Neural Joint Entropy Estimation

Yuval Shalev, Amichai Painsky, and Irad Ben-Gal

***Abstract*— Estimating the entropy of a discrete random variable is a fundamental problem in information theory and related fields. This problem has many applications in various domains, including machine learning, statistics, and data compression. Over the years, a variety of estimation schemes have been suggested. However, despite significant progress, most methods still struggle when the sample is small, compared to the variable's alphabet size. In this work, we introduce a practical solution to this problem, which extends the work of McAllester and Statos. The proposed scheme uses the generalization abilities of cross-entropy estimation in deep neural networks (DNNs) to introduce improved entropy estimation accuracy. Furthermore, we introduce a family of estimators for related information-theoretic measures, such as conditional entropy and mutual information (MI). We show that these estimators are strongly consistent and demonstrate their performance in a variety of use cases. First, we consider large alphabet entropy estimation. Then, we extend the scope to MI estimation. Next, we apply the proposed scheme to conditional MI estimation, as we focus on independence testing tasks. Finally, we study a transfer entropy (TE) estimation problem. The proposed estimators demonstrate improved performance compared to existing methods in all of these setups.**

***Index Terms*— Cross-entropy, joint entropy, mutual information (MI), neural networks, transfer entropy (TE).**

I. Introduction

ENTROPY is one of the basic building blocks of information theory [1]. It quantifies the minimum average number of bits required to represent an event that follows a given probability distribution rule. Many important information-theoretic measures such as mutual information (MI) and conditional MI (CMI) include marginal, conditional, and joint entropies. These measures have many applications in machine learning, such as feature selection [2], [3], representation learning [4], [5], and analyses of the learning mechanism [6], [7].

One of the first entropy estimation methods is the classic plug-in scheme. In this scheme, an empirical distribution replaces the true (unknown) probability rule, and the corresponding empirical entropy is the estimated entropy. In addition to its simplicity, the plug-in scheme enjoys several favorable properties (consistent, asymptotically unbiased, and others (see [8] and references therein). Unfortunately, it does not scale well as the dimension of the problem increases [9]. A variety of parametric and nonparametric methods have been proposed to improve entropy estimation, such as in [9], [10], and [11]. Recently, a neural network-based method was proposed to estimate entropy by minimizing the cross-entropy (CE) loss [12] as an upper bound of the entropy. The CE measures the average number of bits required to represent an event that is generated from a probability distribution P by a different probability distribution Q. CE achieves its minimum when $P = Q$. Thus, minimizing CE implies searching for a Q that is as similar as possible in a log-loss [13], [14] sense to P. This approach has several advantages. First, it uses the generalization power of neural networks and their universality [15], [16], [17]. Second, CE is less prone to negative bias and high variance in large entropy values [12]. However, this approach has certain limitations. First, it requires prior assumptions on the true underlying distribution, as discussed in Section III. Second, the statistical properties of this CE estimator are currently unexplored. Therefore, the existence of a neural network-based estimator that can provide an accurate estimation of entropy is not guaranteed.

These challenges in entropy estimation are also related to other information-theoretic measures. For example, one of the most common MI estimation schemes is the K-nearest neighbor (KNN) estimator [18]. This estimator was shown to introduce a significant negative bias in setups with high dependencies between the variables, resulting in large MI values [19]. Neural-network-based approaches have been recently proposed to overcome this problem using variational bound optimization [19], [20], [21]. Although a significant improvement in the MI estimation has been achieved, the results are not yet satisfying and suffer from theoretical limitations that are primarily manifested in large MI values [12], [20]. There is also a large body of work on fundamental estimation bounds for different information-theoretic measures (see [9], [22] and related work).

In this article, we address the inherent estimation challenges discussed above. The proposed estimation scheme focuses on joint entropy estimation. This problem is similar to the standard entropy estimation problem as any discrete univariate random vector may be represented, for example, as a binary multivariate vector. In particular, we combine the chain rule with the CE loss minimization procedure using neural networks to obtain a more accurate joint entropy estimation. We denote this estimation procedure as the neural joint entropy estimator (*NJEE*). We study the properties of *NJEE* and show that it is strongly consistent. In a similar manner, we obtain the conditional *NJEE* (*C-NJEE*), as an estimator for the joint conditional entropy between two or more multivariate variables.

Having these two estimators, we can use the difference between the marginal entropy of one random variable and

Manuscript received 22 December 2020; revised 11 September 2021 and 19 April 2022; accepted 1 September 2022. This work was supported in part by the Digital Living 2030 Grant, in part by the Koret Foundation under the Grant for Smart Cities and Digital Living, and in part by the Israel Science Foundation under Grant 963/21. *(Corresponding author: Yuval Shalev.)*

The authors are with the Department of Industrial Engineering, Tel-Aviv University, Ramat-Aviv 6997801, Israel (e-mail: yuvalshalev@mail.tau.ac.il).

Digital Object Identifier 10.1109/TNNLS.2022.3204919

the conditional entropy of this variable given another to estimate the MI among these variables. Adding a second conditioning variable results in the CMI estimator. Additionally, we apply the proposed scheme to transfer entropy (TE) estimation. Given two time series, the TE is defined as the CMI between the "past" of the first series and the "future" of the second series given its "past." TE is used to explore the information flow and causality among time-dependent data in neuroscience [23], [24], finance [25], [26], process control [27], [28], and many other applications. We show that using an autoregressive neural network model, such as a recurrent neural network, *C-NJEE* can be used for efficient TE estimation.

The advantages of the estimators proposed in this article are demonstrated in various use cases. First, we study the entropy estimation of a discrete random variable with a large alphabet size. Applying *NJEE* to this problem, we outperform existing methods when the sample size is much smaller than the alphabet size. Further, we focus on MI estimation between two multivariate variables. A commonly used toy problem is used for this task. The performance of the proposed MI estimator demonstrates improved results in terms of lower bias and variance, compared to existing methods. This result is specifically manifested in larger values of MI. Next, we demonstrate the performance of the suggested CMI estimator, as we focus on conditional independence tests. We study a real protein dataset where dependencies among the variables (protein elements) are known. Here to, the proposed estimation scheme demonstrates better results than existing methods. Finally, the CMI estimator is applied to a TE estimation task. Specifically, we study a real financial dataset of stock index prices and show that the *C-NJEE*-based estimation provides additional insights on the information flow between the time series that are not discovered by the other methods. These insights are in line with domain knowledge and the world financial timeline.

To summarize, the contributions of this article are threefold. First, we extend the work of [12] and introduce strongly consistent estimators for joint entropy and conditional joint entropy. The proposed estimators, *NJEE* and *C-NJEE*, are based on minimization of the CE loss while applying the entropy chain rule property. Second, we apply these estimators to obtain estimators for related measures such as MI, CMI, and TE. Third, we propose a practical implementation scheme of these estimators that demonstrates better performance than existing methods on various tasks and datasets.

The remainder of this article is organized as follows. Related works on entropy, MI, CMI, and TE estimation are discussed in Section II. In Section III, definitions and related mathematical overview are given to support the scheme and ideas proposed in this article. The primary results are shown in Section IV. An empirical study of various tasks and comparisons with different benchmark methods are provided in Section V. We conclude this article in Section VI.

II. Related Work

Estimating information-theoretic measures is a well-studied problem. We refer the reader to [9], [11], [18], [20], [29], [30] for a comprehensive review of these measures. The following literature review focuses on estimators that are relevant for this work.

A. Entropy Estimation in Large Alphabet

As mentioned in Section I, the simplest method for estimating the entropy of a discrete random variable is the so-called plug-in estimator [1]. The Miller-Madow (MM) estimator [31] adds a bias correction to the plug-in estimator. This correction depends on the ratio between the number of symbols from the alphabet that appear at least once in the sample and the sample size.

More recently, the Chao–Shen (CS) estimator [10] was proposed to estimate the entropy of species in a community (in this biological context, the entropy is called the diversity index), where the number of species (alphabet size) is large and unknown. This estimator is based on the Horvitz–Thompson estimator for population size and the Good-Turing estimator for the probability of unseen events. In [9], an entropy estimator is obtained using a polynomial approximation for the terms in the entropy sum that involve small probabilities with respect to $\log k$, where k is the alphabet size. For larger probabilities, an unbiased plug-in estimator is used that is similar to the MM estimator. Thus, improved results are demonstrated on simulated data of discrete random variables with large alphabet sizes where many symbols have relatively low probability.

B. MI and CMI Estimation

In this section, we provide a brief review of MI and MI-related estimators for both discrete and continuous variables.

The KNN-based *KSG* estimator [18], uses KNN-based density estimation over a shared space of the marginal and conditional entropy. Using the connection between MI and entropy (see Section III-B), the entropies' bias terms are subtracted to provide a more accurate MI estimation. This estimator is also shown to be consistent. However, it underestimates the MI when the true MI is large [19], [32]. An intuitive explanation is that the *KSG* estimator approximates the probability density in a k nearest neighbors ball or max-norm rectangle, under the assumption that in this local neighborhood, the density is uniform. If strong correlations exist, the density in the shared space will be more singular, hence the uniform density assumption becomes problematic [32]. Another drawback of the *KSG* estimator is that there is no clear way to choose the most appropriate value of k, since this is an unsupervised estimation procedure [33].

The recent advances in deep learning motivated various researchers to address the dimensionality problem by estimating the MI with neural networks. This is usually obtained by finding variational lower bound for the MI (typically, a differentiable function that is called a *critic*, which its supremum is the MI). These functions are approximated by neural networks to maximize the lower bound [19], [20], [21]. These methods yield improved results compared to the KNN-based estimators. However, they are quite limited in cases

where the MI increases, since their estimation complexity increases exponentially with the number of samples [12], [20].

A different MI estimation approach, which also utilizes neural network CE minimization, is proposed in [12]. There, a MI estimate is obtained by subtracting the estimated conditional entropy from the estimated marginal entropy. This approach motivates our proposed estimation scheme as discussed in further detail in Section III-C. A similar approach for MI estimation using the softmax function (e.g., as the output layer in a neural network), is suggested in [34]. However, this scheme is limited to the case where the input variable is multivariate, while the target variable is univariate.

An additional important approach utilizes an embedding of the data to reproducing kernel Hilbert spaces (RKHS) for estimating the Rényi's entropy. Rényi's quadratic entropy is the log function of the statistical mean embedding of the projected data in RKHS [35]. Let x and y be samples from two Borel measures P and Q. Since the embedding is injective, transforming the respective samples to a RKHS (with a corresponding kernel $G(\cdot,\cdot)$) implies that $E(G(x,.)) = E(G(y,.))$ if $P = Q$ [36], [37], [38]. This shows that one does not need to explicitly define functional approximators to estimate information descriptors in this framework. To obtain the estimated value, all is needed is to compute the mean value of the projected samples using the kernel trick and apply a log function. Note that this can be extended to any value α of Rényi's entropy, which includes Shannon entropy for $\alpha \to 1$. In a more recent result, Giraldo *et al.* [39] utilizes the eigenvalues of the normalized Gram matrix to estimate MI, which is more flexible than the statistical embedding.

Additional approaches using RKHS were recently proposed. The kernel KL divergence estimator (*KKLE*) [40] is a nonparametric method which is suggested to reduce the optimization problem of searching a tight lower bound to the MI to a convex problem. It is also shown that this estimator is strongly consistent. However, as noted by Ahuja [40], this approach still suffers from a large estimation error in cases where the dimensions of the variables increase. Sreekar *et al.* [41] proposed to optimize the variational lower bound, such as those described earlier, while limiting the search for functions in the RKHS, thus controlling the complexity of the hypothesis space. This regularization is applied by an automated spectral kernel learning (*ASKL*) to learn the appropriate kernel. It is shown that using *ASKL*, MI estimations with lower bias and variance are obtained, specifically in larger values of MI.

As for the CMI estimation problem, a classifier based conditional MI (*CCMI*) is proposed in [42]. A two-sample classifier is used to distinguish between samples from the joint distribution and samples from the marginal distribution. Combining conditional generative models [e.g., conditional generative adversarial networks (CGANs) or conditional variational autoencoders (CVAEs)], an estimator for the CMI was developed. This approach introduced a significant improvement over other recently proposed methods.

C. TE Estimation

The TE is defined as a form of CMI between time series. Specifically, $\mathrm{TE}(Y_{\text{future}}; X_{\text{past}}) = \mathrm{CMI}(Y_{\text{future}}, X_{\text{past}}|Y_{\text{past}})$ (see a formal definition of TE in Section III-B). There are two primary approaches for TE estimation. The first approach considers every variable in every timestamp as a separate variable, and uses any MI or CMI estimator to estimate the TE [43], [44], [45]. The second approach applies a sequential model that considers the time dependencies among different time lags to extract an estimator for the TE and its related measures [46], [47]. As a representative of the first approach, a recently proposed estimator [45] called the intrinsic transfer entropy neural estimator (*ITENE*) applies a neural network two-sample classifier to estimate the TE. Using the second approach, the context tree weighting (CTW) algorithm [48] is utilized in [46] for directed information estimation (a closely related measure to TE [49]). Both works investigate a financial time series of index prices to evaluate their estimators. We use the same dataset to evaluate the proposed method.

III. Background

A. Notations

The following notations are used throughout this article. A univariate discrete random variable is denoted by an upper-case letter (e.g., X), that obtains values x from the alphabet $\mathcal{A}_x = \{1, \ldots, a_x\}$. A multivariate variable with dimensions d_x is denoted by an underline, (e.g., $\underline{X}$), where its values are denoted by underlined lower-case letter $\underline{x}$. The mth component of $\underline{X}$ is denoted as X_m, which obtains values x_m from the alphabet $\mathcal{A}_{x_m} = \{1, \ldots, a_{x_m}\}$ which can be different for different values of m. The vector of the first k components of $\underline{X}$ is denoted by $\underline{X}^k$.

We denote $\widehat{H}_n(\underline{X})$ as the estimator of $\underline{X}$'s entropy given a sample $S = \{\ldots\}_{i=1}^n$, where it is implied from the text that S is a collection of n samples of $\underline{X}$. This notation holds for other estimators as well. For example, $\widehat{I}_n(\underline{X}; \underline{Y}|\underline{Z})$ is an estimator of the CMI between $\underline{X}$ and $\underline{Y}$ given $\underline{Z}$, from a collection of n samples from the joint distribution of $\underline{X}$, $\underline{Y}$ and $\underline{Z}$. To avoid an overload of notation, we denote x_i as the ith sample in S, while X_m is the mth component of the random vector $\underline{X}$.

For the time notation, a multivariate variable in time t is represented by a bracket index, e.g., $\underline{X}_{(t)}$ and a matrix that represents its past l time lags is represented by $X_{(t)}^{(l)} = [\underline{X}_{(t-l)}, \ldots, \underline{X}_{(t)}]$.

B. Definitions

Let $\underline{X}$ be a discrete random variable that follows a probability distribution $P(\underline{X})$. Shannon's entropy is defined as

$$H(\underline{X}) = -\mathbb{E}_{P(\underline{X})}\big[\log P(\underline{x})\big]. \tag{1}$$

The entropy (1) can be represented by the chain rule

$$\begin{aligned} H(\underline{X}) &= H(X_1, X_2, \ldots, X_{d_x}) \\ &= \sum_{m=1}^{d_x} H(X_m|X_{m-1}, \ldots, X_1) \end{aligned} \tag{2}$$

where $H(X_1|X_0)$ abbreviates $H(X_1)$.

The CE between any two distribution functions $P(\underline{X})$ and $Q(\underline{X})$ is defined as

$$\mathrm{CE}(Q(\underline{X})) = -\mathbb{E}_{P(\underline{X})}\big[\log Q(\underline{X})\big] \tag{3}$$

where the expectation is over the distribution of $\underline{X}$, namely, $P(\underline{X})$.

The following inequality holds for every pair of distributions $P(\underline{X})$ and $Q(\underline{X})$:

$$\mathrm{CE}(Q(\underline{X})) \geq H(\underline{X}) \tag{4}$$

where an equality is obtained for $Q(\underline{X}) = P(\underline{X})$.

A related measure to CE is the Kullback–Leibler divergence (D_{KL}) between $P(\underline{X})$ and $Q(\underline{X})$

$$D_{\mathrm{KL}}(P(\underline{X})||Q(\underline{X})) = \mathbb{E}_{P(\underline{X})}\left[\log \frac{P(\underline{X})}{Q(\underline{X})}\right]. \tag{5}$$

The D_{KL} is a nonnegative measure and equals zero iff $P(\underline{X}) = Q(\underline{X})$.

The MI, denoted as $I(\underline{X}; \underline{Y})$, quantifies in bits the entropy reduction in $\underline{X}$ given the knowledge obtained from another random variable $\underline{Y}$, that is,

$$I(\underline{X}; \underline{Y}) = H(\underline{X}) - H(\underline{X}|\underline{Y}). \tag{6}$$

Another important measure that is represented by the difference of entropies is conditional MI (CMI)

$$I(\underline{X}; \underline{Y}|\underline{Z}) = H(\underline{Y}|\underline{Z}) - H(\underline{Y}|\underline{X}, \underline{Z}). \tag{7}$$

CMI is also used to evaluate the TE, which is defined in [50]

$$\mathrm{TE}_{X\to Y} = I\left(\underline{X}_{(t)}^{(k)}; \underline{Y}_{(t+1)}|\underline{Y}_{(t)}^{(l)}\right). \tag{8}$$

Assuming discrete time, the $\mathrm{TE}_{X\to Y}$ is the CMI between the past k time lags of $\underline{X}$ and $\underline{Y}$ at time $t+1$ given the past l time lags of $\underline{Y}$.

C. CE-Based Entropy

Let $P(\underline{X})$ be the distribution function of $\underline{X}$. Let $T_\theta(\underline{X})$ be a neural network that approximates it. In [12], the following upper bound for the entropy of $\underline{X}$ was proposed:

$$H_\Theta(\underline{X}) = \inf_{\theta\in\Theta} \mathrm{CE}(T_\theta(\underline{X})) \tag{9}$$

and $H_\Theta(\underline{X}) = H(\underline{X})$ iff $P(\underline{X}) = T_\theta(\underline{X})$. Given a sample S of size n, the sample mean is used to estimate the CE

$$\widehat{\mathrm{CE}}_n(T_\theta(\underline{X})) = -\frac{1}{n}\sum_{i=1}^{n} \log T_\theta(\underline{x}_i). \tag{10}$$

This estimator is shown to be unbiased under the conditions of the uniform law of large numbers, which are described in Section IV.

Next, an estimator of the entropy is obtained by

$$\widehat{H}_n(\underline{X}) = \inf_{\theta\in\Theta} \widehat{\mathrm{CE}}_n(T_\theta(\underline{X})). \tag{11}$$

McAllester and Stratos [12] suggest an entropy estimator based on the above. However, they require a prior knowledge of $P(X)$. The reason for this requirement is that [12] treat (11) as a maximum likelihood optimization problem, where the parameters θ of the function T are obtained by a training procedure given samples from $\underline{X}$. As such, one should define in advance what is the family of functions for which T belongs to, and optimize its parameters accordingly.

IV. Measuring the Joint Entropy With Neural Networks

In this section we discuss the primary concepts of this article. First, the neural network classifier and its respective CE are formally defined. Then, the NJEE is introduced. Next, we define a strongly consistent estimator and show that the proposed joint entropy estimator satisfies this property. We also provide an algorithmic implementation of the proposed estimator and discuss practical aspects of its implementation. Next, estimator for the joint conditional entropy is provided with the corresponding algorithmic implementation. Using the estimators of the joint entropy and the conditional joint entropy, estimators for MI, CMI, and TE are obtained.

Throughout this work we focus on measures that are continuous functions in an n-dimensional space, unless explicitly stated otherwise. We also use continuously differentiable activation functions, such that the conditions for the universal approximation theorem holds [15], [51].

A. Neural Network Classifier and Classification CE

The following basic definitions are used throughout this section.

Definition 1 (Neural Network Classifier): Let $G_\theta(Y|\underline{X})$ be a neural network model with a random variable input $\underline{X}$ and parameters θ in a compact domain $\Theta \in \mathbb{R}^k$. The outputs of $G_\theta(Y|\underline{X})$ are defined over the probability simplex: $\{G_\theta(y|\underline{x}) \in \mathbb{R}^{a_y} : \sum_{y=1}^{a_y} G_\theta(y|\underline{x}) = 1, G_\theta(y|\underline{x}) \geq 0\}$, where $Y \in \mathcal{A}_y = \{1, \ldots, a_y\}$, $a_y \geq 2$.

Intuitively, a neural network classifier provides a mapping from an input X to an output Y. This output is the probability to obtain every symbol in the alphabet of Y given the input X. For example, mapping a vector of pixel values to probabilities over possible image classes in image classification task. In practice, the probability distribution of Y is obtained by a softmax layer with number of nodes that is equal to the alphabet size of Y (see [52], Section IV for more details).

Next, we define the CE of this classifier.

Definition 2 (Classifier CE): Let $G_\theta(y|\underline{x})$ be a neural network classifier. The CE of this classifier is defined as

$$\mathrm{CE}(G_\theta(Y|\underline{X})) = -\mathbb{E}_{P(\underline{X},Y)} \log G_\theta(y|\underline{x}). \tag{12}$$

We assume that $-\log(G_\theta(y|\underline{x})) \leq \eta$ for all $\underline{x} \in \underline{X}$ and for all $\theta \in \mathbb{R}^k$, for any value of Y. Practically, this assumption is used in many model training procedures to avoid an unbounded loss [14]. The empirical estimator of this CE is given in [53], namely

$$\widehat{\mathrm{CE}}_n(G_\theta(Y|\underline{X})) = -\frac{1}{n}\sum_{i=1}^{n} \log(G_\theta(y_i|\underline{x}_i)). \tag{13}$$

Note that under these definitions, the input $\underline{X}$ is not necessarily discrete. However, the proposed entropy estimator that is introduced in the following section assumes that $\underline{X}$ is discrete as well.

B. Neural Joint Entropy Estimation

Given (2) and Definitions 1 and 2, we define the estimator of the joint entropy.

Definition 3 NJEE: Let $\widehat{H}_n(X_1)$ be an estimated marginal entropy of the first components in $\underline{X}$ and let $G_{\theta_m}(X_m|\underline{X}^{m-1})$ be a neural network classifier. Then, *NJEE* is defined as

$$\widehat{H}_n(\underline{X}) = \widehat{H}_n(X_1) + \sum_{m=2}^{d_x} \widehat{\mathrm{CE}}_n(G_{\theta_m}(X_m|\underline{X}^{m-1}). \tag{14}$$

In words, the joint entropy estimator consists of a marginal estimator for the first component, followed by estimators for the conditional entropies $H(X_m|X^{m-1})$, for $m = 2, \ldots, d_x$.

The main difference between the *NJEE* and the estimator defined in (11) is the use of the chain rule, which enables a self-supervised procedure, where the input to the neural network is composed from the first $m-1$ components of $\underline{X}$, and the target goal is to infer the class of component m of $\underline{X}$. Thus, it does not require any prior knowledge about the underlying distribution of either $\underline{X}$ or $\underline{Y}$.

Definition 4 [Strong Consistency (Following [19])]: The estimator $\widehat{H}_n(\underline{X})$ is strongly consistent if for all $\epsilon, \delta > 0$ and a constant $C > 0$, there exists a positive integer N and a choice of a neural network such that

$$\forall n \geq N, \quad |H(\underline{X}) - \widehat{H}_n(\underline{X})| \leq \; C \cdot \epsilon + \delta, \quad a.e.$$

Theorem 1: NJEE is strongly consistent.

C. Proof of Strong Consistency Property

In this section we follow the scheme shown in [19] to prove Theorem 1. This proof includes the following main steps.

1) Connecting the true CE of a classifier-based neural network and the conditional entropy $H(Y|\underline{X})$ (Lemmas 1 and 2).
2) Showing the convergence of the empirical CE to the true CE (Lemma 4).
3) Showing that the empirical CE can approximate with high accuracy the conditional entropy (Lemma 5).
4) Applying the chain-rule property and the previous steps to show that the proposed estimator of the joint entropy is strongly consistent.

We begin with the first step. Formally, since neural networks are universal approximation functions [15], [16], [17], the following holds.

Lemma 1: For any $\epsilon > 0$, and any conditional distribution function $P(Y|\underline{X})$ that is continuous over its support, there exists a neural network $G_\theta(Y|\underline{X})$ such that

$$D_{\mathrm{KL}}(P(Y|\underline{X})||G_\theta(Y|\underline{X})) \leq \frac{\epsilon}{2}, \quad a.e. \tag{15}$$

That is, it is possible to find a neural network that can approximates $P(Y|\underline{X})$ in any desired approximation level.

The next Lemma states that the CE can be used to estimate the conditional entropy.

Lemma 2: Let $P(Y|\underline{X})$ be a conditional distribution and let $H(Y|\underline{X})$ be the entropy associated with this distribution. Then, for any $\epsilon > 0$, there exists a neural network $G_\theta(Y|\underline{X})$ such that

$$|\mathrm{CE}\big(G_\theta(Y|\underline{X})\big) - H(Y|\underline{X})| \leq \frac{\epsilon}{2}, \quad a.e. \tag{16}$$

The proof of this lemma follows the ideas shown in [12]

$$\begin{aligned} H(Y|\underline{X}) &= \mathbb{E}_{P(\underline{X},Y)} \log \frac{1}{P(y|\underline{x})} \\ &= \mathbb{E}_{P(\underline{X},Y)} \log \frac{1}{G_\theta(y|\underline{x})} \frac{G_\theta(y|\underline{x})}{P(y|\underline{x})} \\ &= \mathbb{E}_{P(\underline{X},Y)} \log \frac{1}{G_\theta(y|\underline{x})} - D_{\mathrm{KL}}(P(y|\underline{x})||G_\theta(y|\underline{x})) \\ &\geq \mathrm{CE}(G_\theta(Y|\underline{X})) - \frac{\epsilon}{2} \end{aligned} \tag{17}$$

where the last line follows Lemma 1. As shown in (4), we have that

$$\mathrm{CE}(G_\theta(Y|\underline{X})) - H(Y|\underline{X}) \geq 0 \tag{18}$$

therefore

$$|\mathrm{CE}\big(G_\theta(Y|\underline{X})\big) - H(Y|\underline{X})| \leq \frac{\epsilon}{2}.$$

The empirical estimator for this classifier CE is obtained from (13). The conditions for the convergence of this estimator are defined by the uniform law of large numbers.

Lemma 3: The uniform law of large numbers [54]. Let Θ be a compact set of parameters. Let $f_\theta(\underline{x}_i)$ be a continuous function at each $\theta \in \Theta$ and $\underline{x}_i \in \underline{X}$. Assume there exists an upper bound $\eta(\underline{X})$ such that $\|f(\underline{x})\| \leq \eta(\underline{x})$ for all $\theta \in \Theta$ and $\mathbb{E}[\eta(\underline{X})] < \infty$. Then, $E[f_\theta(\underline{X})]$ is continuous and

$$\sup_{\theta \in \Theta} \left\| \frac{1}{n} \sum_{i=1}^{n} f_\theta(\underline{x}_i) - \mathbb{E}[f_\theta(\underline{X})] \right\| \xrightarrow{p} 0. \tag{19}$$

Using Lemma 3, the convergence of the classifier CE is obtained

Lemma 4: For any $\epsilon > 0$ and $\forall \theta \in \Theta$, there exists a positive integer $n \geq N$ such that

$$P\left(|\widehat{\mathrm{CE}}_n(G_\theta(Y|\underline{X})) - \mathrm{CE}(G_\theta(Y|\underline{X}))| \leq \frac{\epsilon}{2}\right) = 1. \tag{20}$$

The proof of this Lemma is an immediate application of (13) with

$$f_\theta((\underline{x}_i, y_i)) = -\log(G_\theta(y_i|\underline{x}_i)) \tag{21}$$

since $-\log(G_\theta(y_i|\underline{x}_i)) \leq \eta$, then $f_\theta((\underline{x}_i, y_i)) \leq \eta$ and Lemma 3 holds.

Lemma 5: The estimator $\widehat{\mathrm{CE}}_n(G_\theta(Y|\underline{X}))$ is strongly consistent. That is, for all $\epsilon > 0$, there exists a positive integer $n \geq N$ and a choice of neural network such that

$$|H(Y|\underline{X}) - \widehat{\mathrm{CE}}_n(G_\theta(Y|\underline{X}))| \leq \epsilon, a.e. \tag{22}$$

This lemma is obtained using the triangular inequality with Lemmas 2 and 4

$$\begin{aligned} &|H(Y|\underline{X}) - \widehat{\mathrm{CE}}_n(G_\theta(Y|\underline{X}))| \\ &\quad \leq |\mathrm{CE}\big(G_\theta(Y|\underline{X})\big) - H(Y|\underline{X})| \\ &\qquad + |\widehat{\mathrm{CE}}(G_\theta(Y|\underline{X})) - \mathrm{CE}(G_\theta(Y|\underline{X}))| \leq \epsilon. \end{aligned} \tag{23}$$

Restating (2)

$$H(\underline{X}) = H(X_1) + \sum_{m=2}^{d_x} H(X_m|\underline{X}^{m-1}). \tag{24}$$

Suppose there exists $d_x - 1$ neural networks that approximate each term in the sum with an ϵ accuracy. Then, the total error of the sum expression is $\epsilon \cdot (d_x - 1)$. The marginal entropy $H(X_1)$ is estimated with an estimator $\widehat{H}_n(X_1)$ that guarantees an error that is not larger than certain $\delta > 0$. Several estimators can provide such a guarantee, e.g., [9], [11]. In this case

$$\begin{aligned}
&|H(\underline{X}) - \widehat{H}_n(\underline{X})| \\
&\quad = \left| H(X_1) - \widehat{H}_n(X_1) + \sum_{m=2}^{d_x} H(X_m|\underline{X}^{m-1}) \right. \\
&\qquad \left. - \sum_{m=2}^{d_x} \widehat{\mathrm{CE}}_n(G_{\theta_m}(X_m|\underline{X}^{m-1}) \right| \\
&\quad \leq |H(X_1) - \widehat{H}_n(X_1)| \\
&\qquad + \left| \sum_{m=2}^{d_x} H(X_m|\underline{X}^{m-1}) - \sum_{m=2}^{d_x} \widehat{\mathrm{CE}}_n(G_{\theta_m}(X_m|\underline{X}^{m-1}) \right| \\
&\quad \leq \delta + C \cdot \epsilon
\end{aligned} \tag{25}$$

where $C = d_x - 1$. □

D. Algorithmic Implementation of NJEE

The implementation of the *NJEE* estimator is described in Algorithm 1.

Algorithm 1 NJEE

input: Sample $S = \{\underline{x}_i\}_{i=1}^n$ from $P(\underline{X})$
$h_m \leftarrow 0$, **for** $m = \{1, \ldots, d_x\}$
$h_1 \leftarrow \widehat{H}_n(X_1)$
Initialize $\{\theta_m\}_{m=2}^{d_x}$
for m in 2 to d_x **do**
  $h_m \leftarrow$ Minimize $\widehat{CE}_n(G_{\theta_m}(X_m|\underline{X}^{m-1}))$
end for
$\widehat{H}_n(\underline{X}) \leftarrow h_1 + \sum_{m=2}^{d_x} h_m$
return: $\widehat{H}_n(\underline{X})$

Practically, Algorithm 1 can be implemented in parallel per each value of m. Another approach is to use a recurrent neural network (RNN) that replaces the $d_x - 1$ networks. In this case, the sequential input to the RNN is the components vector of $\underline{X}$ (e.g., see distribution estimation with RNN in [55]). Then, the estimated entropy would be the sum of all the CE losses in every time step. The empirical results of this implementation demonstrate similar performance to Algorithm 1.

We also note that using the CE loss, it is possible to replace the neural network model with any other classifier to estimate the entropy. However, in this case, Lemma 1 may not apply, and strong consistency is not guaranteed.

E. Conditional-Neural Joint Entropy Estimation

The conditional entropy of two multivariate random variables $\underline{X}$ and $\underline{Y}$ is

$$H(\underline{X}|\underline{Y}) = \sum_{m=1}^{d_x} H(X_m|\underline{Y}, \underline{X}^{m-1}). \tag{26}$$

To estimate (26), a slight change is made to *NJEE*, where all components in the proposed estimator are neural networks [c.f. (14)].

Definition 5: C-NJEE. Let $G_{\theta_m}(X_m|\underline{Y}, \underline{X}^{m-1})$ be a neural network classifier with inputs $\underline{Y}$ and $\underline{X}^{m-1}$. Then *C-NJEE* is defined as

$$\widehat{H}_n(\underline{X}|\underline{Y}) = \sum_{m=1}^{d_x} \widehat{\mathrm{CE}}_n(G_{\theta_m}(X_m|\underline{Y}, \underline{X}^{m-1})). \tag{27}$$

Corollary 1: C-NJEE is strongly consistent

$$\left| H(\underline{X}|\underline{Y}) - \sum_{m=1}^{d_x} \widehat{\mathrm{CE}}_n(G_{\theta_m}(X_m|\underline{Y}, \underline{X}^{m-1})) \right| \leq d_x \cdot \epsilon, \quad a.e. \tag{28}$$

The proof of Corollary 1 is straightforward. Notice that every conditional entropy in the sum expression of (26) can be estimated by a classifier CE with ϵ estimation error. Since there are d_x conditional entropies estimators, the total estimation error of $\widehat{H}(\underline{X}|\underline{Y})$ is $d_x \cdot \epsilon$. The implementation of *C-NJEE* is described in Algorithm 2.

Algorithm 2 C-NJEE

input: Sample $S = \{\underline{x}_i, \underline{y}_i\}_{i=1}^n$ from $P(\underline{X}, \underline{Y})$
$h_m \leftarrow 0$, **for** $m = \{1, \ldots, d_x\}$
Initialize $\{\theta_m\}_{m=1}^{d_x}$
for m in 1 to d_x **do**
  $h_m \leftarrow$ Minimize $\widehat{CE}_n(G_{\theta_m}(X_m|\underline{Y}, \underline{X}^{m-1}))$
end for
$\widehat{H}_n(\underline{X}|\underline{Y}) \leftarrow \sum_{m=1}^{d_x} h_m$
return: $\widehat{H}_n(\underline{X}|\underline{Y})$

We now apply *NJEE* and *C-NJEE* to introduce an estimator for the MI

$$\begin{aligned}
\widehat{I}_n(\underline{X}; \underline{Y}) = \widehat{H}_n(X_1) &+ \sum_{m=2}^{d_x} \widehat{\mathrm{CE}}_n(G_{\theta_m}(X_m|\underline{X}^{m-1}) \\
&- \sum_{m=1}^{d_x} \widehat{\mathrm{CE}}_n(G_{\theta_m}(X_m|\underline{Y}, \underline{X}^{m-1})).
\end{aligned} \tag{29}$$

Similarly, given a variable $\underline{Z}$, an estimator for the CMI (7) can be obtained

$$\begin{aligned}
\widehat{I}_n(\underline{X}; \underline{Y}|\underline{Z}) = &\sum_{m=1}^{d_x} \widehat{\mathrm{CE}}_n(G(X_m|\underline{Z}, \underline{X}^{m-1})) \\
&- \sum_{m=1}^{d_x} \widehat{\mathrm{CE}}_n(G(X_m|\underline{Z}, \underline{Y}, \underline{X}^{m-1})).
\end{aligned} \tag{30}$$

Again, since all models are trained independently, the worst case error of these estimators is the sum of the errors of *NJEE* and *C-NJEE*, thus these estimators are also strongly consistent. Note that unlike the entropy estimator, $\widehat{I}_n(\underline{X}; \underline{Y})$ and $\widehat{I}_n(\underline{X}; \underline{Y}|\underline{Z})$ are well defined also in cases where $\underline{X}$ is discrete while Y and/or Z are continuous. This important class of MI estimation problems is discussed in detail in [56].

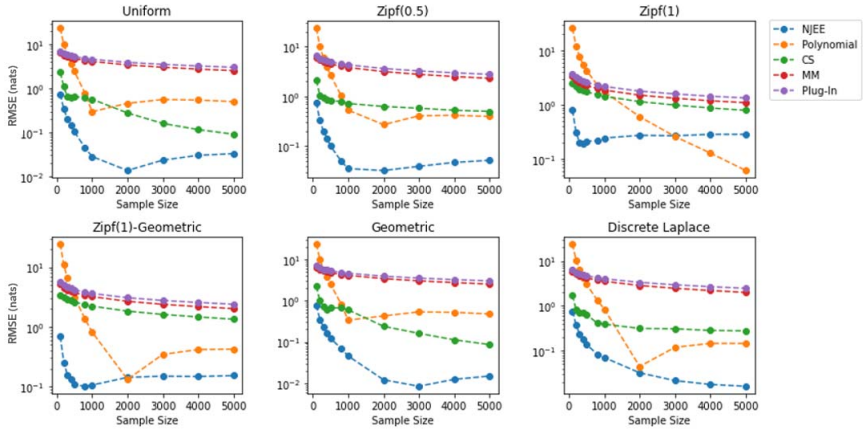


Fig. 1. RMSE (log scale) of entropy estimations versus the log of the sample size for *NJEE* and the benchmark methods from Section II-A (Polynomial [9], CS [10], MM [31], and plug-in [11]), in different simulated studies. The results are the average of 100 measurements per each sample size and distribution type. The standard deviation of the RMSE is negligible with respect to its average value for each estimator.

V. Experiments

In this section, we demonstrate the performance of the proposed estimators in various estimation tasks. A python implementation of the code, including the presented experiments, is located in https://github.com/YuvalShalev/NJEE.

To apply these estimators, we train a set of neural networks. Unless stated otherwise, the following basic network structure is considered throughout these experiments: an input layer, two fully connected layers with 50 nodes, a ReLU activation function, and an output softmax layer. The loss is optimized with the ADAM [57] optimizer with the following parameters ($lr = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$).

A. Entropy Estimation With Large Alphabet

We begin this experimental section with large alphabet entropy estimation using *NJEE*. Prior to applying *NJEE*, we change the univariate representation values of the alphabet to their binary representation. Any other small alphabet representation, such as ternary, is also valid. The evaluation is preformed on six simulated studies, most of which were used in previous works (e.g., [9]).

1) Uniform distribution.
2) Zipf's law distribution with parameters $\alpha = 1, 2$.
3) Geometric distribution with $p = 1/10^5$.
4) Symmetric mixture of a Zipf's law distribution ($\alpha = 1$) and Geometric distribution ($p = 2/10^5$).
5) Discrete Laplace (DL), where $\mathrm{DL}(X, \sigma) \propto (1/2\sigma)e^{-(X/\sigma)}$ and $\sigma = 10^{-4}$.

The alphabet size of X is set to 10^5 (excluding the last experiment where the alphabet is not limited, yet is approximated to 10^5 given the value of σ). Every simulated study (defined by a distribution type and a sample size) is repeated 100 times.

Fig. 1 demonstrates the root mean squared error (RMSE) of the entropy estimation as a function of the sample size for *NJEE* and other entropy estimators described in Section II-A.[1]

[1]The code of the polynomial method is provided by [9] in *https://github.com/Albuso0/entropy*. See the Entropy R package in [58] for the implementation of the other benchmark methods.

As shown, *NJEE* demonstrates the lowest RMSE in most cases. Specifically, *NJEE* demonstrates the lowest error in all the experiments where $n \leq 1000$. This result should not be surprising, since it was demonstrated that neural networks can generalize well on small dataset [59].

B. Multivariate MI Estimation

In the following set of experiments we apply the proposed scheme to a simple multivariate MI estimation problems. This setup is commonly used to benchmark estimators of the MI [19], [20], [21], [40], [41], [42].

The setup is defined as follows. Let $\underline{X}$ and $\underline{Y}$ be two random vectors in $\mathbb{R}^d$ such that

$$\begin{aligned} \begin{bmatrix} \underline{X} & \underline{Y} \end{bmatrix}^T &\sim \mathcal{N}(0, \Sigma_{\mathrm{XY}}) \\ \Sigma_{\mathrm{XY}} &= \begin{bmatrix} I_d & \rho I_d \\ \rho I_d & I_d \end{bmatrix}. \end{aligned} \tag{31}$$

Notice that the correlation between the pairs (X_i, Y_j) is ρ when $i = j$ and zero otherwise. Further, $\mathrm{Cov}(\underline{X}) = \mathrm{Cov}(\underline{Y}) = I_d$, and the MI between $\underline{X}$ and $\underline{Y}$ is thus simply

$$I(\underline{X}; \underline{Y}) = -\frac{d}{2} \cdot \log(1 - \rho^2).$$

In this study, samples are generated from the model above, using different values of ρ (or equivalently, different values of MI).

Our proposed scheme (Algorithm 1), is designed for discrete variables. To apply our scheme to continuous variables, we suggest to first quantize them. Binning of continuous data for MI estimation has been extensively studied over the years [12], [26], [44], [46], [60], [61]. Here, a simple (heuristic) binning scheme is applied. We start by binning the data to a small number of equal-probability bins and estimate the MI using *NJEE*. This procedure is repeated for increasing number of bins, where larger number of bins results in larger MI values. We continue until the change in the obtained MI value as a function of the number of bins is below a predefined threshold, that equals to 0.01 nats in this study. Note that this

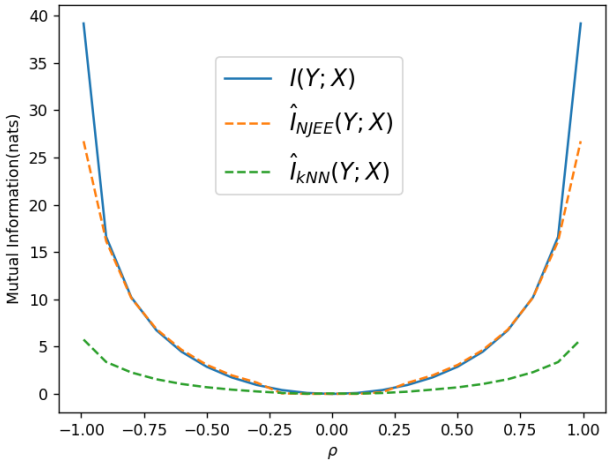


Fig. 2. MI estimation of the study in (31) with various values of ρ. $\widehat{I}_n(\underline{X};\underline{Y})$ is compared to the KNN ($k = 3$) method [18]. The dimensions of $\underline{X}$ and $\underline{Y}$ are 20. The results are obtained from ten repetitions of the simulation with 50 000 samples each. Bin number for the *NJEE*-based estimator is 60. The variance of the RMSE is small with respect to its average value for each estimator.

TABLE I
BEST RESULTS OF EVERY ESTIMATOR FOLLOWING A HYPERPARAMETER GRID SEARCH FOR THE GAUSSIAN SETUP (31) (UPPER ROWS) AND ITS CUBIC TRANSFORMATION (LOWER ROWS). THE TRUE MI VALUES ARE SHOWN IN THE FIRST ROW. THE RESULTS OF THE BENCHMARK METHODS FOR 2 TO 10 NATS ARE ALSO REPORTED IN [20]. THE RESULTS FOR THE *NJEE*-BASED APPROACH ARE OBTAINED WHEN THE DATA IS QUANTIZED TO 250 BINS, USING THE BINNING SCHEME AS DESCRIBED ABOVE

	TRUE MUTUAL INFORMATION					
	2.0	4.0	6.0	8.0	10.0	20.0
GAUSSIAN SETUP						
NJEE	2.2	**4.1**	**5.9**	**7.8**	**9.6**	**19.6**
α	**1.9**	3.8	5.7	7.4	8.8	11.7
JS	1.2	3.0	4.8	6.5	8.1	15.5
NWJ	1.6	3.5	5.2	6.7	8	10.8
InfoNCE	**1.9**	3.6	4.9	5.7	6	6.2
CUBIC SETUP						
NJEE	**2.2**	**4.1**	**5.9**	**7.8**	**9.6**	**18.8**
α	1.7	3.6	5.4	6.9	8.2	-
JS	1	2.8	4.5	6.1	7.6	-
NWJ	1.5	3.2	4.7	5.9	6.9	-
InfoNCE	1.7	3.2	4.1	4.6	4.8	-

binning scheme cannot be applied for entropy estimation of continuous data.

In Fig. 2, the *NJEE*-based algorithms are compared to the KNN MI estimation method [18]. With small absolute values of ρ, the two methods yield accurate results with variance of 0.01 bits. As ρ increases (and thus the MI increases), the KNN estimator significantly deviates from the true value, as demonstrated in [19]. This verifies the results obtained in [32], where it is stated that the KNN estimator requires exponential number of samples in the value of MI to provide accurate estimation. On the other hand, *NJEE* yields better results for greater MI, similar to [12], yet without a prior assumption on the characteristics of the underlying distribution (e.g., normal distribution function). Although not presented here, a similar experiment with low dimensions ($d \leq 6$) and small correlations $\rho \leq 0.5$, demonstrates comparable performance of the KNN estimator to *NJEE* and other neural network-based estimators. To conclude, in problems with low dimensional space and expected small MI, the KNN estimator is a good choice. Otherwise, in possibly more challenging setups, a neural network-based approach should be considered.

Let us now compare NJEE to alternative neural-network-based estimation schemes. Here, we focus on the experimental setup of [19].

Again, we draw samples from the model described in (31). In this experiment, we begin with $\rho = 0$ and draw a total of 4000 batches with 64 samples in each batch. Then, we estimate the MI from the drawn samples. We increase ρ and repeat the previous step. We terminate at $\rho = 1$. $\widehat{I}_n(\underline{X};\underline{Y})$ is compared to the recently proposed variational methods.[2] As demonstrated in Fig. 3, the results achieved by the proposed estimator exhibits lower bias and variance with respect to the variational benchmark methods. The upper rows of Table I demonstrate the best estimation results for each method obtained by hyperparameter grid search. The proposed *NJEE* scheme yields better results for most MI values ranging from 2 to 20.

These results are in line with [12], [20], who discuss the limitations of lower bound estimators in large values of MI. Specifically, in [12], it is shown that the MINE-based lower bound estimator [19] involves a negative exponential term of the *critic*. In this case, the bound is dominated by large values of the *critic*. The probability of extreme events (defined as extreme values that do not appear in a sample) are proven to be large enough to show that such a high confidence lower bound cannot be higher than $\log N$. There, it also shown that this statement is true for any high confidence lower bound. Otherwise, in possibly more challenging problems, a CE-based estimators (11) and *NJEE*, do not have a lower bound. In fact, when bounding the CE loss, estimating the MI reduces to a problem of standard sample mean estimator of the expectation of a bounded variable [12].

Let us now study the estimators' sensitivity to invertible transformation, in which we do not expect any change in the MI under such transformations [1], [63]. The cubic transformation $y \Rightarrow z = (Wy)^3$ is chosen for this experiment, where W is an invertible $d \times d$ matrix with the entries $w_{ij} \sim \mathcal{N}(0, 1)$. The lower rows of Table I summarize the results. As shown, the proposed MI estimator yields identical results to the original problem, while the alternative methods yield lower estimates. Due to stability issues in the benchmark methods, we could not obtain estimates for the cubic transformation when the true MI equals 20.0 nats.

We now study the robustness of *NJEE* to a change in the dimensions of the random variables. We compare the performance of *NJEE* to recently proposed RKHS-based methods, namely the *KKLE* and the *ASKL*-based approaches. In the following experiment, we use the same scheme (31), but this

[2]Poole *et al.* [20] for providing us with the implementation code for the variational methods.

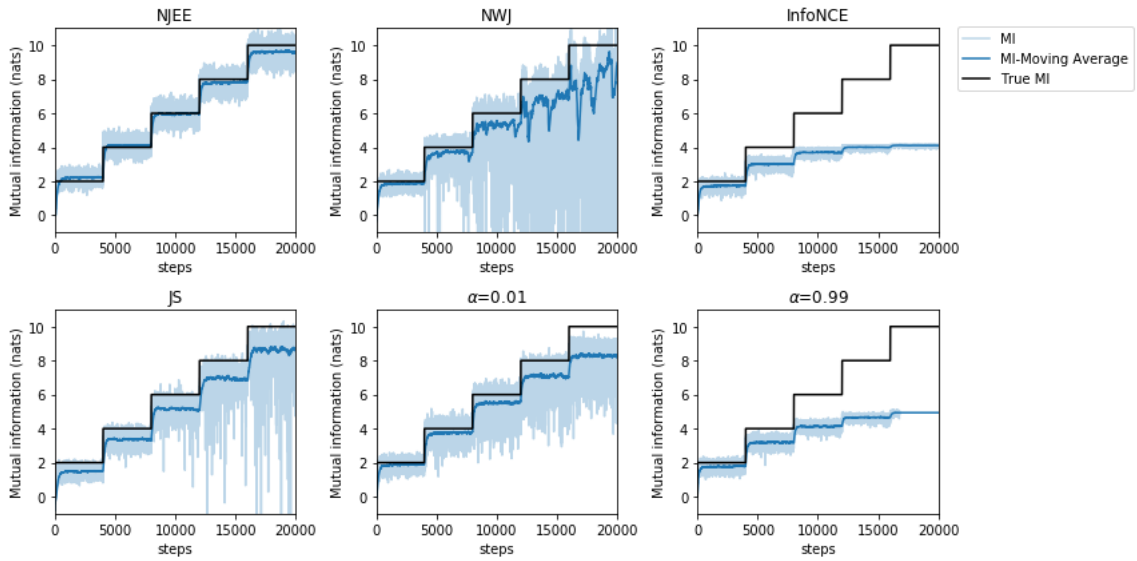


Fig. 3. MI estimation with *NJEE* versus recently proposed variational methods from [20]. Samples from two multivariate random variables in $d = 20$ are generated according to (31) with an increasing ρ every 4000 batches. The estimated MI in every batch appears in light blue, the moving average of the MI over a rolling window of 200 batches is shown in dark blue and the true MI value is represented by the black line. The variational bounds shown in this figure are further discussed in the literature (see *NWJ* [62], *InfoNCE* [5], Jensen-Shannon lower bound (*JS*), and the interpolated bound between *NWJ* and *NCE* with $\alpha = 0.01$ and $\alpha = 0.99$ [20]).

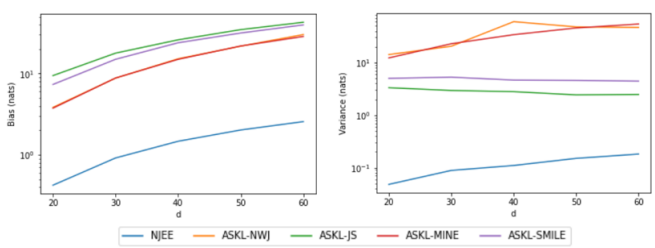


Fig. 4. Bias (left) and the variance (right) of MI estimation of (31), as a function of the variables dimensions d. *NJEE* is compared to the *ASKL*-based approach [41] that limits the *critic*'s hypothesis space to RKHS. As in the original article, this approach is applied on the following variational lower bounds: *NWJ* [62], *JS* [20], *MINE* [19], and *SMILE* [21].

time we change the MI by gradually increasing the features dimensions d, while holding constant the value of ρ. In this specific experiment, we use $20 \leq d \leq 60$ and $\rho = 0.9$. Using publicly available code implementation of *KKLE* provided by Ahuja [40],[3] we were not able to obtain MI estimation in such values of d, due to convergence issues. This result is in line with the results obtained in [40], where *KKLE* introduced a large estimation error when $d = 5$. In Fig. 4, the bias and the variance of the *NJEE*-based and the *ASKL*-based approaches [41], are compared.[4]

As demonstrated, the *NJEE*-based approach provides significantly lower bias and variance in all values of d, specifically in larger values, where the *ASKL*-based approach demonstrates large bias and variance values.

C. Independence Test

Two random variables are independent if and only if the MI between them equals zero [64]. Therefore, we can apply an independence test using MI estimation. We follow the simulated experiment in [39] and [65] to compare between our proposed method and RKHS-based schemes [39].[5] We begin with sampling n examples from two independent univariate random variables, each chosen at random from the following list.

1) Uniform.
2) Normal.
3) Student's t with three degrees of freedom.
4) Student's t with five degrees of freedom.
5) Laplace.
6) Exponential.

For simplicity, we scale all the distributions to zero mean and a unit variance. For example, in an arbitrary simulation run, one set of samples is drawn from a uniform distribution while the other is sampled from a Laplace distribution.

Next, the samples are mixed by a rotation matrix with an angle $0 \leq \gamma \leq (\pi/4)$. Notice that for $\gamma = 0$, two independent univariate samples are obtained, while the strongest dependency is obtained for $\gamma = (\pi/4)$. To generate random samples with a dimension d larger than one, we add a vector of n samples from a standard normal distribution per each additional dimension. Then, the samples are multiplied with an arbitrary orthogonal matrix to obtain dependency across all dimensions. We estimate the corresponding MI using our proposed scheme and [39]. This experiment is repeated 100 times for different pairs of n and γ. Under the null hypothesis H_0,

[3]https://github.com/ahujak/KKLE

[4]The code implementation of the *ASKL*-based approach is provided by its authors in https://github.com/blackPython/mi_estimator

[5]The code implementation of the method in [39] can be found at https://github.com/jthoth/InfiniteDivisibleKernels.

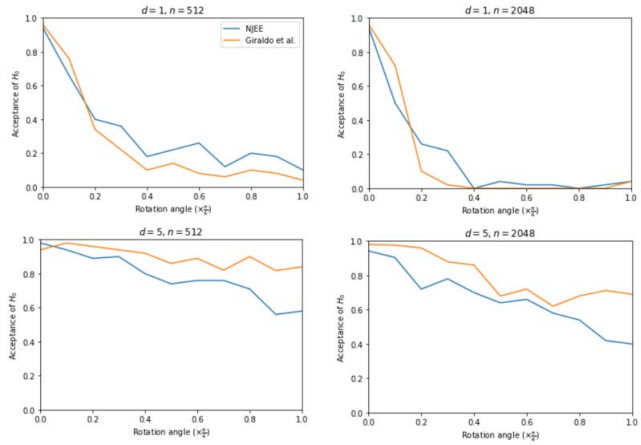


Fig. 5. Independence testing experiment. The acceptance rate of the null hypothesis as a function of the rotation angle γ. Top: Univariate case ($d = 1$). Bottom: Multivariate case ($d = 5$).

the two set of samples are independent. To reject H_0 with a confidence level α, we evaluate the p-value of the sample; the probability to attain the observed MI (or greater than it) under the null hypothesis. Unfortunately, we do not have an analytical expression for the null distribution. Therefore, we simulate it by additional shuffled draws, such that the samples are independent. Then, we compute the p-value as the quantile of the (numerically evaluated) distribution, and reject the null if the quantile is smaller than α. Fig. 5 demonstrates the results we achieve for different values of n and d, as a function of the rotation angle γ, for $\alpha = 0.05$. We expect a decrease in the acceptance rate of H_0 as γ increases, where the ideal estimator would accept H_0 only for $\gamma = 0$ and reject it otherwise. The top row of Fig. 5 shows the results for $d = 1$. In this case, the RKHS-based method [39] outperforms our method, as it provides a grater rejection rate. This is not quite surprising, since our method is mainly designed for problems of larger dimensions. The bottom of row of Fig. 5 demonstrates such a regime. Specifically, it is shown that for $d = 5$, *NJEE* outperforms [39] as it introduces a greater rejection rate for all n and $\gamma > 0$.

D. Conditional Independence Test

We now study the proposed method in conditional independent testing (CIT). CIT is a basic task in statistics with applications to a variety of domains, such as Bayesian networks and causality analysis [66], [67], [68]. In this experiment, we use a flow-cytometry dataset [69]. This dataset describes the connections between eleven proteins in different experimental setups. Sachs *et al.* [69] introduced a consensus Bayesian network (see Fig. 3 in their work) that is considered the ground truth of the connections mapping among the proteins. The flow-cytometry dataset was extensively studied in several works. Mukherjee *et al.* [42] introduced a CIT method that incorporates a two-sampled classifier and generative models. In [67], a KNN bootstrap and binary classifier procedure was proposed to perform the CIT.

Before we describe the results of the experiment, we provide some preliminaries on Bayesian networks that are used for this experiment. In a Bayesian network, features are represented by nodes, and their dependencies are represented by edges [70]. Node A is a parent of node B if there is a directed edge from A to B, and B is considered a child of A. Y is conditionally independent of $\underline{X}$, when there exists a subset of features $\underline{Z}$, which holds all the available information about Y. Using the Bayesian networks convention described above, $\underline{Z}$ includes the parents of Y, its children and the parents of its children (Markov Blanket [71]). Based on these notations, one can choose multiple combinations of dependent and conditionally independent triplet sets of variables. Following the procedures proposed in [42] and [67], 50 dependent and 50 conditionally independent triplets $(\underline{X}, Y, \underline{Z})$ are randomly chosen and their CMI is estimated using $\widehat{I}_n(\underline{X}; Y|\underline{Z})$. For every triplet, we have the ground truth (dependent/independent), and its corresponding estimate $\hat{I}_n(\underline{X}; Y|\underline{Z})$. Since the estimates $\hat{I}_n(\underline{X}; Y|\underline{Z})$ are continuous (nonnegative) numbers, we may set a decision threshold. Specifically, we say that a triplet is conditionally independent if its $\hat{I}_n(\underline{X}; Y|\underline{Z})$ value is lower than a decision threshold ϵ (and vice versa). Thus, one could construct an ROC curve where every point in the curve represents a value of the threshold ϵ, the value of the false positive rate (the horizontal axis) and the true positive rate (the vertical axis). Fig. 6 illustrates the ROC curve and the area under the curve (AUC) values of the independence test performed with $\widehat{I}_n(\underline{X}; Y|\underline{Z})$ and with the benchmarks as reported in [42]. As shown, $\widehat{I}_n(\underline{X}; Y|\underline{Z})$ outperforms the alternative methods.

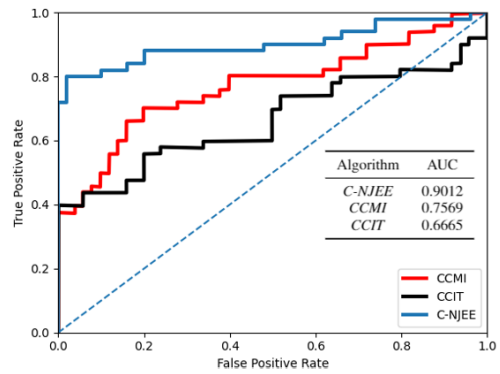


Fig. 6. ROC curve and the AUC values of *C-NJEE* based estimation, *CCIT* [67] and *CCMI* [42] for conditional independence testing task on the flow-cytometry dataset. The dashed line denotes a random model.

E. Estimating TE on Financial Dataset

Finally, we apply *C-NJEE* to TE estimation. For this experiment, we study a financial dataset that contains the daily closing prices of the Dow-Jones Index (DJI - the stock index of 30 large companies in the U.S. stock exchange) and the Hang Seng Index (HSI - the stock index of 50 large companies in the Hong-Kong stock exchange) between 1990 and 2011. As the DJI index is considered more influential than the HSI on the world's financial markets, we expect the TE $\text{TE}_{\text{DJI}\to\text{HSI}}$ to be significantly greater than $\text{TE}_{\text{HSI}\to\text{DJI}}$. Additionally, we expect to see changes in the TE that are coordinated with related economic events (e.g., significant financial crises).

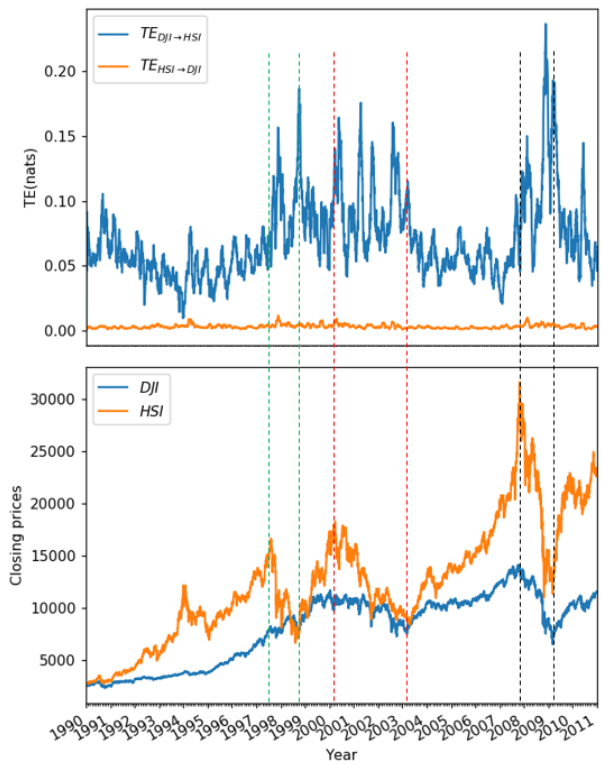


Fig. 7. TE and daily closing prices of the DJI and the HSI. The top chart demonstrates the 30-day moving average of the TE estimated by the *C-NJEE* of DJI to HSI (DIJ $\rightarrow$ HSI) and in the opposite direction (HSI $\rightarrow$ DJI). The bottom chart demonstrates the original closing prices of the two time series. Periods of financial stress with a significant decrease in the index prices are defined between a pair of dotted lines of the same color: the green lines represent the beginning and end of the Asian financial crisis, the red lines represent the beginning and end of the dot-com crisis, and the black lines represent the beginning and end of the 2008 global financial crisis.

To estimate the TE, we reproduce the preprocessing used in [45] and [46], and bin the data to three levels of daily price change. A negative change of more than -0.8% is denoted by -1, an absolute change that is below 0.8% is denoted by 0, and a change that is greater than 0.8% is denoted by $+1$. Then, the *C-NJEE* algorithm is applied with a recurrent neural network that has the following structure: an input layer, followed by an LSTM cell [72] with 50 nodes, a fully connected layer with 50 nodes with ReLU activation and an output softmax layer. This time, a recurrent neural network (RNN) architecture with LSTM cell is chosen, since it is designed for sequential data. The input data to the LSTM network are divided into windows of length five (i.e., five consecutive trading days, the length of a business week). That is, $k = l = 5$ in (8). The optimization procedure includes an ADAM optimizer [57], with the following parameters: $lr = 0.001, \beta_1 = 0.9$, $\beta_2 = 0.999$.

To obtain an average TE over a predefined period of time (e.g., the last 30 days), we first calculate the daily TE. On each day, the TE is estimated using an input window to the model of five days preceding this day. Then, we obtain a series of daily TEs, for which we can calculate the moving average.

The upper chart of Fig. 7 illustrates the 30 day moving average of $\mathrm{TE}_{\mathrm{DJI}\to\mathrm{HSI}}$ and $\mathrm{TE}_{\mathrm{HSI}\to\mathrm{DJI}}$, as measured by *C-NJEE*. As expected, the information flow from DJI to HSI exceeds that of the opposite direction. Compared to the real prices in the lower chart of Fig. 7, a relatively sharp increase in $\mathrm{TE}_{\mathrm{DJI}\to\mathrm{HSI}}$ is observed in times of financial stress where prices are decreasing sharply, such as in the Asian financial crisis (1997–1998), the dot-com crisis (2000–2002), and the 2008–2009 financial turmoil [73]. This phenomenon is well known in the financial literature (e.g., [26]).

Comparing the results of the proposed method to the CTW-based approach [46] and *ITENE* [45], we observe that these methods also found that the information flow from DJI to HSI is much larger than in the opposite direction. However, they did not clearly determine a connection between information values and the world's financial timeline. The reason might be their limited capacity to dynamically analyze complex events in sequential data. E.g., *ITENE* does not consider the sequential characteristics of the data. The CTW-based approach [46] is limited by expressive power of the CTW algorithm, when compared to more advanced class of models such as neural networks.

VI. Conclusion

In this work, we introduce a NJEE. The proposed estimator is based on minimizing the CE using neural networks. Expanding earlier works, we show that *NJEE* is strongly consistent and provide a simple algorithmic implementation that is based on a classification procedure.

We apply the proposed approach to entropy estimation of random variables, specifically those with a large alphabet, using a simple binary transformation. Further, we introduce the C-NJEE, which is an estimator for conditional joint entropy. We use *NJEE* and *C-NJEE* to estimate both MI and CMI.

We demonstrate the performance of the proposed schemes in synthetic and real-world experiments. *NJEE* achieves a lower RMSE on various simulated setups of random variables with large alphabets and relatively small sample size. Moreover, the proposed MI estimator exhibits lower bias and variance compared to newly-proposed variational lower bounds methods. This result is specifically evident in large MI values. The proposed MI estimator is utilized for independence test applications as well, demonstrating better results than the benchmark method when the dimension of the problem increases. The CMI estimator is further used to execute conditional independence tests. Again, the proposed estimator yields larger AUC value than other existing methods. Finally, we demonstrate the abilities of *C-NJEE* in estimating the TE. We investigate the dynamics of information flow among financial time series and show their correlation with significant economic events. Certain important characteristics of these dynamics are not captured by other estimation methods that were implemented on the same dataset.

We further emphasize that the theoretical arguments regarding the existence of neural network-based estimator for the measures discussed in this article, are essentially to justify the use of neural network to estimate information theoretic measures. Additionally, it is important to indicate that cross-entropy minimization is a preferable way to achieve this goal. Unfortunately, there is no guarantee for a specific

network structure that would demonstrate the best results. However, guidelines to the use of neural network as an estimator are provided in detail and can be easily followed. That includes, the use of a specific loss function (cross-entropy), were its input should be softmax layer. Then, classification optimization procedure should be applied, aiming to provide the best classification results using a common hyperparameters search. This way, we compare among several estimators, arguing that the one with the lowest cross-entropy loss is the best entropy estimator. In other words, one may adjust his choice of neural network, considering the data at hand. We believe that future research will use the proposed entropy estimators to develop advanced compression schemes for various types of datasets. Additionally, the MI and CMI estimation capabilities can be used to improve the understanding of complex systems and deep learning frameworks.

REFERENCES

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[2] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learn. Res.*, vol. 5, pp. 1531–1555, Nov. 2004.

[3] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[4] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[5] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[6] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," 2000, *arXiv:physics/0004057*.

[7] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr. 2015, pp. 1–5.

[8] Z. Zhang and X. Zhang, "A normal law for the plug-in estimator of entropy," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 2745–2747, May 2012.

[9] Y. Wu and P. Yang, "Minimax rates of entropy estimation on large alphabets via best polynomial approximation," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3702–3720, Jun. 2016.

[10] A. Chao and T.-J. Shen, "Nonparametric estimation of Shannon's index of diversity when there are unseen species in sample," *Environ. Ecol. Statist.*, vol. 10, no. 4, pp. 429–443, 2003.

[11] L. Paninski, "Estimation of entropy and mutual information," *Neural Comput.*, vol. 15, no. 6, pp. 1191–1253, 2003.

[12] D. McAllester and K. Stratos, "Formal limitations on the measurement of mutual information," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 875–884.

[13] A. Painsky and G. Wornell, "On the universality of the logistic loss function," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 936–940.

[14] A. Painsky and G. W. Wornell, "Bregman divergence bounds and universality properties of the logarithmic loss," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1658–1673, Mar. 2020.

[15] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[16] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," 2016, *arXiv:1611.03530*.

[17] K. F. E. Chong, "A closer look at the approximation capabilities of neural networks," 2020, *arXiv:2002.06505*.

[18] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, Jun. 2004, Art. no. 066138.

[19] M. I. Belghazi *et al.*, "MINE: Mutual information neural estimation," 2018, *arXiv:1801.04062*.

[20] B. Poole, S. Ozair, A. van den Oord, A. A. Alemi, and G. Tucker, "On variational bounds of mutual information," 2019, *arXiv:1905.06922*.

[21] J. Song and S. Ermon, "Understanding the limitations of variational mutual information estimators," 2019, *arXiv:1910.06222*.

[22] J. Jiao, K. Venkat, Y. Han, and T. Weissman, "Minimax estimation of functionals of discrete distributions," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2835–2885, May 2015.

[23] R. Vicente, M. Wibral, M. Lindner, and G. Pipa, "Transfer entropy—A model-free measure of effective connectivity for the neurosciences," *J. Comput. Neurosci.*, vol. 30, no. 1, pp. 45–67, 2011.

[24] P. Wollstadt, M. Martínez-Zarzuela, R. Vicente, F. J. Díaz-Pernas, and M. Wibral, "Efficient transfer entropy analysis of non-stationary neural time series," *PLoS ONE*, vol. 9, no. 7, Jul. 2014, Art. no. e102833.

[25] R. Marschinski and H. Kantz, "Analysing the information flow between financial time series," *Eur. Phys. J. B-Condens. Matter Complex Syst.*, vol. 30, no. 2, pp. 275–281, Nov. 2002.

[26] T. Dimpfl and F. J. Peter, "The impact of the financial crisis on transatlantic information flows: An intraday analysis," *J. Int. Financial Markets, Inst. Money*, vol. 31, pp. 1–13, Jul. 2014.

[27] L. Barnett, A. B. Barrett, and A. K. Seth, "Granger causality and transfer entropy are equivalent for Gaussian variables," *Phys. Rev. Lett.*, vol. 103, no. 23, Dec. 2009, Art. no. 238701.

[28] P. Duan, F. Yang, T. Chen, and S. L. Shah, "Direct causality detection via the transfer entropy approach," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 6, pp. 2052–2066, Nov. 2013.

[29] S. Verdú, "Empirical estimation of information measures: A literature guide," *Entropy*, vol. 21, no. 8, p. 720, 2019.

[30] T. Bossomaier, L. Barnett, M. Harré, and J. T. Lizier, *An Introduction to Transfer Entropy*. Cham, Switzerland: Springer, 2016, pp. 65–95.

[31] G. Miller, "Note on the bias of information estimates," in *Information theory in Psychology II-B*, H. Quastler, Ed. Glencoe, IL, USA: Free Press, 1955, pp. 95–100.

[32] S. Gao, G. V. Steeg, and A. Galstyan, "Efficient estimation of mutual information for strongly dependent variables," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, vol. 38. San Diego, CA, USA: PMLR, 2015, pp. 277–286.

[33] J. Walters-Williams and Y. Li, "Estimation of mutual information: A survey," in *Proc. Int. Conf. Rough Sets Knowl. Technol.* Berlin, Germany: Springer, 2009, pp. 389–396.

[34] Z. Qin, D. Kim, and T. Gedeon, "Rethinking softmax with cross-entropy: Neural network classifier as mutual information estimator," 2019, *arXiv:1911.10688*.

[35] J. C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. New York, NY, USA: Springer, 2010.

[36] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces," *J. Mach. Learn. Res.*, vol. 5, pp. 73–99, Jan. 2004.

[37] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf, "Injective Hilbert space embeddings of probability measures," in *Proc. 21st Annu. Conf. Learn. Theory*. Madison, WI, USA: Omnipress, 2008, pp. 111–122.

[38] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *J. Mach. Learn. Res.*, vol. 11, pp. 1517–1561, Jan. 2010.

[39] L. G. S. Giraldo, M. Rao, and J. C. Principe, "Measures of entropy from data using infinitely divisible kernels," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 535–548, Jan. 2015.

[40] K. Ahuja, "Estimating Kullback–Leibler divergence using kernel machines," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 690–696.

[41] P. A. Sreekar, U. Tiwari, and A. Namboodiri, "Reducing the variance of variational estimates of mutual information by limiting the critic's hypothesis space to RKHS," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 10666–10674.

[42] S. Mukherjee, H. Asnani, and S. Kannan, "CCMI: Classifier based conditional mutual information estimation," 2019, *arXiv:1906.01824*.

[43] J. Runge, J. Heitzig, V. Petoukhov, and J. Kurths, "Escaping the curse of dimensionality in estimating multivariate transfer entropy," *Phys. Rev. Lett.*, vol. 108, no. 25, 2012, Art. no. 258701.

[44] A. Montalto, L. Faes, and D. Marinazzo, "MuTE: A MATLAB toolbox to compare established and novel estimators of the multivariate transfer entropy," *PLoS ONE*, vol. 9, no. 10, 2014, Art. no. e109462.

[45] J. Zhang, O. Simeone, Z. Cvetkovic, E. Abela, and M. Richardson, "ITENE: Intrinsic transfer entropy neural estimator," 2019, *arXiv:1912.07277*.

[46] J. Jiao, H. H. Permuter, L. Zhao, Y.-H. Kim, and T. Weissman, "Universal estimation of directed information," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6220–6242, Oct. 2013.

[47] Y. Shalev and I. Ben-Gal, "Context based predictive information," *Entropy*, vol. 21, no. 7, p. 645, Jun. 2019.

[48] F. M. J. Willems, "The context-tree weighting method: Extensions," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 792–798, Mar. 1998.

[49] Y. Liu and S. Aviyente, "The relationship between transfer entropy and directed information," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Aug. 2012, pp. 73–76.

[50] T. Schreiber, "Measuring information transfer," *Phys. Rev. Lett.*, vol. 85, no. 2, p. 461, 2000.

[51] F. Scarselli and A. C. Tsoi, "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results," *Neural Netw.*, vol. 11, no. 1, pp. 15–37, 1998.

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

[53] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8778–8788.

[54] K. Newey and D. McFadden, "Large sample estimation and hypothesis," in *Handbook of Econometrics*, vol. 4, R. F. Engle and D. L. McFadden, 1994, pp. 2112–2245.

[55] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle, "Neural autoregressive distribution estimation," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 7184–7220, 2016.

[56] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLoS ONE*, vol. 9, no. 2, 2014, Art. no. e87357.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[58] J. Hausser and K. Strimmer, "Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks," *J. Mach. Learn. Res.*, vol. 10, no. 50, pp. 1469–1484, 2009. [Online]. Available: http://jmlr.org/papers/v10/hausser09a.html

[59] M. Olson, A. J. Wyner, and R. Berk, "Modern neural networks generalize on small data sets," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3623–3632.

[60] K. Hlavácková-Schindler, M. Paluš, M. Vejmelka, and J. Bhattacharya, "Causality detection based on information-theoretic approaches in time series analysis," *Phys. Rep.*, vol. 441, no. 1, pp. 1–46, 2007.

[61] T. Dimpfl and F. J. Peter, "Using transfer entropy to measure information flows between financial markets," *Stud. Nonlinear Dyn. Econometrics*, vol. 17, no. 1, pp. 85–102, 2013.

[62] X. Nguyen, M. J. Wainwright, and M. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5847–5861, Nov. 2010.

[63] N. Carrara and J. Ernst, "On the estimation of mutual information," *Proceedings*, vol. 33, no. 1, p. 31, 2020.

[64] W. Li, "Mutual information functions versus correlation functions," *J. Stat. Phys.*, vol. 60, nos. 5–6, pp. 823–837, Sep. 1990.

[65] A. Gretton and L. Györfi, "Consistent nonparametric tests of independence," *J. Mach. Learn. Res.*, vol. 11, no. 20, pp. 1391–1423, 2010.

[66] L. M. de Campos, "A scoring function for learning Bayesian networks based on mutual information and conditional independence tests," *J. Mach. Learn. Res.*, vol. 7, pp. 2149–2187, Oct. 2006.

[67] R. Sen, A. T. Suresh, K. Shanmugam, A. G. Dimakis, and S. Shakkottai, "Model-powered conditional independence test," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2951–2961.

[68] K. Zhang, J. Peters, D. Janzing, and B. Schoelkopf, "Kernel-based conditional independence test and application in causal discovery," 2012, *arXiv:1202.3775*.

[69] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan, "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, vol. 308, no. 5721, pp. 523–529, Apr. 2005.

[70] I. Ben-Gal, "Bayesian networks," in *Encyclopedia of Statistics in Quality and Reliability*. Hoboken, NJ, USA: Wiley, 2007.

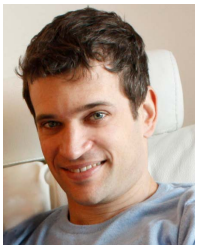
[71] A. Statnikov, N. I. Lytkin, J. Lemeire, and C. F. Aliferis, "Algorithms for discovery of multiple Markov boundaries," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 499–566, Feb. 2013.

[72] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[73] M. McAleer, J. Suen, and W. K. Wong, "Profiteering from the dot-com bubble, subprime crisis and Asian financial crisis," *Jpn. Econ. Rev.*, vol. 67, no. 3, pp. 257–279, 2016.

Yuval Shalev received the B.Sc. and M.Sc. degrees in physics from The Hebrew University of Jerusalem, Jerusalem, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in industrial engineering from Tel Aviv University, Tel Aviv, Israel, in 2022.

Since 2008, he has held various positions as a leading Data Scientist and Manager in the fintech and the automotive industries. He focused his research on the relationship between information theory and machine learning algorithms, specifically neural networks.



Amichai Painsky received the B.Sc. degree in electrical engineering from Tel Aviv University, Tel Aviv, Israel, in 2007, the M.Eng. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 2009, and the Ph.D. degree in statistics from the School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel, in 2017.

He was a Post-Doctoral Fellow, co-affiliated with the Israeli Center of Research Excellence in Algorithms (I-CORE), with The Hebrew University of Jerusalem, Jerusalem, Israel, and the Signals, Information and Algorithms (SIA) Laboratory, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, from 2016 to 2018. Since 2019, he has been a Faculty Member with the Industrial Engineering Department, Tel Aviv University, where he leads the Statistics and Data Science Laboratory. His research interests include data mining, machine learning, statistical learning and inference, and their connection to information theory.

Irad Ben-Gal is currently a Faculty Member with the Industrial Engineering Department, Tel Aviv University, Tel Aviv, Israel, where he leads the Laboratory of AI and Machine Learning Business and Data Analytics (LAMBDA). He held a visiting professor position with Stanford University, Stanford, CA, USA, and is currently co-heading the TAU/Stanford Digital Living 2030 research project. He is an Expert in machine learning, data science, and predictive analytics with more than 25 years of experience in the field, including close research and development collaborations with world-leading companies and AI-based startups. He published four books and more than 150 scientific articles and patents, supervised dozens of graduate students and received numerous awards for his work.