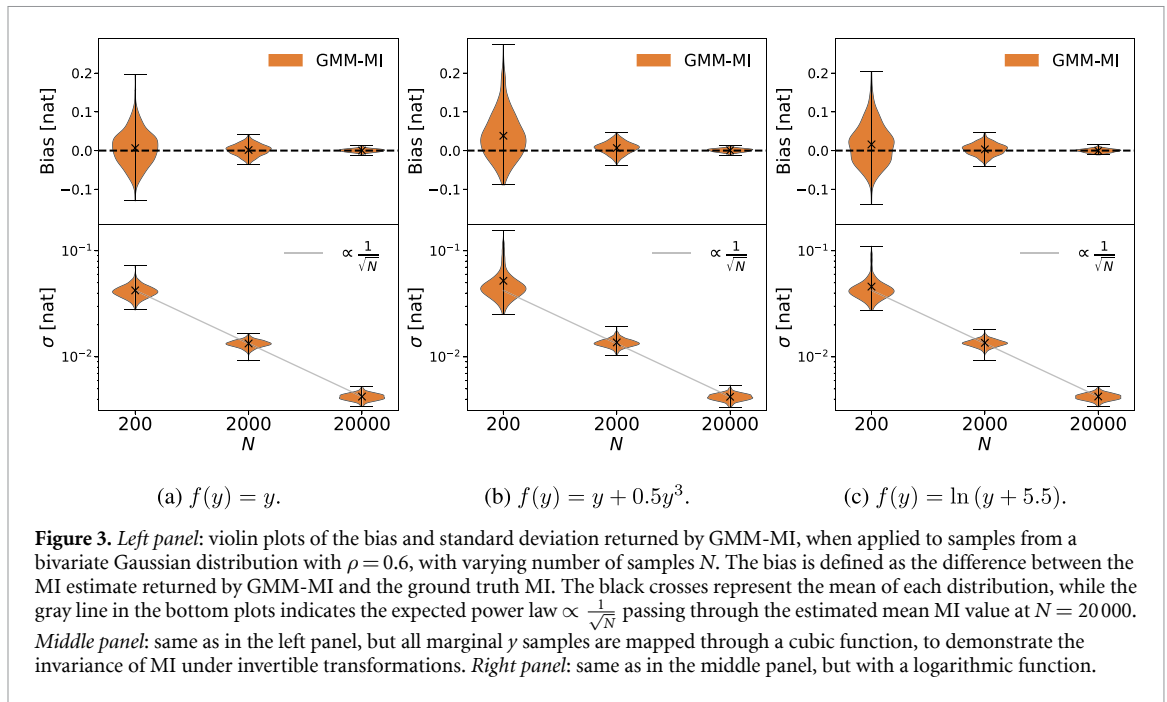


Figure 3. *Left panel:* violin plots of the bias and standard deviation returned by GMM-MI, when applied to samples from a bivariate Gaussian distribution with $\rho = 0.6$, with varying number of samples N . The bias is defined as the difference between the MI estimate returned by GMM-MI and the ground truth MI. The black crosses represent the mean of each distribution, while the gray line in the bottom plots indicates the expected power law $\propto \frac{1}{\sqrt{N}}$ passing through the estimated mean MI value at $N = 20000$. *Middle panel:* same as in the left panel, but all marginal y samples are mapped through a cubic function, to demonstrate the invariance of MI under invertible transformations. *Right panel:* same as in the middle panel, but with a logarithmic function.

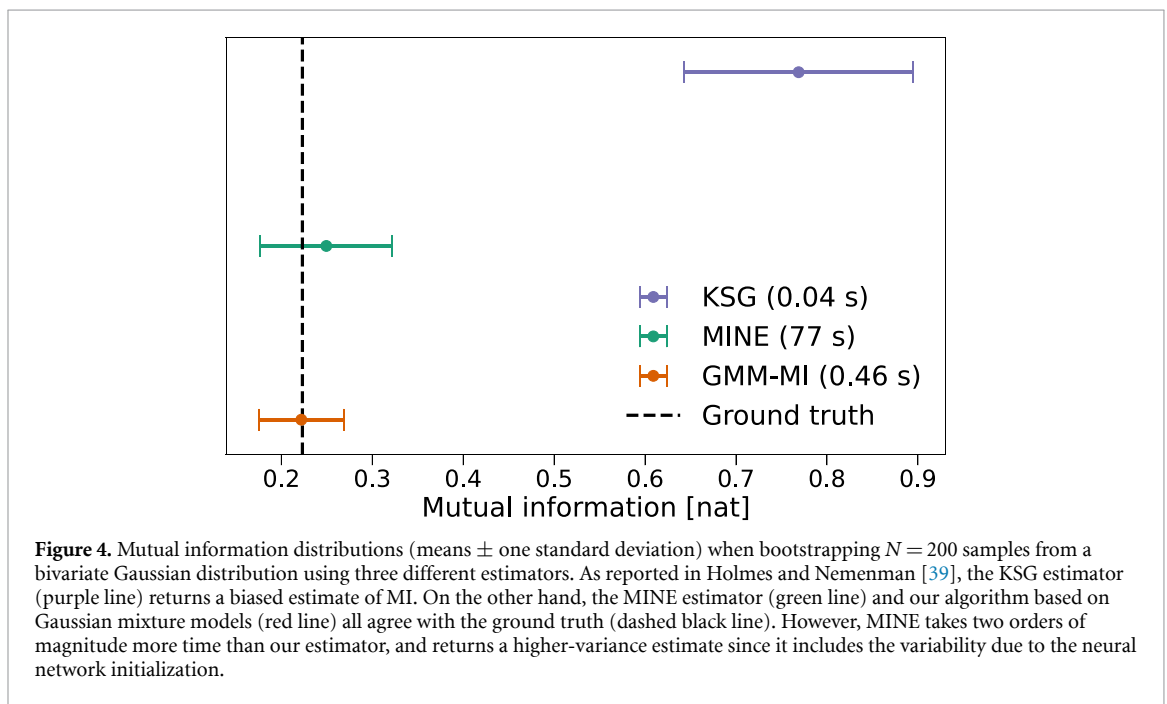


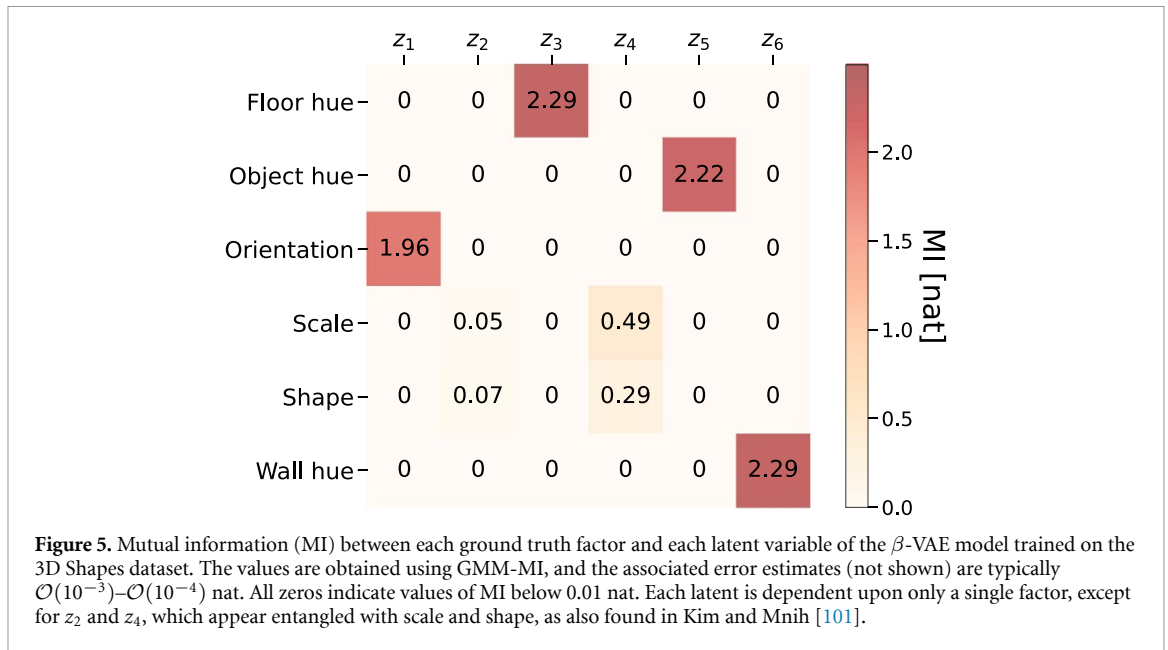
Figure 4. Mutual information distributions (means \pm one standard deviation) when bootstrapping $N = 200$ samples from a bivariate Gaussian distribution using three different estimators. As reported in Holmes and Nemenman [39], the KSG estimator (purple line) returns a biased estimate of MI. On the other hand, the MINE estimator (green line) and our algorithm based on Gaussian mixture models (red line) all agree with the ground truth (dashed black line). However, MINE takes two orders of magnitude more time than our estimator, and returns a higher-variance estimate since it includes the variability due to the neural network initialization.

extra MI. In this section, we address this concern and empirically show that, despite including a bootstrap step, our procedure does not lead to biased estimates of MI.

We consider the same experiment described in Holmes and Nemenman [39], where a single data set of $N = 200$ bivariate Gaussian samples with $\rho = 0.6$ is bootstrapped 20 times. We apply the KSG (with three neighbors, following Holmes and Nemenman [39]) and MINE estimators to each bootstrapped realization, and compare it against our estimator with $n_b = 20$. The results are reported in figure 4. The KSG estimator returns a mean MI biased by a factor of 4, while both MINE and our procedure return an accurate estimate. However, MINE is two orders of magnitude more computationally demanding, and returns an error bar which is larger than with our procedure, since it tends to overestimate the variance, as discussed in section 3.

4. Results

In this section, we apply our estimator to interpret the latent space of representation-learning models trained on three different datasets, ranging from synthetic images to cosmological simulations. We use our MI



estimator to quantify the level of disentanglement of latent variables, and link them to relevant physical parameters. In the following experiments, we consider $k = 3$ folds, $n_{\text{init}} = 5$ different initializations, a log-likelihood threshold on each individual fit of 10^{-5} , $n_b = 100$ bootstrap realizations, $M = 10^5$ MC samples, and a regularization scale of $\omega = 10^{-15}$; as in the experiments described in section 3, we found GMM-MI to be robust to the hyperparameter choices. Obtaining the full distribution of MI with our algorithm typically takes $\mathcal{O}(10)$ s on the datasets we analyze, using the same hardware described in section 3.

4.1. 3D Shapes

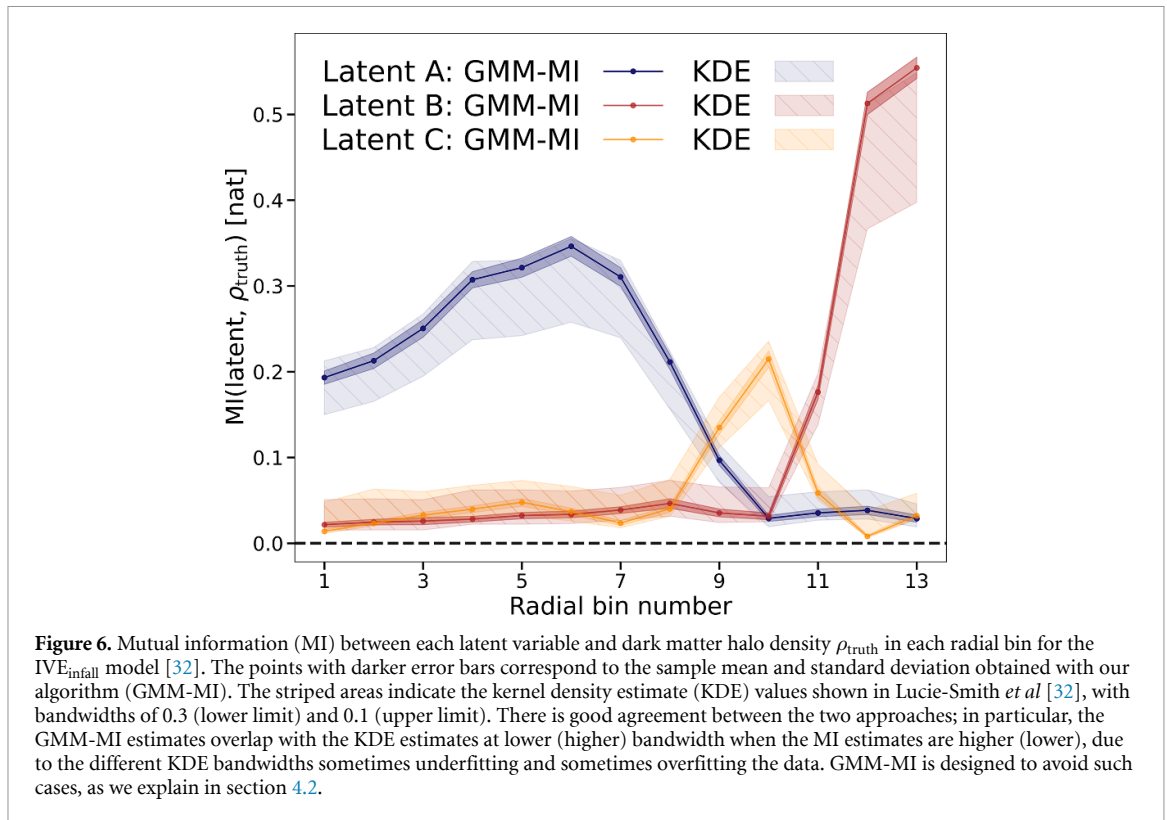
We consider the 3D Shapes dataset [101, 102], which consists of images of various shapes that were generated by the following factors: shape (4 values), scale (8 values), orientation (15 values), floor color (10 values), wall color (10 values), and object color (10 values). Each combination of factors is included in the dataset exactly once, for a total of 480 000 images. We train a β -VAE, as described in section 2.3, on this dataset, using a six-dimensional latent space and setting the value of β using cross-validation.

After training, we encode 10% of the data, which were not used for training or validation, and sample one point from each latent distribution. To interpret what each latent variable z_i has learned about each generative factor of variation f_j , we measure the MI $I(z_i, f_j)$ using equation (3). In figure 5 we report the MI values for all latents and factors using GMM-MI: except for scale and shape, each latent variable carries information about a single factor of variation. The difficulty in disentangling scale and shape was also reported in Kim and Mnih [101]. To assess the level of dependence between latent variables, we calculate $I(z_i, z_j)$: these values are below 10^{-4} nat for all pairs, except for the one carrying information about both scale and shape, i.e. $I(z_2, z_4) = 0.04 \pm 0.01$ nat.

4.2. Dark matter halo density profiles

In the standard model of cosmology, only 5% of our Universe consists of baryonic matter, while the remainder consists of dark matter (25%) and dark energy (70%) [103]. In particular, dark matter only interacts via the gravitational force, and gathers into stable large-scale structures, called ‘halos’, where galaxy formation typically occurs. Given the highly non-linear physical processes taking place during the formation of such structures, a common tool to analyze dark matter halos are cosmological N -body simulations, where particles representing the dark matter are evolved in a box under the influence of gravity [104–106].

Dark matter halos forming within such simulations exhibit a universal spherically-averaged density profile as a function of their radius [107–109]; this universality encompasses a huge range of halo masses and persists within different cosmological models. While the universality of the density profile is still not fully understood, Lucie-Smith *et al* [32, LS22 hereafter] showed that it is possible to train a deep representation learning model to compress raw dark matter halo data into a compact disentangled representation that contains all the information needed to predict dark matter density profiles. Following LS22, we consider 4332 dark matter halos from a single N -body simulation, and encode them using their IVE_{infall} model with three latent variables. The latent representation is used to predict the dark matter halo density profile in 13 different radial bins.



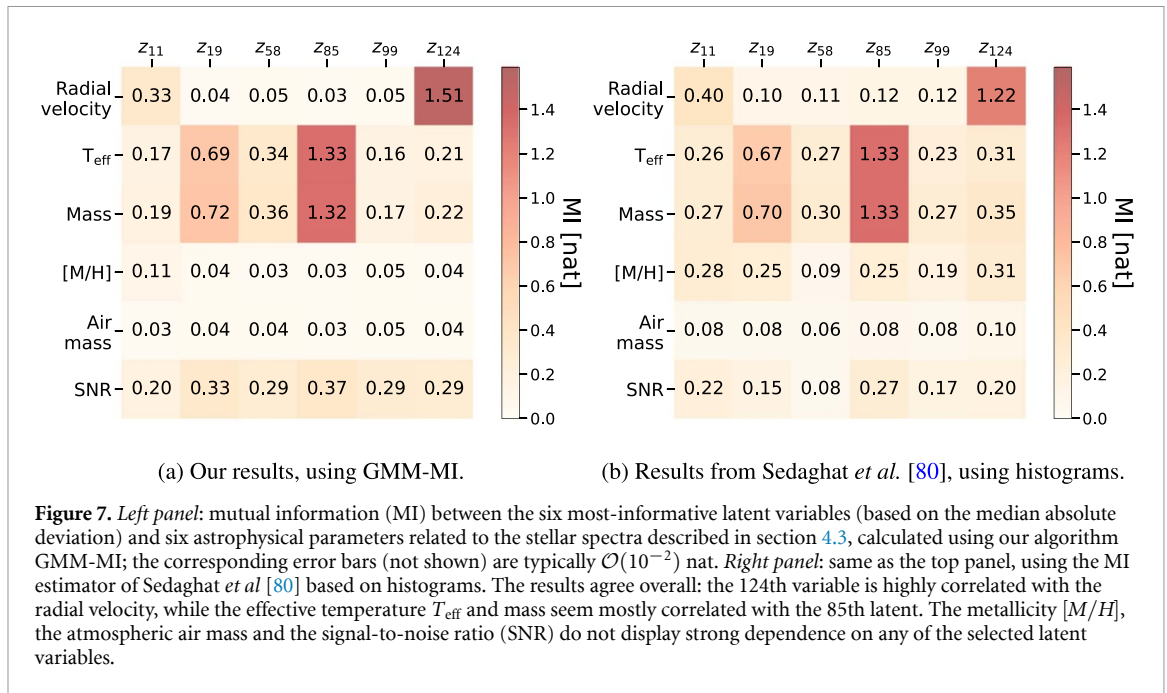
We calculate the MI between the ground-truth halo density in each radial bin and each latent variable, aiming to reproduce the middle panel of figure 4 in LS22, where further details can be found. We show the trend of MI for all radial bins and latent variables in figure 6. We compare the estimates from GMM-MI with those obtained using kernel density estimation (KDE) with different bandwidths, as done in LS22. A major difference between the two approaches is that our bands indicate the error coming from the limited sample size, while their bands represent the sensitivity of the KDE to different bandwidths. The results are in good agreement: in particular, GMM-MI returns estimates closer to the KDE approach with smaller bandwidth when MI is high; in this case, the higher KDE bandwidth value underfits the data. On the other hand, for lower values of MI, GMM-MI yields estimates consistent with the KDE ones at higher bandwidth, since the lower bandwidth overfits the data. This confirms that GMM-MI avoids underfitting and overfitting of the data by design. We also checked that the latent variables of the $\text{IVE}_{\text{infall}}$ model are independent: as in LS22, the MI between each pair of latents is $\mathcal{O}(10^{-2})$ nat.

4.3. Stellar spectra

We consider the model presented in Sedaghat *et al* [80, S21 hereafter], where a β -VAE is trained on about 7000 real unique spectra with a 128-dimensional latent space. These spectra were collected by the High-Accuracy Radial-velocity Planet Searcher instrument ([110, 111]) in the spectral range 378–691 nm, and include mainly stellar spectra, even though Jupiter and asteroid contaminants are present in the dataset. All details about the data, the preprocessing steps and the training procedure can be found in S21.

To select the most informative latent variables, the median absolute deviation (MAD) is calculated for each of them; the rest of the analysis is carried out on the six most informative latents only. We calculate MI between each of these six variables and six known physical factors, all treated as continuous variables. These are the star radial velocity, its effective temperature T_{eff} , its mass, its metallicity $[M/H]$, the atmospheric air mass and the signal-to-noise ratio.

The MI estimates obtained with GMM-MI are shown in the top panel of figure 7: the 124th latent variable shows high dependence on the radial velocity, while the 85th latent appears entangled with both the effective temperature and the mass. The other physical parameters do not show a dependence on a particular latent amongst the ones with the highest MAD, even though in S21 a more complete analysis exploring latent traversals and investigating subsets of data is presented. The bottom panel of the same figure shows the results obtained with the procedure outlined in S21, which uses histograms with a certain number of bins (40



in this case) as density estimators. The trend agrees with our results, even though the particularly high number of bins chosen might overfit the data and overestimate MI (compare e.g. the $[M/H]$ MI estimates), analogously to the KDE results in figure 6. On the other hand, our algorithm provides a robust way to select the hyperparameters, thus avoiding underfitting or overfitting the samples.


5. Conclusions

We presented GMM-MI (pronounced ‘Jimmie’), an efficient and robust algorithm to estimate the MI between two random variables given samples of their joint distribution. Our algorithm uses GMMs to fit the available samples, and returns the full distribution of MI through bootstrapping, thus including the uncertainty on MI due to the finite sample size. GMM-MI is demonstrably accurate, and benefits from the flexibility and computational efficiency of GMMs. Moreover, it can be applied to both discrete and continuous settings, and is robust to the choice of hyperparameters.

We extensively validated GMM-MI on toy datasets for which the ground truth MI is known, showing equal or better performance with respect to established estimators like KSG [56] and MINE [69]; we also tested that GMM-MI respects MI invariance under invertible transformations, is unbiased and returns MI errors that scale as expected with sample size. We demonstrated the application of our estimator to interpret the latent space of three different deep representation-learning models trained on synthetic shape images, large-scale structure in cosmological simulations and real spectra of stars. We calculated both the MI between latent variables and physical factors, and the MI between the latent variables themselves, to investigate their degree of disentanglement, reproducing MI estimates obtained with various techniques, including histograms and kernel density estimators. These results further validate the accuracy of GMM-MI and confirm the power of MI for gaining interpretability of DL models.

We plan to extend our work by improving the density estimation with more flexible tools such as normalizing flows (NFs, [112, 113]), which can be seamlessly integrated into neural network-based settings and can benefit from graphics processing unit acceleration. Moreover, combining NFs with a differentiable numerical integrator would make our estimator amenable to backpropagation, thus allowing its use in the context of MI optimization. We will explore this avenue in future work.

Data availability statement

GMM-MI is publicly available in this GitHub repository (<https://github.com/dpiras/GMM-MI>, also accessible by clicking the icon ) , together with all data and results from the paper.

The data that support the findings of this study are openly available at the following URL/DOI: <https://github.com/dpiras/GMM-MI>.

Acknowledgments

We thank Nima Sedaghat, Martino Romaniello and Vojtech Cvrcek for sharing the stellar spectra model and data. We are also grateful to Justin Alsing for useful discussions about initialization procedures for GMM fitting. DP was supported by the UCL Provost's Strategic Development Fund, and by a Swiss National Science Foundation (SNSF) Professorship Grant (No. 202671). The work of HVP was supported by the Göran Gustafsson Foundation for Research in Natural Sciences and Medicine and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 101018897 CosmicExplorer). HVP and LLS acknowledge the hospitality of the Aspen Center for Physics, which is supported by National Science Foundation Grant PHY-1607611. The participation of HVP and LLS at the Aspen Center for Physics was supported by the Simons Foundation. This study was supported by the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 818085 GMGalaxies. AP is additionally supported by the Royal Society. NG was funded by the UCL Graduate Research Scholarship (GRS) and UCL Overseas Research Scholarship (ORS). This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. This work used computing equipment funded by the Research Capital Investment Fund (RCIF) provided by UKRI, and partially funded by the UCL Cosmoparticle Initiative. This work used facilities provided by the UCL Cosmoparticle Initiative.

Author contributions

DP: formal analysis; investigation; validation; software; writing—original draft preparation, review & editing; visualization. **HVP**: conceptualization; methodology; validation; writing—review & editing; funding acquisition. **AP**: conceptualization; methodology; validation; writing—review & editing; funding acquisition. **LLS**: methodology; validation; resources; writing—review & editing. **NG**: software; validation; writing—review & editing. **BN**: writing—review & editing.

Appendix A. Comparison of convergence criteria for GMMs

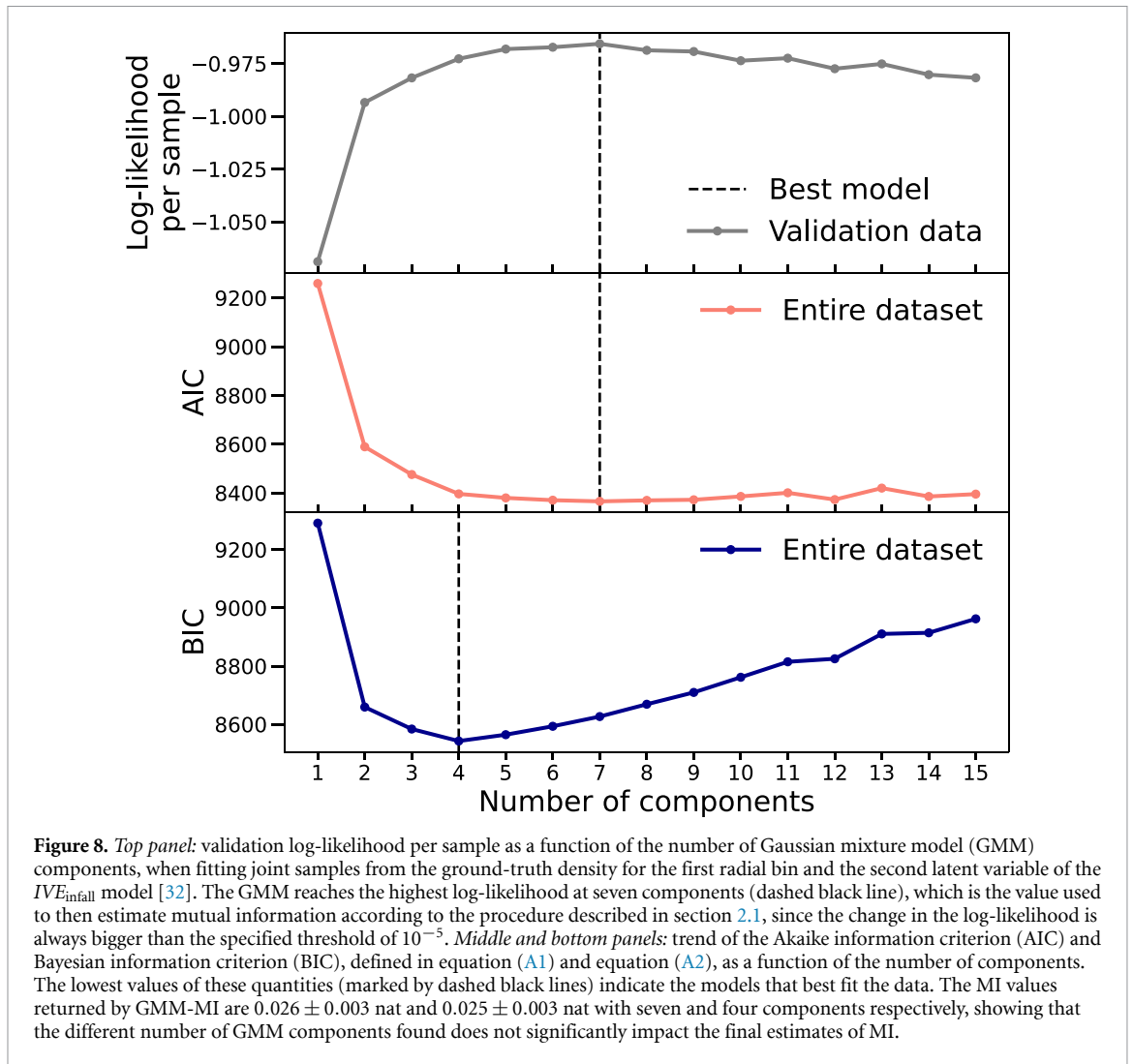
By default, our proposed procedure considers the validation log-likelihood to select the best number of components of the GMM model. Alternatively, one can use the Akaike or the Bayesian information criteria (AIC or BIC, respectively), which are defined as:

$$AIC = 2p - 2\ell, \quad (A1)$$

$$BIC = p \ln N - 2\ell, \quad (A2)$$

where ℓ is the log-likelihood on the training data (the entire dataset in this case), and $p = 6c - 1$ is the total number of GMM parameters, with c the number of GMM components. These criteria include a term for the goodness of fit (ℓ), plus a penalization term to avoid overfitted models. The model with the lowest AIC or BIC should be chosen, and ample discussions are available as to which criterion works best [88, 114, 115].

As an example, we compare the trend of the validation log-likelihood, AIC and BIC in the context of the dark matter halo density profiles (section 4.2) when considering the second latent variable (latent 'B') and the density in the first radial bin. We increase the number of GMM components from 1 to 15, and report the results in figure 8. The validation log-likelihood reaches its maximum at seven components, and then starts to slowly decrease. The AIC also prefers seven components, while the BIC is in favor of fewer components (four). This is not surprising, since the penalization term is stronger in the BIC case, given the high number of samples. All three metrics considered are efficient to compute, and since the MI estimates returned by GMM-MI with seven and eight components are 0.026 ± 0.003 nat and 0.025 ± 0.003 , respectively, we conclude that our approach is robust to the choice of the metric used to select the number of GMM components.



Appendix B. Derivation of the MI between a continuous and a categorical variable

While equation (3) is not novel, in this appendix we detail the assumptions made in its derivation. We first rewrite equation (1) as:

$$I(X, Y) = \int_{\mathcal{X} \times \mathcal{Y}} p_{(X|Y)}(x|y) p_Y(y) \ln \frac{p_{(X|Y)}(x|y)}{p_X(x)} dx dy. \tag{B1}$$

Then, we assume a generalized probability density function for the categorical variable F over \mathcal{F} :

$$p_F(f) = \sum_{i=1}^v p_F(f=f_i) \delta(f-f_i) = \frac{1}{v} \sum_{i=1}^v \delta(f-f_i), \tag{B2}$$

where δ is the Dirac delta function, and in the last step we assumed that F can take the values $f_{1,v}$ with equal probability. Combining the last two equations, we obtain:

$$\begin{aligned} I(X, F) &= \int_{\mathcal{X} \times \mathcal{F}} dx df p_{(X|F)}(x|f) p_F(f) \ln \frac{p_{(X|F)}(x|f)}{p_X(x)} \\ &= \frac{1}{v} \sum_{i=1}^v \int_{\mathcal{X}} dx p_{(X|F)}(x|f_i) \left[\ln p_{(X|F)}(x|f_i) - \ln \frac{1}{v} \sum_{j=1}^v p_{(X|F)}(x|f_j) \right], \end{aligned} \tag{B3}$$

as reported in equation (3).

Appendix C. Ground truth values of MI

We report the true values of MI for the bivariate distributions considered in section 3. These values can be obtained via direct integration of equation (1), and depend on a real-valued parameter $\alpha > 0$. For the gamma-exponential distribution [54, 56, 99, 100] as defined in equation (13):

$$I(X, Y) = \psi(\alpha + 1) - \ln \alpha, \quad (C1)$$

where ψ is the digamma function, defined as:

$$\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}.$$

For the ordered Weinman exponential distribution [54, 56, 99, 100] as defined in equation (14):

$$I(X, Y) = \begin{cases} \ln\left(\frac{1-2\alpha}{2\alpha}\right) + \psi\left(\frac{1}{1-2\alpha}\right) - \psi(1) & \alpha < \frac{1}{2} \\ -\psi(1) & \alpha = \frac{1}{2} \\ \ln\left(\frac{2\alpha-1}{2\alpha}\right) + \psi\left(\frac{2\alpha}{2\alpha-1}\right) - \psi(1) & \alpha > \frac{1}{2} \end{cases}. \quad (C3)$$

ORCID iD

Davide Piras  <https://orcid.org/0000-0002-9836-2661>

References

- [1] Raghu M and Schmidt E 2020 (arXiv:2003.11755 [cs.LG])
- [2] Cybenko G 1989 *Math. Control, Signals Syst. (MCSS)* **2** 303
- [3] Hornik K, Stinchcombe M and White H 1989 *Neural Netw.* **2** 359
- [4] Hornik K 1991 *Neural Netw.* **4** 251
- [5] Molnar C 2022 *Interpretable Machine Learning* 2nd edn (Victoria: Leanpub)
- [6] Zeiler M D and Fergus R 2014 *Computer Vision – Eccv 2014* ed D Fleet, T Pajdla, B Schiele and T Tuytelaars (Cham: Springer) pp 818–33
- [7] Simonyan K, Vedaldi A and Zisserman A 2014 *Workshop at Int. Conf. on Learning Representations*
- [8] Zhou B, Khosla A, Lapedriza A, Oliva A and Torralba A 2016 *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA: IEEE Computer Society) pp 2921–9
- [9] Ribeiro M T, Singh S and Guestrin C 2016 *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '16)* (New York: Association for Computing Machinery) pp 1135–44
- [10] Selvaraju R R, Cogswell M, Das A, Vedantam R, Parikh D and Batra D 2017 *2017 IEEE Int. Conf. on Computer Vision (ICCV)* pp 618–26
- [11] Shrikumar A, Greenside P and Kundaje A 2017 *Proc. 34th Int. Conf. on Machine Learning (ICML' 17)* vol 70 (JMLR.org) pp 3145–53
- [12] Lundberg S M and Lee S-I 2017 *Proc. 31st Int. Conf. on Neural Information Processing Systems (NIPS'17)* (Red Hook, New York: Curran Associates Inc.) pp 4768–77
- [13] Chattopadhyay A, Sarkar A, Howlader P and Balasubramanian V N 2018 *2018 IEEE Winter Conf. on Applications of Computer Vision (WACV)* pp 839–47
- [14] Li X, Xiong H, Li X, Wu X, Zhang X, Liu J, Bian J and Dou D 2021 (arXiv:2103.10689 [cs.LG])
- [15] Linardatos P, Papastefanopoulos V and Kotsiantis S 2021 *Entropy* **23** 18
- [16] Schmidhuber J 1992 *Neural Comput.* **4** 863
- [17] Bengio Y, Courville A and Vincent P 2013 *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 1798
- [18] Louizos C, Swersky K, Li Y, Welling M and Zemel R S 2016 *4th Int. Conf. on Learning Representations (ICLR 2016)* (San Juan, Puerto Rico, 2–4 May 2016) (*Conf. Track Proc.*), ed Y Bengio and Y LeCun
- [19] Chen X, Duan Y, Houthoofd R, Schulman J, Sutskever I and Abbeel P 2016 *Advances in Neural Information Processing Systems* vol 29
- [20] Lample G, Zeghidour N, Usunier N, Bordes A, Denoyer L and Ranzato M 2017 *Proc. 31st Int. Conf. on Neural Information Processing Systems (NIPS'17)* (Red Hook, New York: Curran Associates Inc.) pp 5969–78
- [21] Higgins I, Matthey L, Pal A, Burgess C P, Glorot X, Botvinick M M, Mohamed S and Lerchner A 2017 *ICLR*
- [22] Jha A H, Anand S, Singh M and Veeravasaru V 2018 *Computer Vision – Eccv 2018*, ed V Ferrari, M Hebert, C Sminchisescu and Y Weiss (Cham: Springer) pp 829–45
- [23] Locatello F, Bauer S, Lucic M, Raetsch G, Gelly S, Schölkopf B and Bachem O 2019 *Proc. 36th Int. Conf. on Machine Learning (Proc. Machine Learning Research)* vol 97, ed K Chaudhuri and R Salakhutdinov (PMLR) pp 4114–24
- [24] Lezama J 2019 *7th Int. Conf. on Learning Representations (ICLR 2019)* (New Orleans, LA, USA, 6–9 May 2019) (OpenReview.net)
- [25] Pandey B and Sarkar S 2017 *Mon. Not. R. Astron. Soc.* **467** L6
- [26] Sarkar S and Pandey B 2020 *Mon. Not. R. Astron. Soc.* **497** 4077
- [27] Bhattacharjee S, Pandey B and Sarkar S 2020 *J. Cosmol. Astropart. Phys.* **2020** 039
- [28] Upham R E, Brown M L and Whittaker L 2021 *Mon. Not. R. Astron. Soc.* **503** 1999
- [29] Malz A I, Lanusse F, Crenshaw J F and Graham M L 2021 (arXiv:2004.05016 [astro-ph.IM])
- [30] Sarkar S, Pandey B and Bhattacharjee S 2021 *Mon. Not. R. Astron. Soc.* **501** 994
- [31] Jeffrey N, Alsing J and Lanusse F 2021 *Mon. Not. R. Astron. Soc.* **501** 954

- [32] Lucie-Smith L, Peiris H V, Pontzen A, Nord B, Thiyaalingam J and Piras D 2022 *Phys. Rev. D* **105** 103533
- [33] Sarkar S, Pandey B and Das A 2022 *J. Cosmol. Astropart. Phys.* **2022** 024
- [34] Fairhall A, Shea-Brown E and Barreiro A 2012 *Curr. Opin. Neurobiol.* **22** 653
- [35] Charzyńska A and Gambin A 2016 *Entropy* **18** 13
- [36] Tkačik G and Bialek W 2016 *Annu. Rev. Condens. Matter Phys.* **7** 89
- [37] Levchenko A and Nemenman I 2014 *Curr. Opin. Neurobiol.* **28** 156
- [38] von Wegner F, Laufs H and Tagliazucchi E 2018 *Phys. Rev. E* **97** 022415
- [39] Holmes C M and Nemenman I 2019 *Phys. Rev. E* **100** 022404
- [40] Uda S 2020 *Biophys. Rev.* **12** 377
- [41] Wicks R T, Chapman S C and Dendy R O 2007 *Phys. Rev. E* **75** 051125
- [42] Dunleavy A J, Wiesner K and Royall C P 2012 *Phys. Rev. E* **86** 041505
- [43] Runge J 2015 *Phys. Rev. E* **92** 062829
- [44] Myers A, Munch E and Khasawneh F A 2019 *Phys. Rev. E* **100** 022314
- [45] Svenkeson A and West B J 2019 *Phys. Rev. E* **100** 022119
- [46] Diego D, Haaga K A and Hannisdal B 2019 *Phys. Rev. E* **99** 042212
- [47] Jiang P and Kumar P 2019 *Phys. Rev. E* **99** 012306
- [48] Jia Z, Lin Y, Liu Y, Jiao Z and Wang J 2020 *Phys. Rev. E* **101** 062113
- [49] Paninski L 2003 *Neural Comput.* **15** 1191
- [50] Vergara J R and Estévez P A 2015 (arXiv:1509.07577 [cs.LG])
- [51] Cover T M and Thomas J A 2006 *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)* (New York: Wiley)
- [52] Fraser A M and Swinney H L 1986 *Phys. Rev. A* **33** 1134
- [53] Moon Y-I, Rajagopalan B and Lall U 1995 *Phys. Rev. E* **52** 2318
- [54] Darbellay G and Vajda I 1999 *IEEE Trans. Inf. Theory* **45** 1315
- [55] Kwak N and Choi C-H 2002 *IEEE Trans. Pattern Anal. Mach. Intell.* **24** 1667
- [56] Kraskov A, Stögbauer H and Grassberger P 2004 *Phys. Rev. E* **69** 066138
- [57] Suzuki T, Sugiyama M, Sese J and Kanamori T 2008 *Proc. Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery at ECML/PKDD 2008 (Proc. Machine Learning Research)* vol 4, ed Y Saeyns, H Liu, I Inza, L Wehenkel and Y V d Pee (Antwerp, Belgium: PMLR) pp 5–20
- [58] Saxe A M, Bansal Y, Dapello J, Advani M, Kolchinsky A, Tracey B D and Cox D D 2019 *J. Stat. Mech.* **2019** 124020
- [59] Pichler G, Colombo P, Boudiaf M, Koliander G and Piantanida P 2022 (arXiv:2202.06618 [cs.LG])
- [60] Kozachenko L F and Leonenko N N 1987 *Probl. Inf. Transm.* **23** 95
- [61] Gao S, Ver Steeg G and Galstyan A 2014 (arXiv:1411.2003 [cs.IT])
- [62] Hutter M 2002 *Advances in Neural Information Processing Systems* vol 14, ed T G Dietterich, S Becker and Z Ghahramani (Cambridge, MA: MIT Press) pp 399–406
- [63] Hutter M and Zaffalon M 2005 *Comput. Stat. Data Anal.* **48** 633
- [64] Archer E, Park I M and Pillow J W 2013 *Entropy* **15** 1738
- [65] Tishby N and Zaslavsky N 2015 *2015 IEEE Information Theory Workshop (ITW)* (IEEE) pp 1–5
- [66] Alemi A A, Fischer I, Dillon J V and Murphy K 2016 arXiv:1612.00410 [cs.LG]
- [67] Brakel P and Bengio Y 2017 (arXiv:1710.05050 [stat.ML])
- [68] Kolchinsky A, Tracey B D and Wolpert D H 2019 *Entropy* **21** 1181
- [69] Belghazi M I, Baratin A, Rajeshwar S, Ozair S, Bengio Y, Courville A and Hjelm D 2018 *Proc. 35th Int. Conf. on Machine Learning (Proc. Machine Learning Research)* vol 80, ed J Dy and A Krause (PMLR) pp 531–40
- [70] van den Oord A, Li Y and Vinyals O 2018 *CoRR* arXiv:1807.03748
- [71] Moyer D, Gao S, Brekelmans R, Galstyan A and Ver Steeg G 2018 *Advances in Neural Information Processing Systems* vol 31, ed S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi and R Garnett (Curran Associates, Inc.)
- [72] Poole B, Ozair S, van den Oord A, Alemi A A and Tucker G 2019 arXiv:1905.06922 [cs.LG]
- [73] Peng X B, Kanazawa A, Toyer S, Abbeel P and Levine S 2019 *7th Int. Conf. on Learning Representations (ICLR 2019) (New Orleans, LA, USA, 6–9 May 2019)* (OpenReview.net)
- [74] Hjelm R D, Fedorov A, Lavoie-Marchildon S, Grewal K, Bachman P, Trischler A and Bengio Y 2019 *7th Int. Conf. on Learning Representations (ICLR 2019) (New Orleans, LA, USA, 6–9 May 2019)* (OpenReview.net)
- [75] Song J and Ermon S 2020 *Int. Conf. on Learning Representations*
- [76] Gökmen D E, Ringel Z, Huber S D and Koch-Janusz M 2021 *Phys. Rev. E* **104** 064106
- [77] Kullback S and Leibler R A 1951 *Ann. Math. Statist.* **22** 79–86
- [78] Donsker M and Varadhan S 1983 *Commun. Pure Appl. Math.* **36** 183
- [79] Chen R T Q, Li X, Grosse R B and Duvenaud D K 2018 *Advances in Neural Information Processing Systems* vol 31, ed S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi and R Garnett (Curran Associates, Inc.)
- [80] Sedaghat N, Romaniello M, Carrick J E and Pineau F-X 2021 *Mon. Not. R. Astron. Soc.* **501** 6026
- [81] Ait Kerroum M, Hammouch A and Aboutajdine D 2010 *Pattern Recognit. Lett.* **31** 1168
- [82] Eirola E, Lendasse A and Karhunen J 2014 *2014 Int. Joint Conf. on Neural Networks (IJCNN)* pp 1606–13
- [83] Lan T, Erdogmus D, Ozertem U and Huang Y 2006 *The 2006 IEEE Int. Joint Conf. on Neural Network Proc.* pp 5034–9
- [84] Leiva-Murillo J M and Artés-Rodríguez A 2004 *Independent Component Analysis and Blind Signal Separation*, ed C G Puntonet and A Prieto (Berlin: Springer) pp 271–8
- [85] Nilsson M, Gustafson H, Vang Andersen S and Kleijn W B 2002 *2002 IEEE Int. Conf. on Acoustics, Speech and Signal Processing* vol 1 p I-525–I-528
- [86] Polo F M and Vicente R 2022 *Neural Comput. Appl.* 1–13
- [87] Ueda N, Nakano R, Ghahramani Z and Hinton G 1998 *Neural Networks for Signal Processing VIII. Proc. 1998 IEEE Signal Processing Society Workshop (Cat. No.98TH8378)* pp 274–83
- [88] Bovy J, Hogg D W and Roweis S T 2011 *Ann. Appl. Stat.* **5** 1657
- [89] Shireman E M, Steinley D and Brusco M J 2016 *Multivariate Behav. Res.* **51** 466
- [90] Baudry J-P and Celeux G 2015 *Stat. Comput.* **25** 713
- [91] Melchior P and Goulding A D 2018 *Astron. Comput.* **25** 183
- [92] Dempster A P, Laird N M and Rubin D B 1977 *J. R. Stat. Soc. B* **39** 1–22

- [93] Lloyd S 1982 *IEEE Trans. Inf. Theory* **28** 129
- [94] Arthur D and Vassilvitskii S 2007 *Proc. of the Eighteenth Annual ACM-Siam Symp. on Discrete Algorithms (Soda '07)* (Philadelphia, PA: Society for Industrial and Applied Mathematics) pp 1027–35
- [95] Akaike H 1974 *IEEE Trans. Autom. Control* **19** 716
- [96] Schwarz G 1978 *Ann. Stat.* **6** 461
- [97] Ross B C 2014 *PLoS One* **9** 1
- [98] Kingma D P and Welling M 2014 *2nd Int. Conf. on Learning Representations (ICLR 2014)* (Banff, AB, Canada 14–16 April 2014) (*Conf. Track Proc.*)
- [99] Darbellay G and Vajda I 2000 *IEEE Trans. Inf. Theory* **46** 709
- [100] Haeri M A and Ebadzadeh M M 2014 *Fuzzy Optim. Decis. Mak.* **13** 287
- [101] Kim H and Mnih A 2018 *Proc. 35th Int. Conf. on Machine Learning (Proc. Machine Learning Research)* vol 80, ed J Dy and A Krause (PMLR) pp 2649–58
- [102] Burgess C and Kim H 2018 3DShapesDataset (available at:<https://github.com/deepmind/3d-shapes>)
- [103] Dodelson S 2003 *Modern Cosmology* (New York: Academic)
- [104] Navarro J F, Frenk C S and White S D M 1996 *Astrophys. J.* **462** 563
- [105] Tormen G 1997 *Mon. Not. R. Astron. Soc.* **290** 411
- [106] Jenkins A, Frenk C S, Pearce F R, Thomas P A, Colberg J M, White S D M, Couchman H M P, Peacock J A, Efstathiou G and Nelson A H 1998 *Astrophys. J.* **499** 20
- [107] Navarro J F, Frenk C S and White S D M 1997 *Astrophys. J.* **490** 493
- [108] Huss A, Jain B and Steinmetz M 1999 *Astrophys. J.* **517** 64
- [109] Wang J and White S D M 2009 *Mon. Not. R. Astron. Soc.* **396** 709
- [110] Pepe F et al 2002 *The Messenger* **110** 9
- [111] Mayor M et al 2003 *The Messenger* **114** 20
- [112] Dinh L, Krueger D and Bengio Y 2014 (arXiv:[1410.8516](https://arxiv.org/abs/1410.8516) [cs.LG])
- [113] Rezende D J and Mohamed S 2015 *Proc. 32nd Int. Conf. on Int. Conf. on Machine Learning (ICML'15)* vol 37 (JMLR.org) pp 1530–8
- [114] Burnham K P and Anderson D R 2004 *Sociol. Methods Res.* **33** 261
- [115] Holoiien T W-S, Marshall P J and Wechsler R H 2017 *Astron. J.* **153** 249