Raymond & Beverly Sackler School of Physics & Astronomy
Raymond & Beverly Sackler
Faculty of Exact Sciences
Tel Aviv University

Machine-learning iterative calculation of entropy for physical systems

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
M.Sc. Physics

by

Amit Nir

Preparation of this work was guided by Prof. Roy Beck

Tel Aviv University

December 2020

Abstract

Entropy is a fundamental quantity in the study of physical systems. It is strongly related to the level of a physical system's order and plays a crucial role in studying phase transitions, pattern formation, protein folding, and more. Entropy is also fundamental in information theory, where it is related to the amount of information in a given set of data.

However, current entropy estimation methods suffer from a high computational cost, lack of generality or inaccuracy, and the inability to treat complex, strongly interacting systems. Direct enumeration of entropy, for example, becomes computationally infeasible for even small, binary systems. Also, many methods that work well for in equilibrium systems are inapplicable for out-of-equilibrium systems as well.

Inspired by information theory ideas, this study shows that entropy could be calculated by iteratively dividing the system into smaller subsystems and estimating each pair of halves' mutual information. The estimation is performed with a recently proposed machine learning algorithm that works with arbitrary network architectures chosen to fit the system's structure and symmetries at hand. Unlike other recently suggested methods, neural networks are well fitted for 2D and 3D models. This study joins others showing that physics science can benefit from the recent advancements in computer algorithms in general and in machine learning in particular.

The proposed method can be used to calculate various systems' entropy, both thermal and athermal, with state-of-the-art accuracy. Specifically, the method is examined on various classical spin systems and is used to identify the jamming point of a bidisperse mixture of soft disks. Moreover, using methods of transfer learning, the proposed method performs in a reasonably fast manner.

Lastly, it is suggested that besides its role in estimating the entropy, the mutual information itself can provide an insightful diagnostic tool in the study of physical systems. Since mutual information consists of all the relevant physical information in a system, it could identify phase transitions or find correlation lengths.

Table of Contents

Acknowledgments

I thank Prof. Roy Beck for being a wise, patient and devoted advisor. I thank Dr. Yohai Bar-Sinai for a delightful and productive collaboration. Both Roy and Yohai generously shared their experience with me, and I learned a lot from them. I would like to also thank Prof. Eran Sela for his contribution to my work.

List of Figures

1 Introduction

1.1 Thermodynamics

1.1.1 Entropy

In thermodynamics, we study statistical systems using macroscopic quantities, such as heat, free energy, work, and their relation to the physical properties of matter. Entropy (S), a fundamental property of physical systems, is related to the free energy by the relation $F = U - TS$, where F and U are Helmholtz free energy and internal energy of the system, and T is the temperature [4]. The Helmholtz free energy is used for systems at a constant volume and temperature, such as those presented in this work. Under different settings, other free energy functions should be used. The internal energy of the system is often known. The calculation of entropy, however, is much more difficult.

The second law of thermodynamics states that entropy of an isolated system cannot decrease during a spontaneous change [5]. Statistical mechanics has successfully related entropy to the probability distribution of microstates of a system. In that framework, the second law implies that a system spontaneously evolves to the most probable macrostate [5].

For a given set of macroscopic properties, the entropy is interpreted as the amount of uncertainty regarding the system's microstate. It is, therefore, strictly related to the order of a system. Since different phases of matter usually present different order levels, entropy is crucial to identify phase transitions. The behavior of the free energy classifies phase transitions at the critical temperature. Since entropy is related to free energy, it is often used to identify these phase transitions [5].

The mathematical definition of entropy from statistical mechanics point of view is:

$$S = -k_{\mathrm{B}} \sum_i p_i log(p_i), \tag{1.1}$$

where p_i is the probability that the system is in the i-th microstate, and k_{B} is the Boltzmann constant [5].

1.1.2 Continuous Limit

Defining the entropy as a sum over different microstates as in (1.1) is clearly possible only when the number of microstates is finite. When a system has an infinite number of microstates, for example when the system is continuous, Eq. (1.1) is no longer valid.

Shannon offered the differential entropy $\tilde{S} = -\int_x p(x)log(p(x))dx$ as the continuous equivalent to entropy [6]. However, this definition lacks invariance under a change of variables and does not follow entropy's positivity requirement. A correct definition of entropy for continuous variables was introduced later by Jaynes [7].

Although inaccurate, a clear connection between the differential entropy and the discrete entropy could be established for a discretized continuous system. The choice of discretization scheme affects the entropy in a nontrivial manner.

Let us consider a continuous variable $\boldsymbol{x}$, mapped to a discrete variable $I(\boldsymbol{x})$ where I takes one of a finite set of values which we denote $I_1, I_2, \ldots$. Each I_i is associated a pre-subset Ω_i, observation probability p_i and phase-space volume v_i, defined as follows:

$$\begin{aligned} \Omega_i &\equiv \{\boldsymbol{x} \mid I(\boldsymbol{x}) = I_i\}\,, \\ p_i &\equiv \int_{\Omega_i} p(\boldsymbol{x}) d\boldsymbol{x}\,, \\ v_i &\equiv \int_{\Omega_i} 1\, d\boldsymbol{x}\,. \end{aligned} \tag{1.2}$$

In the limit of very fine discretization, i.e. $\max_i\{v_i\} \to 0$, and assuming $p(\boldsymbol{x})$ is not ill-behaved, the second definition can be approximated as

$$p_i \approx p(\boldsymbol{x}_i)v_i, \tag{1.3}$$

where $\boldsymbol{x}_i$ is any point in Ω_i. This approximation is accurate when the discretization is fine enough such that p doesn't change considerably across Ω_i, i.e. when all configurations that are mapped to the same image are roughly equiprobable. When this happens, the differential entropy $\tilde{S}$ can be approximated by a Riemman sum:

$$\begin{aligned} \tilde{S} = -\int p(\boldsymbol{x}) \log p(\boldsymbol{x}) d\boldsymbol{x} &\approx -\sum_i \Big(p(\boldsymbol{x}_i) \log p(\boldsymbol{x}_i)\Big) \cdot v_i \\ \approx -\sum_i \left(\frac{p_i}{v_i} \log\left(\frac{p_i}{v_i}\right)\right) \cdot v_i &= \sum_i (-p_i \log p_i + p_i \log v_i) = S + \sum_i p_i \log v_i\,. \end{aligned} \tag{1.4}$$

We see that $\tilde{S}$ differs from S by a term logarithmic in the resolution size.

As an aside, we note that (1.4) has an intuitive interpretation: $\log v_i$ is exactly the entropy of a uniform distribution over Ω_i (whose probability density is $p = \frac{1}{v_i}$). Therefore, the differential entropy $\tilde{S}$ measures the uncertainty (=entropy) associated with knowing in which Ω_i the observation $\boldsymbol{x}$ resides, plus the average uncertainty (=entropy) associated with knowing where does $\boldsymbol{x}_i$ resides within Ω_i.

1.1.3 Information Theory

Information theory is a field of mathematics, started by Claude Shannon in 1948 [6]. It regards the mathematical theory of storage and communication of information. Entropy is a fundamental concept in the field of information theory as well. Information theory interprets entropy as the amount of "information" in a random variable or a given system.

Shannon derived his formula for entropy using a different method than physicists. He introduced $I(x)$, the information content of an event x. Since the information of an event that always happens should be 0, and the information given by two independent events should be the sum of the different information contents, one can show that $I(x) = -\log p(x)$, where $p(x)$ is the probability of the event x to occur. The minus sign results from the requirement that the information decreases with $p(x)$ increasing. Now given a distribution of events $X \sim P$, Shannon defined the entropy as the average amount of information given by knowing the outcome of X, hence, $S = -\mathbb{E}_{X\sim P}[I(x)] = -\sum_{x\in X} p(x)\log p(x)$.

Understanding entropy from an information theory point of view is very simple. Consider a system of $N(= N_w + N_b)$ pixels, N_w of which are white, and N_b of which are black. In the limit of $N \to \infty$, $N_w = p_w N$, where p_w is the probability to observe a single white pixel (Fig. 1.1). If we look at the number of realizations of the system, Ω, we see that $\log\Omega = \log\frac{N!}{((p_w N)!((1-p_w)N)!}$. Using Stirling's formula we get:

$$\log\Omega = N(-p_w \log p_w - p_b \log p_b) \tag{1.5}$$

We can easily expand Eq. (1.5) to multiple colors $\{i\}$ and get:

$$\begin{aligned} \log\Omega &= N(-\sum_i p_i \log p_i) \\ \frac{1}{N}\log\Omega &= -\sum_i p_i \log p_i \end{aligned} \tag{1.6}$$

Now we can ask how many bits of information are necessary to describe a specific realization of the image. The number of required bits is 2^{NH} where $H = -\sum_i p_i \log p_i$ is Shannon's entropy.

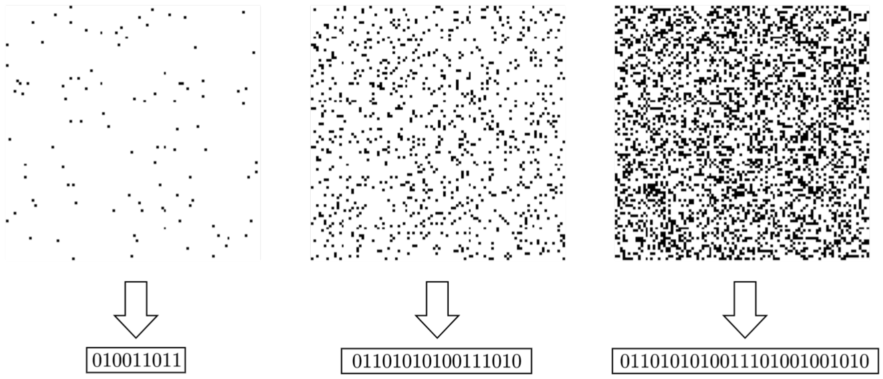


Figure 1.1: Black and white images described using a string of bits. The entropy of a black and white image is $H = -p_w \log p_w - p_b \log p_b$, where p_w, p_b are the ratios of white and black pixels. The number of bits necessary to describe a specific realization of a black and white image is 2^{NH}, where N is the total number of pixels in the image.

A similar relation between entropy and information was introduced by Kolmogorov, using Kolmogorov complexity [8]. Kolmogorov complexity measures the length of the minimal program that is required to generate a sequence of characters. A long sequence of N '0s' is considered to have low complexity since a program of size $\log N$ can compute the sequence. It can be shown that Kolmogorov complexity is asymptotically Shannon's entropy [9, 10].

By definition, the lossless compressed length of some data is bounded by the theoretical notion of Kolmogorov complexity [8], entropy is also an essential concept in the field of communications.

1.2 Methods For Entropy Estimation

Being such a fundamental quantity, many methods have been developed to efficiently and accurately calculate a given system's entropy. Some classical methods were developed from a pure physics perspective, while information theory definition of entropy inspires others [11, 1, 12].

Many considerations should be taken when reviewing such a method. Some methods suffer from a high computational cost, lack of generality or inaccuracy, and inability to treat complex, strongly interacting systems. A general, efficient, and accurate method has yet to be developed. Here we will discuss some of the more common methods.

1.2.1 Analytical

For some systems, a closed analytical expression of entropy can be found. These systems are often simple and weakly interacting. For the canonical ensemble, the analytical expression is often derived using the relation between entropy and the canonical partition function Z:

$$S = \left(\frac{\partial k_B T log(Z)}{\partial T}\right)_V \tag{1.7}$$

Such a solution was derived for the 1D Ising model on a lattice, for example [13]. The Ising model's partition function could be written as a trace of transfer matrices representing bonds between neighboring spins. Then, the partition function is written in terms of these matrices' eigenvalues, and an analytical solution can be derived for the entropy (see full derivation in [14]).

If we consider systems with strong, long-range interactions, finding an analytical expression for the partition function is much more challenging. For these systems, transfer matrices or renormalization group methods could no longer be used. These methods usually rely on the fact that the entire system could be looked like an expansion of a small subsystem. This is impossible when far neighbors interact since each particle is affected by the entire system. Therefore, for complex, strong interacting systems, it is usually infeasible to find an analytical expression of entropy.

1.2.2 Direct Enumeration

For systems in thermodynamic equilibrium, with an average energy and constant volume, it can be shown that the microstates are distributed according to a Maxwell-Boltzmann distribution:

$$p_i = \frac{e^{-\frac{E_i}{k_B T}}}{Z}, \tag{1.8}$$

where E_i is the energy of the i-th microstate. Therefore, it is possible to enumerate over all the microstates of a system, calculate their probabilities, and calculate the entropy of the system according to Eqs. (1.1), (1.8). Even for a binary state particle system, this becomes computationally infeasible very fast as the system grows.

1.2.3 Specific Heat Integration

A standard method of estimating thermodynamic systems' entropy is to integrate the specific heat from low temperatures. This method relies on the relations

$$c_V = T\left(\frac{\partial S}{\partial T}\right)_V, \qquad \text{and} \tag{1.9}$$

$$c_V = \frac{\langle E^2\rangle - \langle E\rangle^2}{T^2}, \tag{1.10}$$

where c_V is the heat capacity, E is the energy, and $\langle\cdot\rangle$ denotes thermal averaging. Eq. (1.9) relates the entropy to a measurable quantity, the heat capacity. This is extremely important since it allows the computation of entropy from experiments. This is why, often, heat capacity measurements would be used by experimentalists to identify phase transitions [15], and to validate thermodynamics rules.

Eq. (1.10) is essential since it relates the heat capacity to quantities that could be easily estimated from simulations - the average energy and its fluctuation. This method allows us to estimate the specific heat, viz. the entropy, relatively easily from a physical simulation. $S(T)$ can be calculated using (1.10) and integrating the specific heat from zero temperature to T. When performing the integral it is assumed that $S(T=0)=0$. This assumption could be violated under some conditions. Ginnings et al. calculated the specific heat of aluminum for $T=0^{\circ}$ to $T=900^{\circ}$ [16]. The entropy was later estimated by integrating the specific heat.

This method is problematic for systems that experience high degeneracy at low temperatures. In these cases, estimating the energy fluctuations accurately becomes very hard. Moreover, the integration process could result in recurring errors from a single misestimation of the specific heat. Also, this method applies only to thermal systems, where the temperature is well-defined.

1.2.4 Compression

Inspired by the ideas presented by Kolmogorov and the relation between information content and entropy, recent studies have suggested that compression algorithms could be used to estimate the entropy of physical systems [17, 1, 12]. The compression-based methods capitalize on decades of research in computer science, which resulted in fast and efficient compression algorithms, such as the Lempel-Ziv algorithm or variants of it [18] which are widely available.

Avinery et al. [1] recently showed that using compression algorithms, one can compute, to a good approximation, the entropy of reasonably complex systems. Avinery et al. assume that given a system, its entropy could be estimated by defining an incompressibility content value, η, which is defined by:

$$\eta = \frac{C_d - C_0}{C_1 - C_0}, \tag{1.11}$$

where C_0, C_1 are the minimal and maximal compressed sizes of files generated by compressing the dataset of samples, and C_d is the average size of a file generated by saving the system's state on a binary file. The relation then estimates the entropy:

$$\frac{S}{k_{\mathrm{B}}} = \eta D \log n_s, \tag{1.12}$$

where D is the number of represented degrees of freedom of the system, and n_s is the number of states of each particle in the system (see Fig. 1.2). This method showed to be both extremely efficient and fairly accurate for several physical systems.

Martiniani et al. [17] showed that using compression, one can measure computable information density (CID) of a system as a quantitative measure of entropy for both equilibrium and out-of-equilibrium systems. Later, Bupathi et al. [12] modified this method to be independent of off-lattice systems' discretization scheme. Zu et al. showed that although this method performs well for in-equilibrium systems and simple out-of-equilibrium particle systems, it performs poorly for more complex systems.

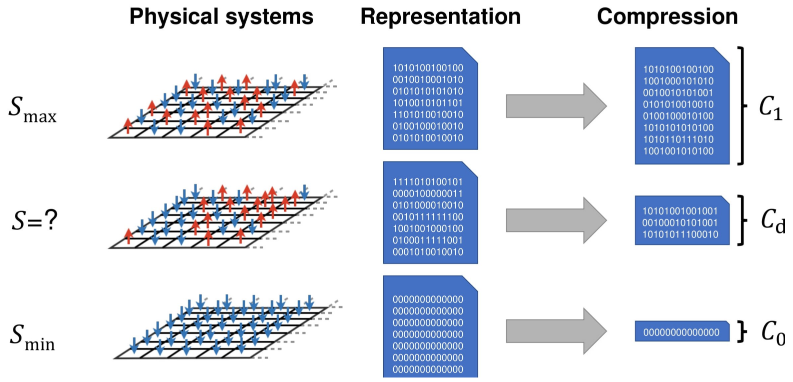


Figure 1.2: Schematic of entropy calculation using a compression algorithm. Simulations of physical systems are preprocessed and encoded into data files. Entropy is directly calculated from the size of the compressed (C_d) and calibration (C_0;C_1) data, as well as the entropy range (S_{min};S_{max}) (see Eq. (1.11)**, Eq.** (1.12)**). Adapted from Avinery et al. [1].**

1.3 Out-of-equilibrium Systems

Studying physical systems that are not in equilibrium with their surroundings is both an old and a new field of study. For example, Boltzmann derived the famous Boltzmann equation, which describes the evolution of the distribution density function of a particle due to collisions and free flight of particles in 1882 [19]. In contrast, Jarzynski's inequality, which relates the difference in free energy between two states in an irreversible process, was derived only in 1996 [20]. Jarzynski's work is part of recent advancements in deriving relations between the thermodynamic properties of matter out-of-equilibrium.

Specifically, the problem of calculating the entropy for non-equilibrium systems is complex both practically and conceptually. Meixner, for example, doubted the existence of unique non-equilibrium entropy [21].

Unlike in-equilibrium, it is impossible to use Eq. (1.8) to estimate the probability to be in microstate i far from equilibrium. Moreover, the definition of temperature is not always possible for these systems. Therefore, new definitions for the entropy were suggested over the years [22]. Often, entropy production would be considered as the studied quantity when discussing out-of-equilibrium systems[23].

However, one can extend the information theory entropy to out-of-equilibrium systems. As explained in Sec. 1.1.3, given a set of microstates $\{i\}$, the entropy of the system is calculated using the information encoded in each microstate i. This is independent of whether or not the distribution p_i is an equilibrium distribution. Recently, several alternative methods were suggested to estimate the entropy for out-of-equilibrium systems [24, 1, 20, 17, 25, 12].

1.4 Mutual Information

In thermodynamics, entropy is considered extensive, hence a quantity that scales linearly with system size [5]. This is only approximately true. The entropy is strictly sub-extensive. The quantity that measures the sub-extensiveness is called mutual information. To be precise, the mutual information ($\mathcal{M}$) between two random variables A, B is defined by the following relation:

$$S(A,B) = S(A) + S(B) - \mathcal{M}(A,B), \tag{1.13}$$

where $S(A), S(B)$ are the entropies of A and B, respectively, and $S(A,B)$ is their joint entropy (Fig. 1.3). It is easy to show that $\mathcal{M}(A,B)$ is strictly non-negative [26]. Therefore, if we think of A and B as two halves of a thermodynamical system, this equation tells us that the entropy of the joint system is smaller than the sum of its components' entropies.

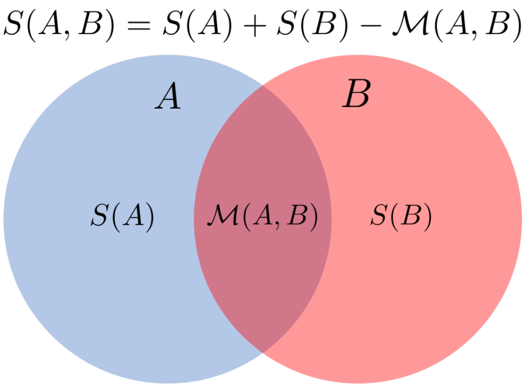


Figure 1.3: Given two systems A, B, the mutual information $\mathcal{M}(A,B)$ quantifies the sub-extensiveness of their mutual entropy.

From an information theory point of view, mutual information quantifies the dependence between two random variables, A and B. Mutual information is used in many fields of thought, such as machine learning, phylogenic profiling, and physics [27, 28, 29]. Koch-Janusz et al. showed that using a machine learning algorithm and mutual information; one can identify the relevant degrees of freedom in physical systems and execute renormalization group steps iteratively to a given system [30].

1.4.1 Mutual Information And Free Energy

Recently, Belghazi et al. proposed a method to calculate the mutual information between high dimensional random variables with neural networks [27]. Their idea is simple and elegant: following a theorem by Donsker and Varadhan [31], the mutual information between two variables, A and B, can be expressed as a solution to a maximization problem.

The proof of this theorem is elegant and straightforward and could be easily understood from a thermodynamics perspective [27]. First, it is known that the mutual information is equal to the Kullback-Leibler divergence (D_{KL}) between the joint distribution, $P_{A,B}$, and the marginal distribution $P_{A\times B}$, where D_{KL} between two probability distributions, P,Q is defined as:

$$D_{KL}(P||Q) := \mathbb{E}_{x\sim P}[\log\frac{p(x)}{q(x)}], \tag{1.14}$$

where $\mathbb{E}_{x\sim P}[x]$ is the mean value of x over P, and $p(x), q(x)$ are the probability distribution functions.

For any function T we can define a partition function $Z = E_Q[e^T]$, and a Boltzmann factor G such that $g(x) = \frac{1}{Z}e^T q(x)$. We get by construction:

$$E_{x\sim P}[T] - \log Z = E_{x\sim P}[\log\frac{g(x)}{q(x)}] \tag{1.15}$$

Now we can define a gap Δ to be:

$$\Delta := D_{KL}(P||Q) - (E_{x\sim P}[T] - \log E_{x\sim Q}[e^T]) \tag{1.16}$$

Using equations (1.14), (1.15) and the positivity of D_{KL} we can easily see that:

$$\Delta = \mathbb{E}_{x\sim P}[\log\frac{p(x)}{q(x)} - \log\frac{g(x)}{q(x)}] = D_{KL}(P||G) \geq 0 \tag{1.17}$$

The inequality Eq. (1.17) is preserved for any function T, and is therefore preserved under taking the supremum of the right hand side. For $G = Q$, namely for optimal functions T^* taking the form $T^*(x) = \log\frac{p(x)}{q(x)} + C, C \in \mathbb{R}$, the bound in (1.17) is tight.

In statistical mechanics, the probability to see some subsystem in a state A is proportional to $e^{-\beta F_A}$, where F_A is the free energy of that state. Therefore, from a statistical mechanics point of view, given two subsystems with a joint distribution $\mathbb{P}_{A,B}$, and a marginal distribution $\mathbb{P}_{A\times B}$, we see that $\Delta = 0$ for the mutual information if:

$$T^*(x) = \log\frac{p_{A,B}(x)}{p_{A\times B}(x)} = \log p_{A,B}(x) - \log p_{A\times B}(x) = -\beta(F_{A,B} - F_{A\times B}), \tag{1.18}$$

where $F_{A,B}$ is the free energy of the joint state, and $F_{A\times B} = F_A + F_B$ is the free energy of the marginal state.

This is correct for a specific set of states, A,B. To calculate the mutual information, we need to average this value over all the states:

$$\mathcal{M} = D_{KL}(P_{A,B}||P_{A\times B}) = -\mathbb{E}_{x\sim P_{A,B}}[\beta F_{A,B} - \beta F_{A\times B}] \tag{1.19}$$

If the two subsystems have no interaction, we see that:

$$\begin{gathered} F = U - TS \\ \mathcal{M} = \mathbb{E}_{P_{A,B}}[S_{A,B} - S_{A\times B}] = \mathbb{E}_{P_{A,B}}[S(A,B) - S(A) - S(B)] \end{gathered} \tag{1.20}$$

where T here is the temperature, and U is the internal energy. We got the same relation we have shown earlier between entropy and mutual information. A more careful treatment should be taken when the subsystems interact, but it will not be shown in this work's scope.

From the previous, we see that $\mathcal{M}(A,B)$ can be written as:

$$\mathcal{M} = \underset{T'\in T}{sup} \left[\left\langle T'(A,B) \right\rangle_{P_{A,B}} - \log \left\langle e^{T'(A,B)} \right\rangle_{P_{A\times B}} \right]. \tag{1.21}$$

where $T : A \times B \to \mathbb{R}$ is a family of functions, $P_{A,B}$ is the joint distribution of A and B, and $P_{A\times B}$ is product of their marginal distributions. We will, later on, define what our T is.

We conclude that estimating the mutual information is equivalent to finding the function that maximizes the right-hand side of (1.21), viz. we transformed the problem of estimating mutual information to an optimization problem.

1.5 Machine Learning and Optimization

Machine learning is attributed to developing computer programs that are not directly programmed but rather "learn" to solve problems independently. In the last decade, machine learning has been the engine behind the rapid increase in computers' capabilities in different tasks, such as image processing [32], classification [33], biometrics [34], and more.

Machine learning algorithms are based on mathematical concepts of probability theory, linear algebra, and optimization. Most machine learning algorithms use the same general setting - building a model based on some data, evaluating the performance of the model, and optimizing it.

In recent years, physicists have found several applications to machine-learning algorithms. Koch-Janusz and Ringel used an artificial neural network to demonstrate renormalization group flow on several statistical physics systems [30]. Kim et al. showed that one could use a machine learning-based technique to compute the entropy production of physical systems [35]. Others have shown that phases transitions could also be detected using machine-learning methods [36, 37].

1.5.1 Artificial Neural Networks

The most common model used in machine learning is the artificial neural network (ANN). ANNs have relatively simple structures; they are based on the perceptron model invented by Frank Rosenblatt in 1958 [38]. The perceptron is a node with n inputs, a weight, a bias, and a non-linear activation function. The inputs of the perceptron are multiplied by the weight, summed, and input to the non-linear activation function (see Fig. 1.4). The most common activation function is the rectified linear unit (ReLU), $ReLU(x) = \max(0,x)$.

ANNs are just layers of perceptrons, with many perceptrons at each layer, where every node is connected to every node in the next layer (see Fig. 1.4). These layers are called fully connected layers. ANNs were researched as early as the early 1940s but did not become a useful machine learning tool until more recent developments allowed efficient training of large networks [39, 40].

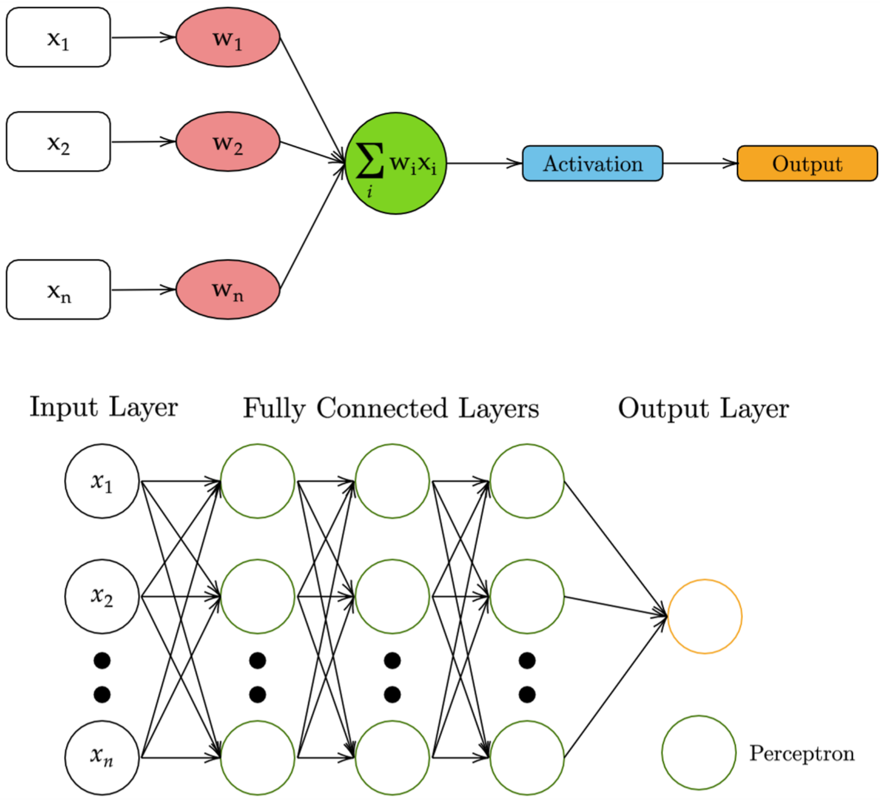


Figure 1.4: The perceptron algorithm - an input vector $x_1, x_2, ..., x_n$ is multiplied by weights $w_1, w_2, .., w_n$ and activated by a non linear function (top figure). A neural network includes many perceptrons connected to each other (bottom figure).

ANNs are known for their ability to represent complex functions. The Universal Approximation Theorem states that using a one-layer neural network with a sigmoid activation function, one could represent any continuous function [41]. This means that theoretically, almost any problem that is solved by finding some optimal input function, as complex as it is, could be solved using an ANN. Of course, the theorem is useless in practice as it will require an enormous number of nodes [42]. In the last decade, advancements in computational tools, specifically the improved graphics processing unit (GPU) performance that enables much faster matrix multiplications and convolutions, have allowed ANNs to become the most useful machine learning models.

1.5.2 Supervised Vs. Unsupervised Learning

Machine learning problems are often divided into two classes. In supervised learning, the data is often labeled, and we can use these labels to evaluate our algorithm's performance. For example, in classification problems, such as identifying cats and dogs images, one could evaluate its current algorithm's performance by measuring how many images are classified correctly using these parameters.

Note that in the presented case, we do not know the mutual information in advance. Therefore, we can not optimize our network with respect to a known value. This also means that we do not know whether or not our result is accurate after optimizing the network. Settings where no labels are available, are called unsupervised settings.

1.5.3 Evaluation

Given a network, we wish to evaluate its performance. In machine learning in general, and particularly in ANNs, this is done using a loss function. A loss function measures the performance of the network. In the learning process, the network tries to minimize the loss function by adjusting its parameters (i.e., weights). In our case, our network's loss function is simply the minus of the right-hand side of equation 1.21, since we want to find a network that minimizes this value.

Often, networks tend to overfit their performance to the specific data set they were given [43]. When a network is tested on a new data set, it might perform worse than we expect. Thus, after it is done training, the final evaluation of a network is usually performed using an independent data set called testing data set.

1.5.4 Optimization

Given a model and an evaluation of its performance, we want to change the model to improve its performance. In ANNs, changing the model means changing the weights and biases of the nodes. The most common method for optimizing ANNs is the gradient descent (GD) method and its variants. GD is an iterative method in which every parameter, w_i, is updated according to $\Delta w_i = -\frac{\partial L}{\partial w_i}$, where L is the loss function. Hence, we minimize the loss function to a local minimum [44].

Although this might sound computational costly, ANNs have a beautiful trick called backpropagation that allows efficient calculation of the gradients [45, 46]. Loosely speaking, backpropagation follows the usage of the chain rule to the derivative of the loss function concerning the last layer and propagates it through every layer back to the first one.

Gradient descent suffers from being computationally costly if calculated over the entire dataset at each iteration [39]. The stochastic gradient descent (SGD) uses a batch of data points at each iteration and computes the gradients solely on them. It can be proven that SGD converges to the same solution as GD in expectation [47, 44]. In our work, we use Adam optimizer (adaptive moment estimation), a variant of SGD [48].

Following the presented methods for evaluation and optimization, neural networks can solve almost any optimization problem. First, a dataset is generated. We then define a loss function we wish to minimize. Using optimization methods, we can find a neural network that minimizes this loss function, i.e., solving the problem on the given dataset.

1.5.5 Transfer Learning

Transfer learning is a method in which weights of a network trained for some setting are used for a different setting. For example, a network trained to classify cats' and dogs' images could classify different images, such as horses and monkeys. In the training process, the network usually learns to extract the input features and then fine tune them to receive the optimal result. Similar settings will usually have similar features. Therefore, transfer learning works very well in practice because it allows the network to use the previously learned features to solve new problems [49].

In practice, transfer learning usually means that the first layers' weights are saved for the new setting, while the last layers of the network are retrained. In our work, we used the transfer learning method to speed-up the training process. For example, we used a network that was trained for some temperature, for nearby temperatures as well.

1.5.6 Convolutional Neural Networks

ANNs suffer from two primary defects - first, the amount of weights rapidly becomes unmanageable for large inputs [42], second, fully connected ANNs perform poorly for input with local spatial features [42]. In a fully connected ANNs, every node is equally connected to every node in the previous layer. Therefore, the architecture has no sense of locality. Although the network could learn to "turn off" connections between distant nodes, this is extremely costly and works poorly in practice. Convolutional neural networks (CNNs) were invented to address these problems.

The lack of locality of an ANN could be solved by connecting each node only to neighbor nodes in the previous layer, an architecture called a locally connected network. CNNs assume that their input's features are both local and symmetric under translations; hence the same set of features could be applied to every region of the input. This concept is called "weight sharing". In a CNN, the network's first layers are built of filters - small matrices of weights convolved with the input (see Fig. 1.5). In such a way, both problems are solved - each filter looks at a spatially small area of the input, and the number of parameters is smaller and independent of the input size [50].

It is considered that the convolutional layers allow the network to extract general features of the input; each layer extracting more delicate features [51]. In most architectures, the last layers of CNNs are standard fully connected layers. In our paper, we used CNN architecture since most of the physical systems we examined show translational invariance.

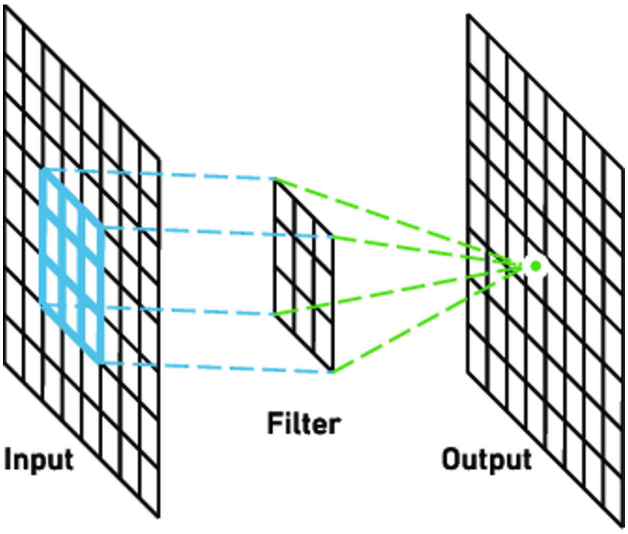


Figure 1.5: A convolutional layer is made of filters. Each filter is a matrix that is convolutionized with the input of the layer. Adapted from [2].

1.6 Physical Models

In the following work we present a method for entropy calculation of physical systems. To demonstrate our method's performance and versatility, we chose four systems representing different classes of collective behavior.

1.6.1 Ising Model

The Ising model is a canonical example of a system with a second-order phase transition. It has been extensively used to solve a variety of physics problems, from liquid-gas models to spin-glasses. The model consists of spins with the binary state (up or down), coupled to their closest neighbors with a positive constant J, on a lattice. The Hamiltonian of the Ising model (under no external field) is given by:

$$H = -\sum_i \sum_{j \in i_{neighbors}} J\sigma_i\sigma_j \tag{1.22}$$

where σ_i is the state of the i-th spin. The problem was analytically solved in one, and two dimensions by Onsager [14]. When examined over a square lattice, the system shows a phase transition from an ordered phase at a low temperature to a randomly oriented phase above the critical temperature (see Fig. 1.6).

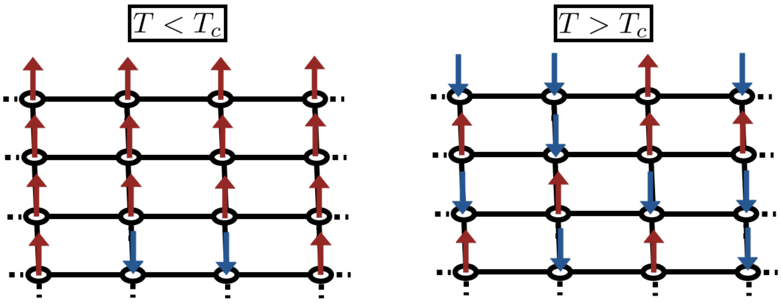


Figure 1.6: The 2D Ising model, a system of coupled binary spins on a square lattice. Below the critical temperature T_c, the spins are ordered (left panel), and above T_c, they are disordered (right panel).

1.6.2 *Antiferromagnetic Ising Model*

The antiferromagnetic Ising model Hamiltonian is described as the ferromagnetic Ising model, but with a coupling constant $J = -1$. This model is examined on a triangular lattice, meaning that each spin has six neighbors (Fig. 1.7). At $T \to 0$, neighboring spins tend to be opposite to each other. On a triangular lattice, this results in a high degeneracy in the number of microstates possible at $T = 0$ and a non-zero entropy, unlike the ferromagnetic case. The antiferromagnetic model has an analytical solution as well [52], and it exhibits no phase transition.

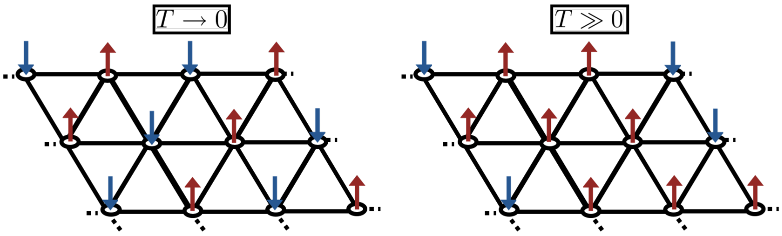


Figure 1.7: The 2D antiferromagnetic Ising model on a triangular lattice. Below the critical temperature T_c, the spins are ordered in opposite directions(left panel), and above T_c, they are disordered (right panel).

1.6.3 *XY Model*

The XY model is another well-studied model of spins. Unlike the Ising model, the spins have a continuous direction. The XY model features a topological phase-transition, called the Kosterlitz-Thouless transition, from a disordered high-temperature state to an ordered low-temperature state.

The phase transition is related to "vortex" points around which spins "turn around" an integer number of times (see Fig. 1.8). At low temperatures, the system consists of bound vortex-antivortex pairs, while at high temperatures, unbound vortices and antivortices are formed [53].

Although an exact solution of the XY model is not possible through the transfer matrix approach, an estimation of the critical temperature could be achieved [54].

The XY model is another fundamental statistical mechanics system, as the Kosterlitz-Thoulsess transition also appears in liquid crystals, thin films of liquid helium, films of superconductors, and more [55, 56].

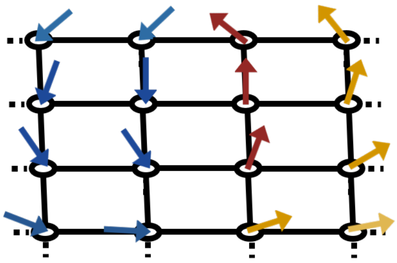


Figure 1.8: The 2D XY model on a square lattice. Spins tend to develop vortices, points around which the spins rotate.

1.6.4 Bidisperse Mixture System

Jammed solids are a prominent class of out-of-equilibrium systems whose entropy plays a crucial role in their dynamics [57]. In these systems, the entropy, which stems from steric interactions, is geometric and measures the number of ways the system's constituents can be ordered in space without overlap. When this depends sensitively on the density, jamming occurs.

Different definitions of what state is considered jammed have been offered, but the most accepted one was offered by O'Hern et al. [58]. They were also the ones who suggested that the jamming point may act as a critical point. By their definition, an unjammed state is a state where the energy per particle E/N is very low, below some cutoff energy. Above the jamming point, the particles are mechanically stable, and they are characterized by finite energy, pressure, and shear yield stress. The energy of the particles is related to overlap with other particles.

The jamming transition is also important as it is thought that understanding it would guide us in understanding one of the most important open problems in condensed matter physics - the glass transition, which is also intimately related to entropic effects [57, 59, 60].

While most jammed systems are 3D, interesting phenomena of jamming are investigated in 2D as well [61]. These systems are easier to investigate as they are simpler and require less computational resources.

A widely investigated system that exhibits a jamming transition is the bidisperse mixture of disks [58]. The 'classic' mixture involves two types of disks, with a radius ratio of $1 : 1.4$, and a $50 : 50$ mixture, although other mixtures have been studied as well.

This system is known to exhibit a jamming transition at a density of $\phi_J \approx 0.84$ [58]. Many methods have been suggested to calculate this density, using dynamic properties such as the jamming length scale or the effective viscosity [62], using static properties such as pair-correlations or fraction of jammed particles, and more [61, 62]. In [61], the jamming transition point ϕ_J is defined as the density from which half of the packings are jammed, according to the definition of [58].

Up to recently, we know of no attempts to estimate the entropy of the bidisperse mixture [63, 64]. Zu and collaborators tried to measure the entropy of such a system using compression based-methods [12]. However, they could not detect a signature of the jamming transition using their estimated entropy.

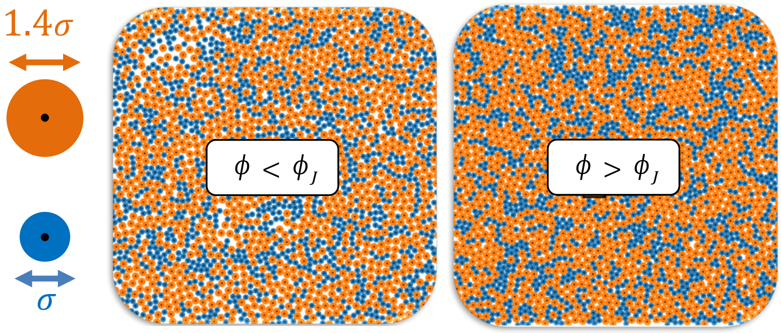


Figure 1.9: The bidisperse mixture of disks below (left) and above (right) the jamming transition. The 'classic' mixture involves two types of disks, with a radius ratio of $1:1.4$, and a $50:50$ mixture. Adapted from [3].

1.7 Simulation Methods

1.7.1 Metropolis Algorithm

Both the ferromagnetic and antiferromagnetic Ising models were simulated using the Metropolis algorithm. Metropolis algorithm is the most common method for simulating many-body systems and generating a representative set of states of a given system [65, 66]. The algorithm works as follows:

- A random initial state is generated.
- A random new state is generated.
- If the new state results in a lower energy state, we advance the system to the new state.
- If the new state is a higher energy state, it is accepted in probability $P = \exp\left(-\beta \Delta E\right)$, where $\beta = \frac{1}{k_B T}$, and ΔE is the difference in energy between the two states.

- Steps 2-4 are repeated until equilibrium is reached.

It can be easily proven that the Metropolis algorithm satisfies two important conditions – detailed balance and ergodicity [66].

1.7.2 Wolff Algorithm

The Metropolis algorithm can fail for some systems, where the system might get "stuck" in local minimal energy states. A simple example could be seen in the XY model. We can consider two states with the same distribution of spins but rotated in a constant angle one from another. Using the Metropolis algorithm, transferring from one state to another would be almost impossible, although they are equally probable.

Many algorithms solve these problems. The Wolff [67] algorithm offers a solution to this problem for spin models by flipping clusters of spins instead of single spins, and it could be described as follows (adapted from [68]):

- A spin i is selected at random and added to a stack.
- A transformation r is generated from a distribution $f(r)$ (a rotation sampled from uniform distribution for example).
- While the stack is not empty
  - Pop a site m from the stack.
  - if the site is not marked:
    - Mark the site.
    - For every neighbor n of the site, add it in probability $min(0, \exp\beta(Z(r\cdot s_m, s_n) - Z(s_m, s_n))$ where Z is the coupling function between two neighbor spins.
    - Take $s_m \to r\cdot s_m$

When the stack is exhausted, all the marked spins are rotated together. This algorithm was used to simulate the XY model system. The algorithm is modified using a "ghost spin" and a modified $\bar{Z}$ coupling function, as discussed in [68].

2 Machine-learning Iterative Calculation of Entropy

2.1 Converting Mutual Information To Entropy

The relation between entropy and mutual information (1.13) allows the calculation of an extensive system's entropy by estimating each of its halves' entropy and the mutual information. Since the computational cost of estimating the entropy grows exponentially with the system size, the latter might be a significantly easier problem than the former.

With this in mind, consider a large physical system X_0, of volume V_0, which we divide to two equal halves. If we deal with translationally invariant systems, as we will assume for the remainder of this work, the two halves are statistically indistinguishable, and we'll denote both of them by X_1. With this notation, Eq. (1.13) takes the form

$$S(X_0) = 2S(X_1) - \mathcal{M}(X_1) \ , \tag{2.1}$$

where $\mathcal{M}(X_k)$ is a shorthand notation for the mutual information between two neighboring subsystems X_k. Each of the halves can be further divided into two statistically indistinguishable halves, and this process can be iterated arbitrarily many times (see Fig. 2.1). After m iterations, we find that:

$$s(X_0) \equiv \frac{S(X_0)}{V} = s_m - \frac{1}{2}\sum_{k=1}^{m} \frac{\mathcal{M}(X_k)}{V_k} \ , \tag{2.2}$$

where $V_k = 2^{-k}V_0$ is the volume (or area in two dimensions) of the k^{th} subsystem, and $s_m \equiv S(X_m)/V_m$ is the specific entropy of the m^{th} subsystem.

Hence, using eq. (2.2) we decomposed the entropy S into contributions from different length scales. At very short scales, the iteration should only be carried out until X_k becomes small enough that its entropy can be directly calculated, either by brute-force enumeration or using other methods. Since V_k decreases exponentially with k, the number of needed iterations is logarithmic in the system size. In many cases, the actual value of the first term in the right-hand side of Eq. (2.2), hence the smallest subsystem's entropy, is an uninteresting additive constant with no physical significance and can be ignored.

In summary, the crux of the method presented in our work is replacing entropy evaluation with the calculation of the mutual information between subsystems of varying sizes. It is left to understand how to calculate the mutual information.

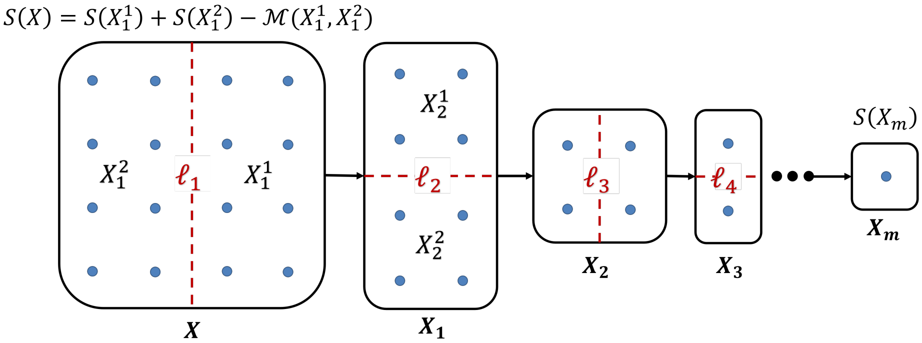


Figure 2.1: The problem of entropy estimation could be converted to a series of mutual information estimations. By estimating the mutual information between two halves of a subsystem $\mathcal{M}(X_{i+1})$, we can estimate $S(X_i)$ from $S(X_i) = 2S(X_{i+1}) - \mathcal{M}(X_{i+1})$.

2.2 Mutual Information Extrapolation

For large enough subsystems, that is, scales much larger than the longest correlation length of the system, we expect $\mathcal{M}$ to grow linearly with the interface length. In precise terms, we expect

$$\mathcal{M}(X_k) = \frac{\ell_k}{\ell_n}\mathcal{M}(X_n) \, , \tag{2.3}$$

where ℓ_k is the interface length between two subsystems in the k-th iteration. If we assume this is obeyed for all systems larger than X_k, this relation can be used to replace the summands in (2.2), and the summation can be done analytically without calculations on subsystems larger than X_k.

2.3 Entropy Estimation

Now that we established that entropy estimation could be transformed into an iterative mutual information estimation problem (Eq. (2.2)), and how ANNs can be used to solve optimization problems, such as mutual information estimation, we can introduce *MICE*, Machine-learning Iterative Calculation of Entropy algorithm which is the core idea of my thesis and the subject of my attached recently published paper [3].

First, using standard methods, we can simulate various physical systems. The simulation results in a sizable dataset of microstates. Then, for each size of the subsystem pair, we generate two datasets: one in which the two subsystems are taken from the same larger sample (the "joint" dataset) and another in which each subsystem is sampled independently (the "product" dataset).

Then, each of the datasets is fed to an ANN, the two averages in Eq. (1.21) are calculated, and the weights of the ANN are updated to maximize the loss (hence, maximize the estimation of $\mathcal{M}$). This process is repeated until the loss stops improving and $\mathcal{M}$ saturates (Fig. 2.2).

Once we have a trained neural network, we feed it with a new, independent testing dataset. We estimate $\mathcal{M}$ on the testing dataset. The estimation is plugged into Eq. (2.2). We use the trained network as the initial state of the neural network for the same subsystem under different conditions (temperature, density).

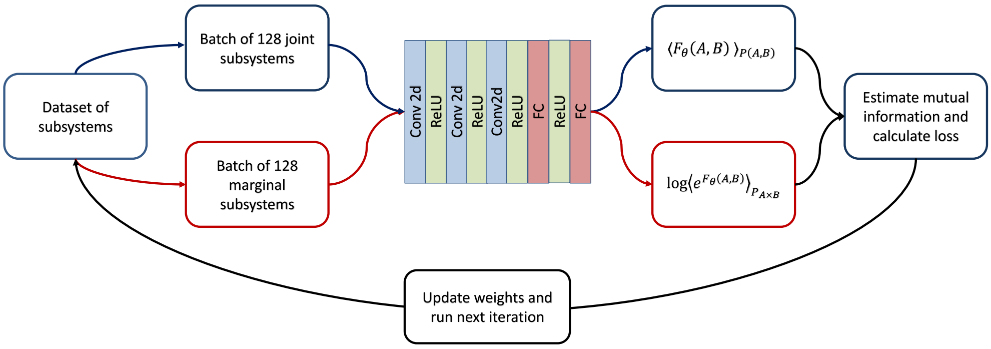

Dataset of subsystems
Batch of 128 joint subsystems
Batch of 128 marginal subsystems
Conv 2d
ReLU
Conv 2d
ReLU
Conv2d
ReLU
FC
ReLU
FC
$\langle F_\theta(A,B) \rangle_{P(A,B)}$
$\log\langle e^{F_\theta(A,B)}\rangle_{P_{A\times B}}$
Estimate mutual information and calculate loss
Update weights and run next iteration

Figure 2.2: The flow of *MICE*. The simulations are used to generate a marginal dataset and a joint dataset. The specific architechture of the ANN shown here was used for subsystem pairs larger than 32×32. Smaller subsystems used 12 convolutional layers. The figure is adapted from [3].

3 Published paper

Machine-learning iterative calculation of entropy for physical systems

Amit Nir (עמית ניר)[a,b], Eran Sela (ערן סלע)[a], Roy Beck[a,b,c,1], and Yohai Bar-Sinai (יוחאי בר-סיני)[a,b,d,1]

[a]The School of Physics and Astronomy, Tel Aviv University, Tel Aviv 69978, Israel; [b]The Center for Physics and Chemistry of Living Systems, Tel Aviv University, Tel Aviv 69978, Israel; [c]The Center for Nanoscience and Nanotechnology, Tel Aviv University, Tel Aviv 69978, Israel; and [d]Google Research, Tel-Aviv 67891, Israel

Edited by Pablo G. Debenedetti, Princeton University, Princeton, NJ, and approved October 16, 2020 (received for review August 11, 2020)

Characterizing the entropy of a system is a crucial, and often computationally costly, step in understanding its thermodynamics. It plays a key role in the study of phase transitions, pattern formation, protein folding, and more. Current methods for entropy estimation suffer from a high computational cost, lack of generality, or inaccuracy and inability to treat complex, strongly interacting systems. In this paper, we present a method, termed machine-learning iterative calculation of entropy (MICE), for calculating the entropy by iteratively dividing the system into smaller subsystems and estimating the mutual information between each pair of halves. The estimation is performed with a recently proposed machine-learning algorithm which works with arbitrary network architectures that can be chosen to fit the structure and symmetries of the system at hand. We show that our method can calculate the entropy of various systems, both thermal and athermal, with state-of-the-art accuracy. Specifically, we study various classical spin systems and identify the jamming point of a bidisperse mixture of soft disks. Finally, we suggest that besides its role in estimating the entropy, the mutual information itself can provide an insightful diagnostic tool in the study of physical systems.

entropy estimation | mutual information | machine learning | jamming

Entropy is a fundamental concept of statistical physics whose computation is crucial for a proper description of many phenomena, including phase transitions (1–3), pattern formation (4), self-assembly (5–7), protein folding (8–10), and many more. In the physical sciences, entropy is typically interpreted as quantifying the amount of disorder of a system or the level of quantum entanglement. Entropy is also a fundamental concept in other fields of thought—statistical learning, economy, inference, and cryptography, among others (11). There it is used to quantify the complexity of statistical distributions. Mathematically, entropy is defined as

$$S = -k_{\mathrm{B}} \sum_i p_i \log p_i, \qquad [1]$$

where p_i is the probability that the system is in the ith microstate, and k_{B} is the Boltzmann constant. For convenience, in what follows we work with units where $k_{\mathrm{B}} = 1$.

Analytic calculation of the entropy is achievable only for simple, weakly interacting systems. Experimentally, the entropy can be obtained, for example, by measuring the temperature (T) dependence of the specific heat down to low temperatures (12). Computationally, for all but the simplest systems, a direct calculation of the entropy is computationally infeasible, as it requires computational resources that grow exponentially with system size (13, 14). For example, a classical numerical approach involves integrating the specific heat, which is inferred from energy fluctuations, down to low temperatures (12). This method is computationally costly and can suffer from inaccuracies for systems with numerous ground states at low T. Other methods estimate directly the free energy (15) or embrace additional knowledge on the system, for example from experiment, to reduce the entropic contribution to a manageable computational task (16).

Recently, we and others have shown that using compression algorithms one can compute, to a good approximation, the entropy of fairly complex systems (8, 17, 18). This method is based on Kolmogorov's theorem that states that the optimal compression of data drawn from a distribution is bounded by the distribution's entropy (19, 20). The compression-based methods capitalize on decades of research in computer science, which resulted in fast and efficient compression algorithms, such as the Lempel–Ziv algorithm or variants of it (21) which are widely available. However, these algorithms treat data as a one-dimensional (1D) discrete string, and manipulating higher-dimensional data into a 1D structure results in information loss. For example, it was recently demonstrated that compression-based algorithms misestimate the entropy of systems with long-range correlations and fail to capture delicate transitions in complex systems (17).

Here, we introduce a generic approach which we term machine-learning iterative calculation of entropy (MICE). Our method improves on existing methods in a number of ways: First, it provides state-of-the-art accuracy. Second, it is scalable, in the sense that its computational cost grows logarithmically with system size. Third, it provides estimations of the actual entropy, with physical units, without additive or multiplicative corrections and with no fitting parameters. Fourth, since the underlying computations are performed with artificial neural nets, MICE can be naturally applied to various physical systems by adjusting the network architecture, rather than the digital representation of the system (e.g., flattening high-dimensional systems to one-dimensional byte arrays as in refs. 8, 17, and 18). Finally, it can be applied to both discrete and continuous distributions.

Below we test MICE on several canonical systems: the Ising model on both square and triangular lattices, the XY model

Significance

The calculation of entropy of a physical system is a fundamental step in learning its thermodynamic behavior. However, current methods to compute the entropy are often system specific and computationally costly. Here, we propose a method that is efficient, accurate, and general for computing the entropy of arbitrary physical systems. Our method is based on computing the mutual information between subsystems within the studied system, using a convolutional neural network. This iterative procedure provides accurate entropy evaluation for systems in and out of equilibrium.

Author contributions: R.B. and Y.B.-S. designed research; A.N., E.S., R.B., and Y.B.-S. performed research; A.N., R.B., and Y.B.-S. analyzed data; and A.N., E.S., R.B., and Y.B.-S. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1] To whom correspondence may be addressed. Email: ybarsinai@gmail.com or roy@tauex.tau.ac.il.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2017042117/-/DCSupplemental.

with and without an external magnetic field (H), and an athermal system of bidisperse soft disks in two dimensions (2D). We show that our approach provides state-of-the-art accuracy and provides insightful information about the physics as a byproduct.

The Method

Entropy and Mutual Information. In thermodynamics, entropy is considered to be an extensive quantity, i.e., a quantity that scales linearly with system size. This is only approximately true. In fact, the entropy is strictly subextensive. The quantity that measures the subextensiveness is called mutual information.

To be precise, the mutual information ($\mathcal{M}$) between two random variables A, B is defined by the relation (11)

$$S(A,B)=S(A)+S(B)-\mathcal{M}(A,B), \qquad [2]$$

where $S(A), S(B)$ are the entropies of A and B, respectively, and $S(A,B)$ is their joint entropy. It is easy to show that $\mathcal{M}(A,B)$ is strictly nonnegative (11). Therefore, if we think of A and B as two halves of a thermodynamical system, this equation tells us that the entropy of the joint system is smaller than the sum of the entropies of its components.

Eq. **2** is the basic relation on which our method relies. It allows calculation of the entropy of a large system by estimating the entropy of each of its halves and the mutual information between them. Since the computational cost of estimating the entropy grows exponentially with the system size, the latter might be a significantly easier problem than the former.

With this in mind, consider a large physical system X_0, of volume V_0, which we divide into two equal halves. If we deal with translationally invariant systems, as we assume for the remainder of this work, the two halves are statistically indistinguishable, and we denote both of them by X_1 (Fig. 1*A*). With this notation, Eq. **2** takes the form

$$S(X_0)=2S(X_1)-\mathcal{M}(X_1), \qquad [3]$$

where $\mathcal{M}(X_k)$ is a shorthand notation for the mutual information between two neighboring subsystems X_k. Each of the halves can be further divided into two statistically indistinguishable halves, and this process can be iterated arbitrarily many times. After m iterations, we find that

$$s(X_0)\equiv\frac{S(X_0)}{V}=s_m-\frac{1}{2}\sum_{k=1}^{m}\frac{\mathcal{M}(X_k)}{V_k}, \qquad [4]$$

where $V_k=2^{-k}V_0$ is the volume (or area in 2D) of the kth subsystem, and $s_m\equiv S(X_m)/V_m$ is the specific entropy of the mth subsystem.

Eq. **4** decomposes the entropy S into contributions from different length scales. At very short scales, the iteration should be carried out only until X_k becomes small enough that its entropy can by directly calculated, either by brute-force enumeration or by using other methods. Since V_k decreases exponentially with k, the number of needed iterations is logarithmic in the system size. In many cases the actual value of the first term in the right-hand side of Eq. **4**, i.e., the entropy of the smallest subsystem, is an uninteresting additive constant with no physical significance and can be ignored.

In summary, the crux of our method is replacing the problem of evaluating the entropy by that of calculating the mutual information between subsystems of varying sizes (Fig. 1*A*). It is left to understand how to actually calculate the mutual information, which is the topic of the next section.

Estimating the Mutual Information. Recently, Belghazi et al. (25) proposed a method to calculate the mutual information between high-dimensional random variables with neural networks. Their idea is simple and elegant: Following a theorem by Donsker and Varadhan (26), the mutual information between two variables, A and B, can be expressed as a solution to a maximization problem:

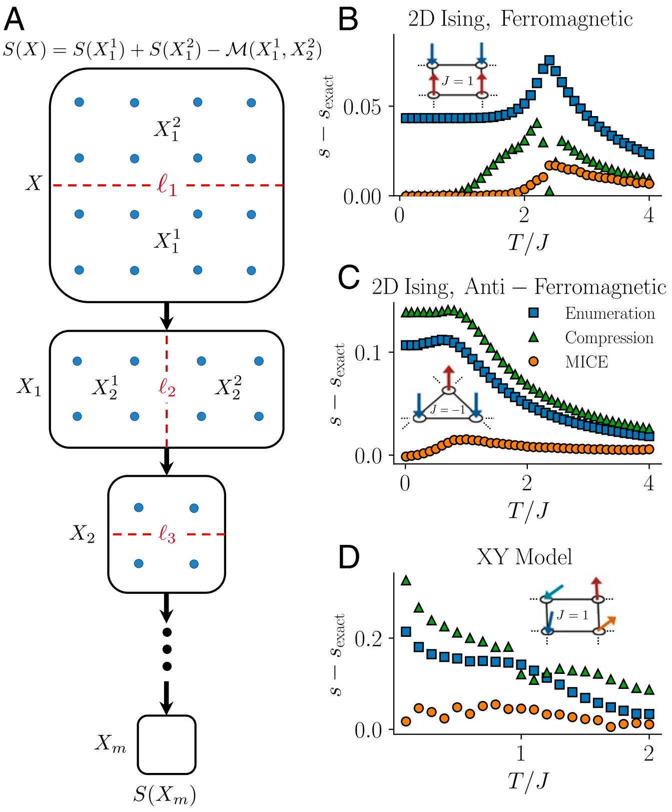


Fig. 1. (*A*) Schematic illustration of MICE. By dividing the system into smaller subsystems and calculating the mutual information between them we reconstruct the entropy of the whole system. The entropy of the smallest subsystem is calculated directly by enumeration. Dashed red lines mark the length of interface (ℓ_i) between two subsystems in the *i*th iteration. (*B*–*D*) The difference between MICE estimations of *s* and known benchmarks. Note that the units are chosen such that $k_B=1$. We present three estimation methods: MICE, naive extrapolation from a system of 16 spins (main text), and a compression-based method (8). MICE shows superior performance in all cases. *B*–*D* show results for (*B*) the ferromagnetic Ising model on a square lattice, (*C*) the antiferromagnetic Ising model on a triangular lattice, and (*D*) the XY model on a square lattice. In *B* and *C* we benchmark against known analytical results for infinite systems (22, 23), respectively. In *D*, we benchmark against the calculation of ref. 24.

$$\mathcal{M}=\sup_{\theta\in\Theta}\left[\langle\mathcal{F}_\theta(A,B)\rangle_{P_{A,B}}-\log\left\langle e^{\mathcal{F}_\theta(A,B)}\right\rangle_{P_{A\times B}}\right]. \qquad [5]$$

Here, $\mathcal{F}_\Theta: A\times B\to\mathbb{R}$ is a family of functions parameterized by a vector of parameters θ, $P_{A,B}$ is the joint distribution of A and B, and $P_{A\times B}$ is product of their marginal distributions. In our case, since A and B are subsystems of a bigger system, $\langle\cdot\rangle_{P_{A,B}}$ means averaging over samples of A and B taken from the same sample of the bigger system, while $\langle\cdot\rangle_{P_{A\times B}}$ means averaging over samples of A and B taken independently. Heuristically, the reason that this representation works is that the mutual information measures how much the joint distribution differs from the product of marginal distributions. In fact, $\mathcal{M}(A,B)$ equals the Kubleck–Leibler divergence between these two distributions (11).

While there is much to be said about Eq. **5**, for the purpose of this work it suffices to note that it reduces the problem of calculating $\mathcal{M}$ to an optimization problem, which naturally suggests the prospect of using artificial neural networks (ANNs) to parameterize the function $\mathcal{F}_\theta$. This is the core idea of Belghazi et al. (25), which we adopt. In machine-learning language, Eq. **5** is taken to be the loss function of the network.

For the complete implementation details see *SI Appendix*, section 1. In broader strokes, the process is as follows: First, using standard methods, a sizable dataset of samples of the system is produced. Then, for each size of subsystem pair we generate two datasets: one in which the two subsystems are taken from the same larger sample (the "joint" dataset) and another in which each subsystem is sampled independently (the "product" dataset). Then, each of the datasets is fed to an ANN, the two averages in Eq. **5** are calculated, and the weights of the ANN are updated to maximize the loss. This process is repeated until the loss stops improving and $\mathcal{M}$ saturates. We found the exponential moving average useful to reduce noise when estimating $\mathcal{M}$ over the final training epochs. Finally, $\mathcal{M}$ is calculated from the trained ANN by averaging Eq. **5** over a separate dataset, different from the one used to train the network.

Results

To demonstrate the performance and versatility of MICE we chose four systems representing different classes of collective behavior: 1) the 2D ferromagnetic Ising model on a square lattice with coupling constant $J=1$, a canonical example of a system with a second-order phase transition; 2) the antiferromagnetic Ising model on a triangular lattice ($J=-1$), a canonical example of a frustrated system with degenerate ground states (27); 3) the continuous XY model on a square lattice, which has a continuous symmetry and features a topological phase transition (27); and 4) finally, an athermal system of a bidisperse mixture of elastic particles which undergoes a jamming transition when its density is increased above a certain threshold (28). For all these systems our method achieves state-of-the-art performance. In addition, in some cases it provides physical insights about the structure and scales of the emergent behavior, as discussed below.

Spin Models. All three spin models were simulated for a system of 64×64 spins with periodic boundary conditions. The distribution was sampled using standard, well-established methods: The Ising models were simulated using Metropolis Monte Carlo simulations as in ref. 8 and the XY model was simulated using the Wolff algorithm as in ref. 29 (*SI Appendix*, section 2).

Lattice systems can naturally be represented as 2D arrays (the triangular lattice can be represented on a square lattice with diagonal interactions) (27). This allows the usage of one of the most successful ANN architectures to parameterize $\mathcal{F}$ of Eq. **5**: feedforward convolutional nets (30, 31). We use one to three convolutional layers, each of 8 to 16 filters of size 3×3, followed by two fully connected layers, using RELU (rectified linear unit) activation, implemented in PyTorch (32). Complete details about the hyperparameters for each model are given in *SI Appendix*, section 1. We calculate $\mathcal{M}$ between subsystems of sizes ranging from a pair of spins to system size. The entropy of a single spin was trivially calculated using brute-force enumeration.

The deviations of our entropy estimations from known results (22–24) are shown in Fig. 1 *B–D*. In all three cases we see impressive quantitative agreement, to a fraction of k_B, with no fitting parameters. We also benchmark our results against the recently proposed compression-based algorithm (8). Relying on highly optimized code and treating the system as effectively 1D, the compression-based algorithm is obviously much faster, about one to two orders of magnitude in terms of runtime. However, while it captures the trend, it offers substantially inferior accuracy in some cases. For example, the low-temperature behavior of the antiferromagnetic Ising model (Fig. 1*C*) is governed by a thermodynamic number of ground states with long-range correlations. There, the error of MICE is smaller by an order of magnitude than that of the compression algorithm method.

It is insightful to compare the performance against another very efficient, albeit naive, estimation of s—calculating s for a small collection of spins by direct enumeration and neglecting the mutual information (i.e., the last term in Eq. **4**). In other words, this is assuming that S is extensive. This estimation, which we refer to as "naive extrapolation," provides only slightly worse accuracy than the compression method, as seen in Fig. 1. In all cases, MICE provides the most accurate calculation with a maximal error of 0.06 k_B per spin for all of the systems and across all temperatures. In *SI Appendix*, section 3 we also use MICE to estimate the heat capacity, showing it outperforms the standard method based on energy fluctuations, since the latter is hard to sample at low temperatures or near a phase transition.

As presented above, our method requires training an ANN for every temperature. This is computationally costly. For example, a single training run for calculating $\mathcal{M}$ between two 64×32 systems of the ferromagnetic Ising model takes several minutes on a standard personal computer. If we were to generate all points in Fig. 1 in this method, the computation time would reach 1 to 2 d. However, drastic improvements in the calculation time can be obtained by leveraging the similarity of the systems between different temperatures. This is done by using the weights (Θ in Eq. **5**) that were obtained by training for a given temperature as the initial conditions of the training process of a different temperature or size. This technique is ubiquitous in the field of machine learning, where it is called "transfer learning" (33). In our case it reduces the training time by one to two orders of magnitude. For additional information see *SI Appendix*, section 1F.

Mutual Information as a Probe. The main purpose of MICE is providing an accurate estimation of S. In addition, the byproduct of the calculation, namely the mutual information between systems at different sizes, which is essentially a decomposition of the entropy to contributions from different length scales, can be an interesting observable in its own right. Here we briefly discuss how it captures insightful aspects of the thermodynamics and can be used to assess the accuracy of the MICE against known limiting behaviors. In passing we note that the mutual information between different scales was shown to be informative in analysis of disordered systems (34, 35).

First, we look at $\mathcal{M}$ between subsystems at various sizes for the ferromagnetic Ising model on a square lattice, plotted in Fig. 2. $\mathcal{M}$ manifestly shows the phase transition (36, 37). Indeed, $d\mathcal{M}/dT$ peaks exactly at the theoretical infinite-system critical temperature $T_c = 2.269J$ (Fig. 2*B*).*

In addition, the accuracy of our calculation can be corroborated against known limits at both high and low temperatures. For $T \ll T_c$, all spins essentially point in the same direction. To be precise, in the low-T limit the ground-state entropy of the whole system, or any subsystem, is exactly $\log(2)$. This implies that the mutual information between any two subsystems is also $\log(2)$ which we indeed observe for all subsystem sizes (Fig. 2*A*).

For $T \gg T_c$, the mutual information between two subsystems can be obtained by a rigorous high-T expansion. The calculation is straightforward but lengthy, and for the sake of brevity its details are given in *SI Appendix*, section 4A. However, the result is short and intuitive: The leading-order behavior at high T is

*In second-order phase transitions the entropy is continuous but its temperature derivative (which is proportional to the heat capacity) (1) diverges. Since S is a sum over $\mathcal{M}(X_i)$ (Eq. **4**), we expect $d\mathcal{M}/dT$ to diverge, rather than $\mathcal{M}$.

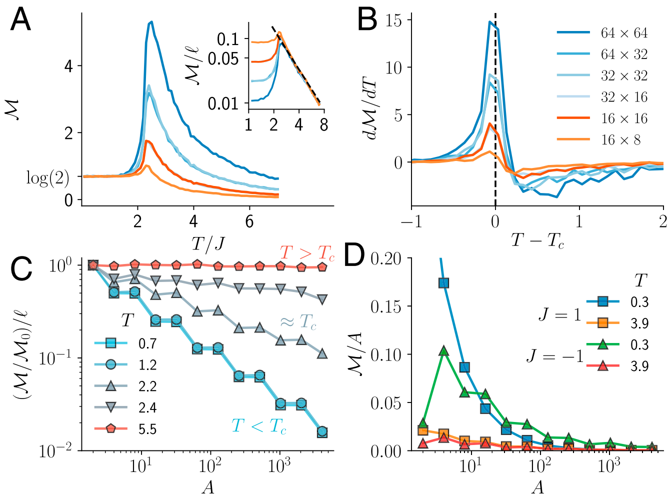


Fig. 2. Analyzing $\mathcal{M}$ for the 2D Ising model on a square lattice for different subsystem sizes. (*A*) $\mathcal{M}$ complies with two known limits: At low temperatures $\mathcal{M} = \log(2)$. At high temperatures $\mathcal{M}$ approaches the theoretical value of Eq. **6**, as shown in *Inset* (dashed line). (*B*) The derivative of the mutual information peaking at the theoretical value $T_c \approx 2.269J$ (23). (*C*) $\mathcal{M}$ normalized by the interface length for varying subsystem sizes (i.e., number of spins). For visual clarity, all curves are normalized to start at unity at zero area. (*D*) $\mathcal{M}$ per area as function of area for the ferromagnetic Ising model on a square lattice (squares) and the antiferromagnetic triangular lattice model (triangles) at various temperatures. $\mathcal{M}$ decays faster for the ferromagnetic model, as the correlation lengths are much shorter.

$$\begin{aligned} \mathcal{M} &= \frac{1}{2}\frac{\ell}{T^2}, \qquad \text{for Ising model} \\ \mathcal{M} &= \frac{1}{4}\frac{\ell}{T^2}, \qquad \text{for XY model,} \end{aligned} \qquad \textbf{[6]}$$

where ℓ is the interface size between the subsystems, i.e., the number of spins in one system that directly interact with spins in the other. As seen in Fig. 2*A*, *Inset*, our method shows excellent agreement with this prediction, again with no fitting parameters. In passing we note that Eq. **6** is akin to the famous area law in quantum entanglement (38).

That is, when $T > T_c$, the mutual information per interface length is independent of the system size, as expected. However, for $T < T_c$ the entropy is not extensive, and $\mathcal{M}/\ell$ decays quickly with the size of the subsystem (Fig. 2*C*). This means that the summands in Eq. **4**, which are $\mathcal{M}$ normalized by the 2D volume (i.e., area), decay quickly for large subsystems. This is visualized in Fig. 2*D*. Fig. 2*D* also shows that in the antiferromagnetic model the summands decay more slowly, which is expected since it features long-range correlations.

Next, in Fig. 3 we examine the entropy and the mutual information in the XY model. At high temperatures $\mathcal{M}$ decays as described in Eq. **6**. Below the critical temperature, the famous Kosterlitz–Thouless transition temperature $T_{KT} = 0.8J$, $\mathcal{M}$ approaches a T-independent plateau for $H \neq 0$ and diverges logarithmically when $H = 0$. This divergence is due to the continuous degeneracy of the XY model, which is lifted in the presence of an external field. In the transition between these limits, $\mathcal{M}$ features a pronounced peak, which becomes smaller and shifts to higher temperatures with increasing H (Fig. 3*C*).

This rich behavior of $\mathcal{M}$ can be understood in simple terms. The high-temperature behavior is accurately described by Eq. **6**, which is a further corroboration of our method (Fig. 3*B*). The low-temperature behavior can be understood, much like in the case of the Ising model, in terms of collective behavior. For $H \neq 0$ and $T < T_{KT}$ all spins are mostly aligned with the field, even if it is relatively small, because of the broken symmetry. In this case, spins fluctuate mildly around their ground state and a harmonic approximation can be made. Within the harmonic approximation the mutual information, $\mathcal{M}_h$ (the subscript h stands for harmonic), can be obtained analytically in terms of block determinants of the Hamiltonian, a derivation which is given in detail in *SI Appendix*, section 4B. The results of this calculation are presented in Fig. 3*C* and show good quantitative agreement.

Finally, we remark that the generic behavior of $\mathcal{M}$—a T-independent plateau at low T followed by a peak and a power-law decay at large T—is also present in very small systems. In fact, even a system of two spins behaves in a qualitatively similar way, although the transition temperatures between the regimes are quite different due to the collective behavior of the spins (Fig. 3*D* and *SI Appendix*, section 5).

A Continuous, Out-of-Equilibrium System

One of the main advantages of MICE is that it is very versatile in terms of the systems it can operate on. As long as a well-defined distribution exists and samples can be drawn from it, and as long as the system can be digitally represented in a manner compatible with ANNs, MICE should be, at least potentially, applicable. In particular, the scheme presented above can be applied to out-of-equilibrium systems, whose entropy calculation is a challenge both technically and conceptually (8, 15, 17, 18, 39, 40). Clearly, the result of MICE will be an estimate of the entropy defined in Eq. **1**, which is the information-theoretic definition of entropy. Relating the result to other thermodynamic properties would depend on the details of the system, which is always the case in calculating thermodynamic properties of out-of-equilibrium systems.

Jammed solids are a prominent class of out-of-equilibrium systems whose entropy plays a crucial role in their dynamics (41). In these systems the entropy, which stems from steric interactions, is geometric in nature and measures the number of ways the

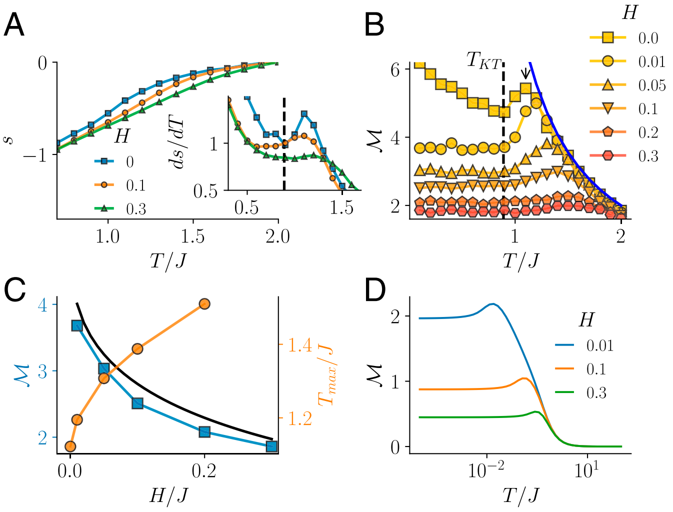


Fig. 3. Analysis of the XY model under the external field (H) using MICE. (*A*) Entropy as a function of temperature for various external fields. *Inset* shows ds/dT, and T_{KT} is marked with a dashed line. (*B*) Mutual information between two systems of size 32 × 16 spins, for varying fields. The arrow marks the peak in $\mathcal{M}(H = 0)$ at $T_{\max}$. The blue line is the high-temperature limit, Eq. **6**. (*C*) Two features of the curves in *B* are replotted: the low-T plateau value (evaluated at $T = 0.1J$), compared to the analytically calculated values at $T \to 0$ in the harmonic approximation, $\mathcal{M}_h$ (black line). $T_{\max}$ is plotted in orange circles. (*D*) Exact numerical calculation of $\mathcal{M}$ between two isolated spins for varying H, showing qualitatively similar behavior to that in *B* (although note that the temperature axis is logarithmic, unlike in *B*).

system's constituents can be ordered in space without overlap. When this depends sensitively on the density, jamming occurs. The jamming transition is also important as it is thought that understanding it would guide us in understanding one of the most important open problems in condensed-matter physics—the glass transition, which is also intimately related to entropic effects (41–43).

As a representative example, we study here a bidisperse mixture of soft disks. This system exhibits a jamming transition at high densities (44). Several works have attempted to identify the jamming transition of this system, using dynamic properties such as the jamming length scale or the effective viscosity (45) and using static properties such as pair correlations or fraction of jammed particles (44, 45). Recently, Zu et al. (17) tried to measure the entropic signature of the jamming transition and have shown that compression-based methods have failed to do so. The authors of ref. 17 have generously shared their dataset with us, to test our method on, which we do below.

The system is an equimolar bidisperse system of disks with one-sided harmonic interactions (Fig. 4*A*). The simulation is performed in a finite box with periodic boundary conditions. The area density of the particles, ϕ, is a control parameter which is changed by changing the number of particles, N. Further details about the simulation are given in *SI Appendix*, section 6. The system is expected to undergo a jamming transition at $\phi_J \approx 0.841$ (28, 45).

There are a few differences between this system and the spin models discussed above. First, it is not a lattice system with discrete states. Rather, here the state space is continuous, parameterized by the positions of the particles. This requires a careful treatment of the discretization scheme. The choice of discretization scheme, and specifically the spatial resolution of discretization, affects the results in a nontrivial manner. Finally, in the analysis of the spin models we employed MICE on subsystems of all sizes, between one spin and the whole system. However, the soft disk systems are so large that doing so will be both impractical and unnecessary (adequate resolution requires $\sim 3 \times 10^6$ pixels, as discussed below). Before describing the results, we briefly discuss how these challenges are resolved, since they are common to many physical systems of interest, both in and out of equilibrium.

Continuous Systems (Differential Entropy). Since the system is continuous, the summation in Eq. **1** should be replaced by integration:

$$\tilde{S} = -\int p(x) \log p(x)\, dx. \qquad [7]$$

This definition is known as differential entropy. Note that $\log p(x)$ is ill defined since it depends on the choice of units of x in a nonmultiplicative manner.

This nonmultiplicative component, which depends logarithmically on the length unit, is fundamentally related to the fact that the digital representation of the system is discrete and thus the differential entropy of Eq. **7** differs from the discrete entropy of Eq. **1** by a factor that diverges logarithmically with the resolution of the discretization. For a detailed derivation see *SI Appendix*, section 7.

Moreover, we also show there that, quite conveniently, the representation of S in terms of Eq. **4** offers a well-defined way to remove this divergence. While $\tilde{S}$ of a continuous system depends logarithmically on the resolution, $\mathcal{M}$ becomes independent of it

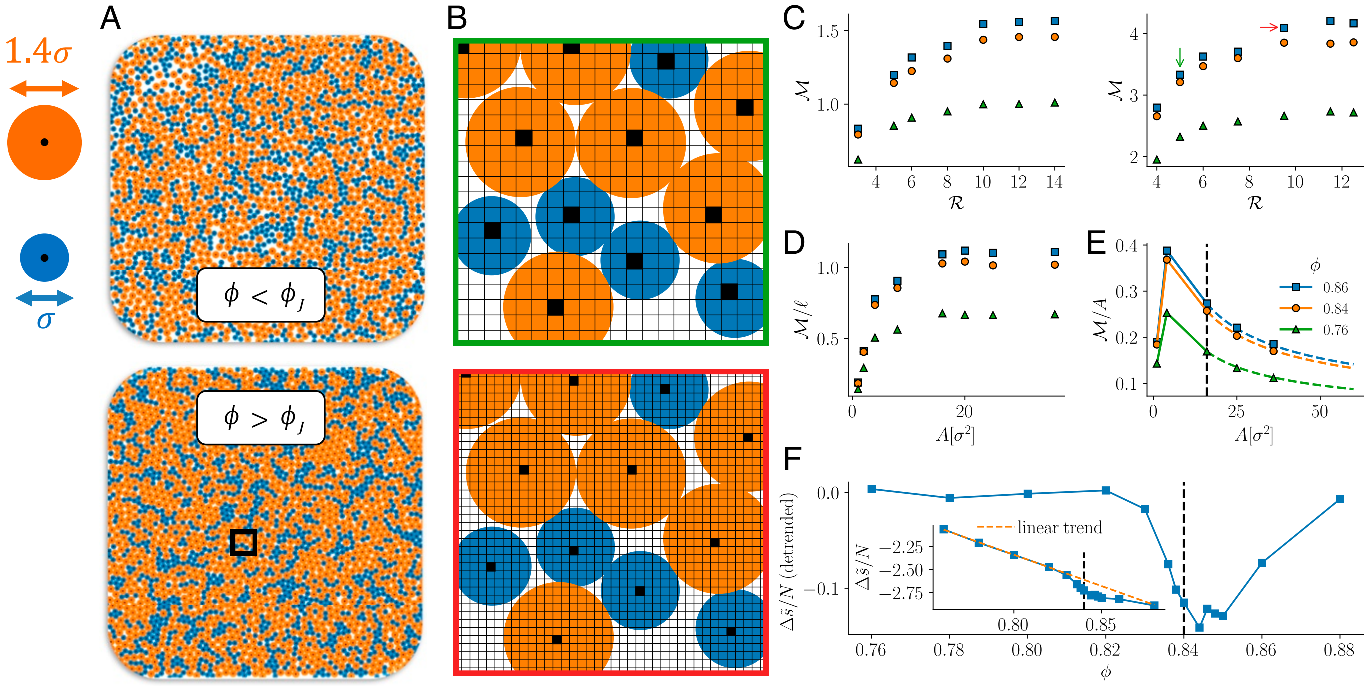


Fig. 4. (*A*) Snapshots from the bidisperse mixture simulation below and above the jamming transition density (ϕ_J). (*B*) A blowup of the marked region in *A*. We discretized the system (colored circles) as Boolean 2D images (black and white pixels). *Top* and *Bottom* show a spatial resolution of $\mathcal{R} = 5$ and $\mathcal{R} = 9.5$, respectively. The pixels are the input to MICE. (*C*) The effect of discretizing with various resolutions ($\mathcal{R}$) and various densities: $\mathcal{M}$ between two subsystems of size $2\sigma \times 1\sigma$ (*Left*) and $4\sigma \times 2\sigma$ (*Right*). At high resolutions, $\mathcal{M}$ becomes independent of $\mathcal{R}$. Green and red arrows indicate the resolutions represented in *B*, *Top* and *Bottom*, respectively. Different markers correspond to different densities; see key in *E*. (*D*) $\mathcal{M}/\ell$ as a function of the area of the subsystem (*A*) at various densities; see key in *E*. For large enough ℓ, $\mathcal{M}$ becomes linear in ℓ. (*E*) $\mathcal{M}/A$ as a function *A* at various densities. $\mathcal{M}$ becomes negligible for large subsystems. The dashed colored lines represent the extrapolation of Eq. **9**, based on the subsystem at the size represented by the black dashed line. (*F*) The density dependence of the excess entropy. *Inset* shows the results of MICE (blue) and the linear trend of $\tilde{s}/N$ at low densities (dashed orange line). For visual clarity, the linear trend in ϕ is subtracted in the main panel. The dashed black line represents the theoretical jamming transition point.

in the limit of very fine resolution. In fact, the necessary resolution is such that no physically relevant information is lost by the discretization, i.e., when all continuous configurations that map to the same discrete representation are equiprobable.

Therefore, when we estimate S according to Eq. **4**, we can avoid the logarithmic divergence simply by omitting the first term in the right-hand side. That is, in what follows we do not present $\tilde{s}$ but rather

$$\Delta\tilde{s} \equiv \tilde{s} - \frac{S(X_m)}{V_m} = -\sum_{k=1}^{m} \frac{\mathcal{M}(X_k)}{2V_k}. \qquad \textbf{[8]}$$

As a side note, we remark that the omitted term, $S(X_m)/V_m$, is simply the entropy density of the smallest subsystem. It corresponds to the entropy of an "ideal gas" composed of copies of the smallest subsystem. Subtracting the entropy of an ideal gas is common in entropy calculations of thermodynamic systems (17, 39). The result of the subtraction is commonly referred to as "excess entropy."

Discretization. Since convolutional ANNs show state-of-the-art capabilities in extracting information from images, we discretize phase space by mapping a state of the system to a 2D image, whose pixels are black if they contain a center of a particle (Fig. 4*B*).[†] The spatial resolution of the image is a hyperparameter of our method. We measure the resolution with the dimensionless number $\mathcal{R} = \sigma/p$, where p is the spatial extent of a pixel and σ is the diameter of the smaller disk. Based on the discussion above, we expect the estimation of $\mathcal{M}$ to converge to a constant value when $\mathcal{R}$ is increased. This is indeed the case, as demonstrated in Fig. 4*C*. In what follows, we use $\mathcal{R} = 10$, for which $\mathcal{M}$ is converged. We note that in terms of resources, the computational cost of discretizing the system is negligible compared to simulating the system or training the ANN. In addition, as shown below, the ANN does not have to be applied on the whole system, so a fine discretization does not lead to a memory bottleneck, at least not in 2D.

Extrapolating the Mutual Information. The resolution required for convergence necessitates $\sim 10^6$ pixels to discretize the whole system. Feeding such a large image to an ANN might be possible, but requires unreasonable computational resources for the task at hand. Luckily, this is not necessary.

As discussed above, for large enough subsystems, that is, scales much larger than the longest correlation length of the system, we expect $\mathcal{M}$ to grow linearly with the interface length (Fig. 2*C*). In precise terms, we expect

$$\mathcal{M}(X_k) = \frac{\ell_k}{\ell_n} \mathcal{M}(X_n). \qquad \textbf{[9]}$$

If we assume this is obeyed for all systems larger than X_k, this relation can be used to replace the summands in Eq. **4**, and the summation can be done analytically without calculations on subsystems larger than X_k. Fig. 4*D* shows that this happens for subsystems of length $\sim 4\sigma$. In Fig. 4*E* we show that Eq. **9**, based on the values of $\mathcal{M}$ for this size, quantitatively reproduces the values of the summands of Eq. **4** for sizes larger than 4σ, i.e., a 2D volume of $A = 16\sigma^2$.

Results. We are now in position to calculate the entropy of the whole system for various densities. Assuming that Eq. **9** is satisfied for $n > m$, Eq. **4** can be analytically summed, yielding (*SI Appendix*, section 8)

$$s = s(x_m) - 2\frac{\mathcal{M}(X_m)}{V_m}. \qquad \textbf{[10]}$$

Fig. 4 *F*, *Inset* shows $\Delta\tilde{s}/N$ as a function of ϕ. It is seen that at low densities $\Delta\tilde{s}$ depends roughly linearly on the density (dashed orange line). To emphasize the phase transition, in the main panel of Fig. 4*F* we plot the same data with this linear trend subtracted. The change in the behavior of $\Delta\tilde{s}$ around the expected jamming point is evident. Importantly, we remind the reader that compression-based entropy estimations were less successful in showing this transition (section 3.5 of ref. 17). A more detailed comparison with the results of ref. 17 is given in *SI Appendix*, section 9.

Discussion and Conclusion

Machine-learning algorithms in general, and neural networks in particular, offer an effective tool to identify patterns in high-dimensional data with complex correlation structure. We have shown that these capabilities can be leveraged to tackle another important challenge—computing the entropy of physical systems.

The crux of the method is mapping the problem of entropy calculation to an iterative process of mutual information estimation. By doing so we were able to estimate the entropy of canonical statistical physics problems, both discrete and continuous, both in and out of equilibrium, outperforming compression-based entropy estimation methods. Finally, we demonstrated that MICE naturally allows us to decompose the entropy into contributions from different scales, providing an insightful diagnostic for the thermodynamics of physical systems.

We surmise that MICE could be a promising tool for the study of many important systems, such as the configurational entropy of amorphous solids (46), the entropy crisis of glassy systems (42), entropy of active matter (40), and more. The main limit of the proposed method would depend on the minimal system size for which Eq. **9** applies, which determines the largest input for which an ANN should be trained. This is the dominant factor in the computational cost of our method. In addition, we believe that with adequate modifications MICE could be used on quantum systems, for which the mutual information is fundamentally related to entanglement of quantum states (47). A relevant direction could be the extraction of entropy from quantum Monte Carlo simulations. These directions will be explored in future works.

Data Availability. All study data are included in this article and *SI Appendix*.

ACKNOWLEDGMENTS. We thank Daan Frenkel, Mengjie Zu, and Arunkumar Bupathy for fruitful discussions and for generously sharing their data and code. In addition, we thank Yuval Binyamini, Yakov Kantor, Haim Diamant, Gil Ariel, and Amit Moscovich-Eiger for fruitful discussions. We acknowledge support by the Israel Science Foundation (550/15, 154/19), the United States–Israel Binational Science Foundation (201696), and Army Research Office (W911NF-20-1-0013). Y.B.-S. also thanks his mother.

[†] Technically, pixels are black if they contain a center of one or more particles, although this never happens in the resolutions we work with.

1. M. Kardar, *Statistical Physics of Fields* (Cambridge University Press, 2007).
2. P. G. De Gennes, J. Prost, *The Physics of Liquid Crystals* (Oxford University Press, 1993), vol. 83.
3. D. Frenkel, Entropy-driven phase transitions. *Phys. Stat. Mech. Appl.* **263**, 26–38 (1999).
4. M. C. Cross, P. C. Hohenberg, Pattern formation outside of equilibrium. *Rev. Mod. Phys.* **65**, 851–1112 (1993).
5. R. Asor, O. Ben-nun Shaul, A. Oppenheim, U. Raviv, Crystallization, reentrant melting, and resolubilization of virus nanoparticles. *ACS Nano* **11**, 9814–9824 (2017).

6. Y. S. Cho *et al.*, Self-organization of bidisperse colloids in water droplets. *J. Am. Chem. Soc.* **127**, 15968–15975 (2005).
7. A. Donev *et al.*, Improving the density of jammed disordered packings using ellipsoids. *Science* **303**, 990–993 (2004).
8. R. Avinery, M. Kornreich, R. Beck, Universal and accessible entropy estimation using a compression algorithm. *Phys. Rev. Lett.* **123**, 178102 (2019).
9. M. C. Baxa, E. J. Haddadian, J. M. Jumper, K. F. Freed, T. R. Sosnick, Loss of conformational entropy in protein folding calculated using realistic ensembles and its implications for NMR-based calculations. *Proc. Natl. Acad. Sci. U.S.A.* **111**, 15396–15401 (2014).
10. G. P. Brady, K. A. Sharp, Entropy in protein folding and in protein–protein interactions. *Curr. Opin. Struct. Biol.* **7**, 215–221 (1997).
11. D. J. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, 2003).
12. C. Kittel, H. Kroemer, *Thermal Physics* (American Association of Physics Teachers, 1998).
13. D. Frenkel, Simulations: The dark side. *Eur. Phys. J. Plus* **128**, 10 (2013).
14. N. Hansen, W. F. Van Gunsteren, Practical aspects of free-energy calculations: A review. *J. Chem. Theor. Comput.* **10**, 2632–2647 (2014).
15. C. Jarzynski, Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.* **78**, 2690 (1997).
16. S. Piana, K. Lindorff-Larsen, D. E. Shaw, Protein folding kinetics and thermodynamics from atomistic simulation. *Proc. Natl. Acad. Sci. U.S.A.* **109**, 17845–17850 (2012).
17. M. Zu, A. Bupathy, D. Frenkel, S. Sastry, Information density, structure and entropy in equilibrium and non-equilibrium systems. *J. Stat. Mech. Theor. Exp.* **2020**, 023204 (2020).
18. S. Martiniani, P. M. Chaikin, D. Levine, Quantifying hidden order out of equilibrium. *Phys. Rev. X* **9**, 011031 (2019).
19. C. E. Shannon, A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 623–656 (1948).
20. A. Kolmogorov, New metric invariant of transitive dynamical systems and endomorphisms of Lebesgue spaces. *Doklady Russian Acad. Sci.* **119**, 861–864 (1958).
21. J. Ziv, A. Lempel, A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory* **23**, 337–343 (1977).
22. G. H. Wannier, Antiferromagnetism. The triangular Ising net. *Phys. Rev.* **79**, 357–364 (1950).
23. L. Onsager, Crystal statistics. I. A two-dimensional model with an order-disorder transition. *Phys. Rev.* **65**, 117–149 (1944).
24. J. F. Yu, Z. Y. Xie, T. Xiang, Two-dimensional classical XY model by HOTRG *Phys. Rev. E.* **89**, 013308. (2014).
25. I. Belghazi *et al.*, "Mine: Mutual information neural estimation" in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy, A. Krause, Eds. (PMLR, 2018), vol. 80, pp. 531–540.
26. M. D. Donsker, S. S. Varadhan, Asymptotic evaluation of certain Markov process expectations for large time. IV. *Commun. Pure Appl. Math.* **36**, 183–212 (1983).
27. D. P. Landau, K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, 2014).
28. C. S. O'Hern, L. E. Silbert, A. J. Liu, S. R. Nagel, Jamming at zero temperature and zero applied stress: The epitome of disorder. *Phys. Rev. E* **68**, 011306 (2003).
29. J. Kent-Dobias, J. P. Sethna, Cluster representations and the Wolff algorithm in arbitrary external fields. *Phys. Rev. E* **98**, 063306 (2018).
30. A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks" in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, Eds. (Curran Associates, Inc., 2012), pp. 1097–1105.
31. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
32. A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library" in *Advances in Neural Information Processing Systems 32*, H. Wallach *et al.*, Eds. (Curran Associates, Inc., 2019), pp. 8024–8035.
33. S. J. Pan, Q. Yang, A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**, 1345–1359 (2010).
34. P. Ronhovde, Z. Nussinov, Multiresolution community detection for megascale networks by information-based replica correlations. *Phys. Rev. E* **80**, 016109 (2009).
35. Z. Nussinov *et al.*, "Inference of hidden structures in complex physical systems by multi-scale clustering" in *Information Science for Materials Discovery and Design*, T. Lookman, F. J. Alexander, K. Rajan, Eds. (Springer, 2016), pp. 115–138.
36. J. Wilms, M. Troyer, F. Verstraete, Mutual information in classical spin models. *J. Stat. Mech. Theory Exp.* **2011**, P10011 (2011).
37. J. Iaconis, S. Inglis, A. B. Kallin, R. G. Melko, Detecting classical phase transitions with Renyi mutual information. *Phys. Rev. B* **87**, 195134 (2013).
38. M. M. Wolf, F. Verstraete, M. B. Hastings, J. I. Cirac, Area laws in quantum systems: Mutual information and correlations. *Phys. Rev. Lett.* **100**, 070502 (2008).
39. G. Ariel, H. Diamant, Inferring entropy from structure. *Phys. Rev. E.* **102**, 022110 (2020).
40. C. Nardini *et al.*, Entropy production in field theories without time-reversal symmetry: Quantifying the non-equilibrium character of active matter. *Phys. Rev. X* **7**, 021007 (2017).
41. A. J. Liu, S. R. Nagel, The jamming transition and the marginally jammed solid. *Annu. Rev. Condens. Matter Phys.* **1**, 347–369 (2010).
42. A. Cavagna, Supercooled liquids for pedestrians. *Phys. Rep.* **476**, 51–124 (2009).
43. R. Monasson, Structural glass transition and the entropy of the metastable states. *Phys. Rev. Lett.* **75**, 2847–2850 (1995).
44. D. Koeze, D. Vågberg, B. Tjoa, B. Tighe, Mapping the jamming transition of bidisperse mixtures. *EPL* **113**, 54001 (2016).
45. D. Vågberg, D. Valdez-Balderas, M. A. Moore, P. Olsson, S. Teitel, Finite-size scaling at the jamming transition: Corrections to scaling and the correlation-length critical exponent. *Phys. Rev.* **83**, 030303 (2011).
46. E. Bouchbinder, J. Langer, I. Procaccia, Athermal shear-transformation-zone theory of amorphous plastic deformation. I. Basic principles. *Phys. Rev.* **75**, 036107 (2007).
47. L. Amico, R. Fazio, A. Osterloh, V. Vedral, Entanglement in many-body systems. *Rev. Mod. Phys.* **80**, 517 (2008).

Supplementary Information for

Machine-learning Iterative Calculation of Entropy for Physical Systems

Amit Nir, Eran Sela Roy Beck, and Yohai Bar Sinai

Corresponding Authors
Roy Beck, `roy@tauex.tau.ac.il`
Yohai Bar-Sinai, `ybarsinai@gmail.com`

This PDF file includes:

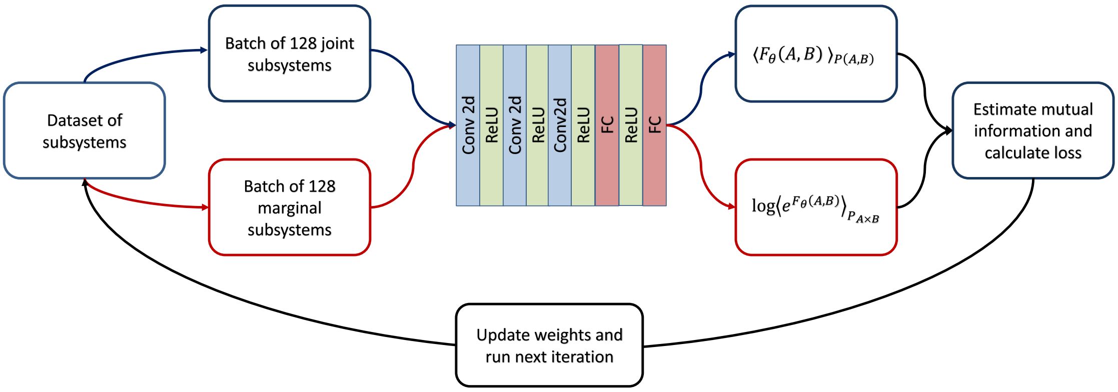
Supplementary text
Figs. S1 to S6
SI References

Supporting Information Text

1. *MICE* implementation details

A. Data preprocessing and augmentation. Input features were normalized between values -1 and 1. For the soft disk system, this means that empty pixels are set to -1 and pixels which contain a particle center are set to 1. Since all our systems are symmetric under reflections, we performed data augmentation by reflecting both vertically and horizontally. In the data of the XY model without an external field, a global random phase was also used for data augmentation. In addition, due to translational symmetry one can sample subsystems anywhere within the larger system. Combining all these, a single snapshot of 64x64 spins can generate about 15,000 training samples.

B. Network Architecture. Our method was implemented using the PyTorch library (1). For subsystems of input size larger than 32×32 we used three convolutional layers with 16 filters of size 3×3 each and a rectified linear unit (ReLU) activation. For smaller subsystems, we use only two convolutional layers. For subsystems of size 4×4 or smaller, only one convolutional layer is used. The convolutional layers are followed by two fully connected layers, with $\frac{k}{2}$ and 1 output neurons, respectively, where k is the number of output neurons in the last convolutional layer. The batch size for training was 128.



Dataset of subsystems
Batch of 128 joint subsystems
Batch of 128 marginal subsystems
Conv 2d
ReLU
Conv 2d
ReLU
Conv2d
ReLU
FC
ReLU
FC
$\langle F_\theta(A,B)\rangle_{P(A,B)}$
$\log\langle e^{F_\theta(A,B)}\rangle_{P_{A\times B}}$
Estimate mutual information and calculate loss
Update weights and run next iteration

Fig. S1. The flow of *MICE*. The simulations are used to generate a marginal dataset and a joint dataset (see main text for definition) dataset. The specific architechture of the ANN shown here was used for subsystem pars larger than 32×32. Smaller subsystems used $1 - 2$ convolutional layers, as detailed in Sec. 1B.

C. Noise Reduction. The output of the neural network (ANN) is averaged over the marginal and joint distributions to give a bound on the mutual information (see Eq. (5) of the main text). As the network learning process progresses, the bound becomes tighter. However, at each iteration the averaging is performed over a small batch of 128 samples. Therefore, the network's output is extremely noisy. To smooth the results we use a moving exponential average:

$$\langle \mathcal{M} \rangle_{i+1} = \langle \mathcal{M} \rangle_i + \alpha \Big(\mathcal{M}_{i+1} - \langle \mathcal{M} \rangle_i \Big). \qquad \text{[S1]}$$

where $\mathcal{M}_j$ is the output of the network after j optimization iterations, and $\langle \mathcal{M} \rangle_i$ is our averaged estimation after i iterations, see Fig. S2. Throughout the manuscript we used the exponential averaging with $\alpha = 10^{-3}$.

D. Validation. For estimating $\mathcal{M}$ we implemented the standard scehme of using a validation set. Two independent datasets with ratio of 80-20 were created before training. The network was trained over the large (training dataset), and the training phase was terminated when the $\mathcal{M}$ estimation on the training set stopped increasing. $\mathcal{M}$ was estimated over the independent validation set as well, and this value was used for subsequent calculations. By comparing the estimation of $\mathcal{M}$ over the training and validation sets, one can verify that the network did not overfit the data.

E. Dataset size. For the spin models we used a dataset of 5000 samples of a 64×64 system. An exception is the XY model with an external field where we used 2000 simulations. For the soft disk system we used a set of 100 simulations. In general, for the systems considered in the manuscript we typically needed about $10^4 - 10^5$ samples (obtained from the the $\sim 10^3$ actual samples by data augmentation, see A above) to achieve reasonable convergence.

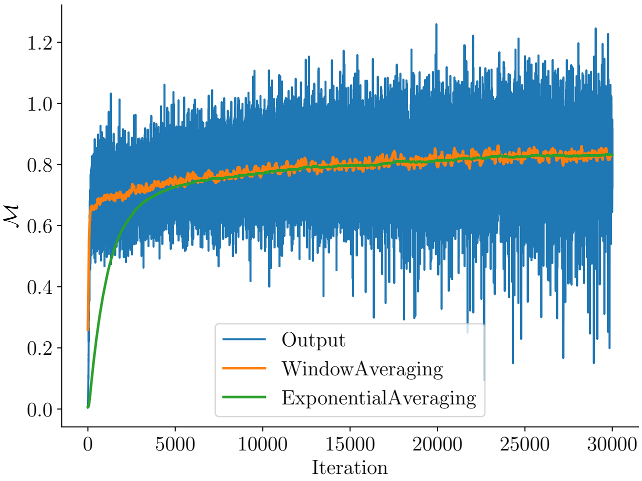


Fig. S2. Noise reduction. The raw output of the network (blue) and an exponential average with $\alpha = 10^{-3}$ (green) are shown during a typical training loop. In addition, we demonstrate another noise reduction method, used by the original authors of (2), a moving average with a window size of 100 iterations (orange).

F. Transfer Learning. When initiating the network weights at random the resulting estimation of $\mathcal{M}$ is roughly zero. During training it increases until a plateau is reached. For our choice of hyperparameters this can take a few thousand training iterations, cf. Fig. S3. This process can be expedited if the network is not initialized at a random initial condition but instead the weights of a network that was trained for a different system are used, a technique called "Transfer Learning"

This can be done in a number of ways - e.g. transfer learning across temperatures or the sizes of the subsystem. In the main text we only used transfer learning across different temperatures. In Fig. S3 we show the result of training with and without transfer learning, which can reduce training time by 1-2 orders of magnitude. We note that transfer learning works better when we first train on high T and then transfer to lower T, similar to simulated annealing strategy in optimization.

We note that transfer learning across subsystem size is slightly more tricky since the input size to the ANN is different. One simple-minded way to overcome this is to pad the smaller subsystems with zeros, which gives reasonable results, cf. Fig. S3B. This is an interesting direction for future research, which we did not further explore. Transfer learning across subsystem size was not used in the main text.

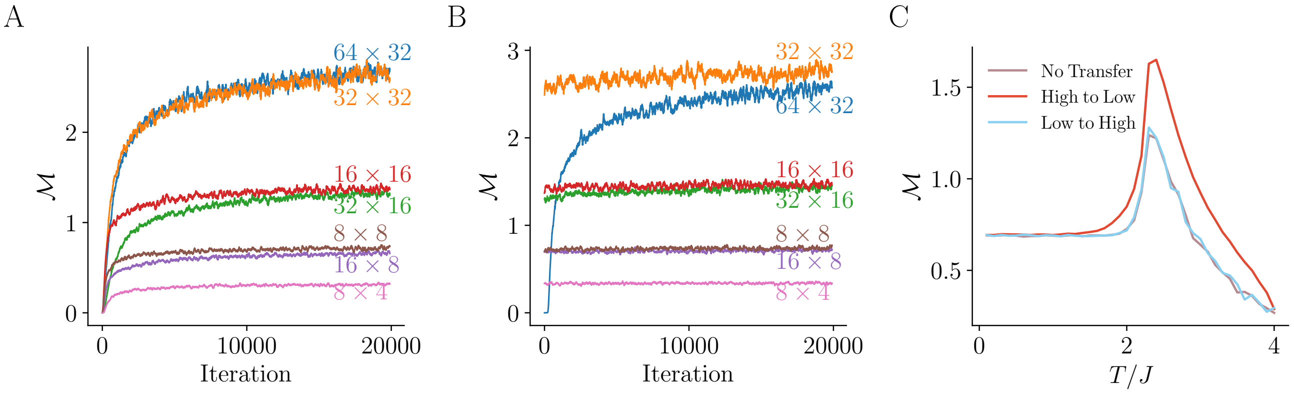


Fig. S3. Effect of transfer learning. (A)-(B) Learning process as function of iteration for various subsystem sizes. (A) Without transfer learning (i.e. random initial weights for each ANN). (B) With transfer learning from one subsystem size to another. $\mathcal{M}$ plateaus at the same level with or without transfer learning, but the number of iterations needed to reach the plateau changes drastically. (C) $\mathcal{M}$ as function of temperature for 16×16 subsystem of the 2d ferromagnetic Ising model. Adding transfer learning from high to low temperature improves the results dramatically while transfer learning in the opposite direction is not effective. All trainings were done for 3000 iterations at every temperature.

2. Spin Model Simulations

Sampling the distribution of the Ising systems was preformed using standard Monte-Carlo sampling.

Sampling the distribution of the XY simulation was performed using the Wolff algorithm implemented in the c++ library provided in Ref. (3). To generate uncorrelated samples the mean cluster size at each temperature, c, was calculated and the simulation was sampled every $2/c$ steps. That is, each spin was flipped twice on average between two subsequent samples at all temperatures.

3. Specific Heat Estimation Using *MICE*

A standard method of estimating the entropy of thermodynamic systems is to integrate the specific heat from low temperatures. This method relies on the relations

$$c_V = T\frac{dS}{dT}\ , \qquad \text{and} \tag{S2}$$

$$c_V = \frac{\langle E^2\rangle - \langle E\rangle^2}{T^2}\ , \tag{S3}$$

where c_V is the heat capacity, E is the energy and $\langle\cdot\rangle$ denotes thermal averaging. $S(T)$ can be calculated using Eq. (S3) and integrating the energy fluctuations from zero temperature to T.

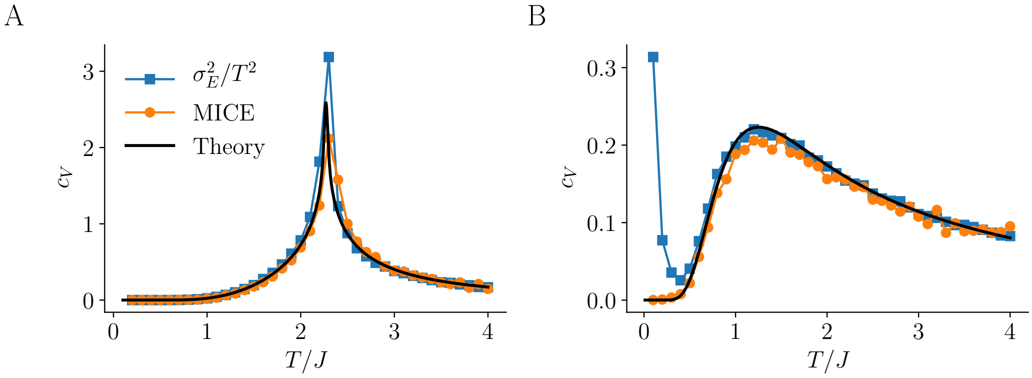


Fig. S4. Estimating c_V using energy fluctuation estimation, (Eq. (S3), blue), and *MICE* (Eq. (S2), orange), compared to the theoretical value (black). (A) 2D Ising model (B) 2D anti-ferromagnetic Ising model.

Alternatively, one take the inverse direction: using the estimation of S, as calculated by *MICE*, together with Eq. (S2) to estimate c_V. In Fig. S4 we compare this estimation of c_V (orange) to the estimation of c_V using energy fluctuations (Eq. (S3), blue). It is evident that the energy fluctuations overestimate c_V in the 2D ferromagnetic Ising model near the phase transition, and at low temperatures in the anti-ferromagnetic triangular lattice, which features high degeneracy of low energy states.

4. Analytic Calculation of $\mathcal{M}$ at high and low temperature limit for spin models

A. High temperature. Here we derive Eq. (6) of the main text by a rigorous high-T expansion of the partition function and marginal probabilities. Physically this expansion relies on the fact that at high temperatures correlations become local. At high temperature we explicitly obtain the area law, $\mathcal{M}(A,B) \propto \ell$, stating that $\mathcal{M}$ is proportional to the area ℓ (or length in two dimensions) of the interface between regions A and B, rather their volume.

The mutual information between subsystems A and B, whose union is $A \cup B = X$, is defined as:

$$\mathcal{M}(A,B) = S(A) + S(B) - S(X), \tag{S4}$$

where the entropy of a subsystem A is given in terms of the marginal probability:

$$S(A) = -\sum_{\alpha} P_A(\alpha)\log P_A(\alpha). \tag{S5}$$

Here, α labels microstates of A. For the spins models, the microstates are given in terms of the configurations of spins $z_a, a \in A$. We assume that the entire system X under consideration is described by an equilibrium distribution:

$$P_X(\boldsymbol{z}) = \frac{e^{-\beta E(\boldsymbol{z})}}{Z}\ , \qquad\qquad Z = \sum_{\{z_i=\pm1\}} e^{-\beta E(\boldsymbol{z})}\ . \tag{S6}$$

Here and in what follows boldface letters (e.g. $\boldsymbol{z}$) denote vectors. The marginal distribution of subsystem A is obtained by tracing out the spins in its complement, $P_A = \mathrm{Tr}_B\, P_X$.

We proceed by an explicit evaluation of $\mathcal{M}$ at high temperature for the Ising model:

$$E_{\text{Ising}}(\boldsymbol{z}) = -J\sum_{\langle i,j\rangle} z_i z_j - H\sum_i z_i\ , \qquad\qquad z_i = \pm 1\ . \tag{S7}$$

The expansion of the partition function in powers of β up to second order is

$$\begin{aligned} Z &= \sum_{\{z_i=\pm1\}} \left(1-\beta E(\boldsymbol{z}) + \frac{1}{2}\beta^2 E(\boldsymbol{z})^2 + \dots\right) = 2^N + \frac{1}{2}\beta^2 \left[J^2 \left(\sum_{\langle i,j\rangle} 1\right) 2^N + H^2 \left(\sum_i 1\right) 2^N \right] + \mathcal{O}\left(\beta^3\right) \\ &= 2^N \left[1 + \frac{1}{2}\beta^2 \left(J^2 N_{\text{links}} + H^2 N\right)\right] + \mathcal{O}\left(\beta^3\right), \end{aligned} \qquad \text{[S8]}$$

where $\sum_{\langle i,j\rangle} 1 = N_{\text{links}}$ is the total number of links and $\sum_i 1 = N$ is the number of sites. In what follows we omit the external field (H) for clarity and conciseness of presentation, and only mention its effect in the end result.

Next, we perform a high temperature expansion up to order β^2 of the marginal probability

$$P_A(\boldsymbol{z}_A) = \sum_{\boldsymbol{z}_b} P(\boldsymbol{z}_A, \boldsymbol{z}_B) = \sum_{\boldsymbol{z}_B} \frac{1-\beta E(\boldsymbol{z}_A,\boldsymbol{z}_B) + \frac{1}{2}\beta^2 E^2(\boldsymbol{z}_A,\boldsymbol{z}_B)}{Z} + \mathcal{O}\left(\beta^3\right) . \qquad \text{[S9]}$$

Here $\boldsymbol{z}_A$ is fixed and spins $\boldsymbol{z}_B$ in B act like an environment for A and are traced out.

Tracing out the first order term in the numerator of Eq. (S9) annihilates any terms that involve at least one spin in B. Therefore, the first order term yields simply the energy of subsystem A,

$$E_A(\boldsymbol{z}_A) = -J \sum_{\langle a,a'\rangle \in A} z_a z_{a'} . \qquad \text{[S10]}$$

Tracing over the second order term in the numerator of Eq. (S9) involves a double sum over neighbors $\sum_{\langle ij\rangle} \sum_{\langle i'j'\rangle} z_i z_j z_{i'} z_{j'}$. The only combinations of i, j, i', j' that are not annihilated by tracing out are:

1. $i, j, i', j' \in A$. Summation over these quadruplets yields $E_A(\boldsymbol{z}_A)^2$.

2. $i, j, i', j' \in B$. Summation over these quadruplets yields $J^2 N^B_{\text{links}}$ where N^B_{links} is the number of links in B.

3. $i \in A, j \in B$ and $\langle i,j\rangle = \langle i', j'\rangle$. Summation over these quadruplets yields $J^2\ell$ where ℓ is the number of links between A and B.

4. In the triangular lattice there's a fourth option where there exist two distinct spins $i, i' \in A$ which have a common neighbor $j \in B$. The sum over such pairs of spins in A is denoted $\sum'_{aa'}$.

Therefore, the numerator of Eq. (S9) yields, to second order in β,

$$P_A(\boldsymbol{z}_a) = 2^{N_B} \frac{1-\beta E_A(\boldsymbol{z}_A) + \frac{1}{2}\beta^2 \left(E_A(\boldsymbol{z}_a)^2 + J^2 N^B_{\text{links}} + J^2\ell + J^2 \sum'_{aa'} z_a z_{a'}\right)}{Z} + \mathcal{O}\left(\beta^3\right) . \qquad \text{[S11]}$$

Proceeding with the expansion, plugging in Eq. (S8) and using $N^A_{\text{links}} + N^B_{\text{links}} + \ell = N_{\text{links}}$, we get

$$P_A(\boldsymbol{z}_a) = \frac{1-\beta E_A(\boldsymbol{z}_a) + \frac{1}{2}\beta^2 E_A(\boldsymbol{z}_a)^2 + \frac{1}{2}\beta^2 J^2 \sum'_{aa'} z_a z_{a'}}{Z_A} + \mathcal{O}\left(\beta^3\right) \text{, with} \qquad \text{[S12]}$$

$$Z_A = 2^{N^A}\left[1 + \frac{1}{2}\beta^2 J^2 N^A_{\text{links}}\right] + \mathcal{O}\left(\beta^3\right) . \qquad \text{[S13]}$$

Eq. (S12) has the form of a Boltzmann distribution (note the similarity of Eq. (S13) to Eq. (S8)) derived from the Hamiltonian E_A, with extra couplings generated by the tracing out of B (the last term in the numerator of Eq. (S12)). A straightforward but tedious calculation, which will not be detailed here, shows that up to quadratic order in β these couplings do not affect the entropy. That is, while they do clearly affect the probabilities of individual states (as explicitly shown in Eq. (S12)) their combined contribution to S cancels out to quadratic order when summed over all states. Therefore, as far as entropy calculations are concerned we can write

$$P_A(\boldsymbol{z}_A) \approx \frac{e^{-\beta E_A(S_a)}}{Z_A} + \mathcal{O}\left(\beta^3\right) , \qquad Z_A = \sum_{\boldsymbol{z}_A} e^{-\beta E_A(\boldsymbol{z}_A)} + \mathcal{O}\left(\beta^3\right) , \qquad \text{[S14]}$$

and treat P_A as a standard Boltzmann distribution, for which we have $S = \partial_T(T \log Z)$. Plugging this into Eq. (S4) gives

$$\mathcal{M}(A,B) = \partial_T \left(T \log \frac{Z_A Z_B}{Z_X}\right) + \mathcal{O}\left(\beta^3\right) . \qquad \text{[S15]}$$

Physically the numerator $(Z_A Z_B)$ is the partition function for all the spins in X without the interactions through links connecting A and B. Finally, using Eq. (S8) and Eq. (S13) we obtain the result

$$\mathcal{M}_{\text{Ising}}(A,B) = \frac{1}{2}\left(\frac{J}{T}\right)^2 \ell + \mathcal{O}\left(\beta^3\right) . \qquad \text{[S16]}$$

Note that neither the sign of J nor the lattice symmetry (square versus triangular) influence the answer to order β^2 – the only relevant parameters are the number of links connecting the two subsystems ℓ and the coupling constant J. Also, up to this order the magnetic field H does not contribute. A very similar calculation leads to the same form for the XY model, with only a change in the prefactor:

$$\mathcal{M}_{\mathrm{XY}}(A,B)=\frac{1}{4}\left(\frac{J}{T}\right)^2\ell+\mathcal{O}\left(\beta^3\right)\ . \tag{S17}$$

Both Eq. (S16) and Eq. (S17) are valid also when A and B do not compose the whole system, but are a part of a larger system.

B. Low-temperature expansion - XY model in a magnetic field. Statistical mechanics problems of continuous variables can be treated at low temperatures via an harmonic treatment of the interactions, i.e. a mapping to a system of coupled harmonic oscillators. This technique can be applied to compute $\mathcal{M}$ too (4), yielding closed-form formulas. Here we apply this method to the XY model in an external magnetic field (H) in the zero-temperature limit.

The XY model in a magnetic field is defined by the partition function

$$Z=\int_0^{2\pi}d\boldsymbol{\theta}e^{-\beta E(\boldsymbol{\theta})},\qquad E(\boldsymbol{\theta})=-J\sum_{\langle i,j\rangle}\cos(\theta_i-\theta_j)-H\sum_i\cos\theta_i. \tag{S18}$$

At low temperature $T\ll J,H$ the variables $\boldsymbol{\theta}$ explore only the vicinity of the minimum of the external potential $-H\cos\theta_i$, and since we consider a frustration-free lattice (square lattice), also the differences $\theta_i-\theta_j$ on neighbouring links $\langle i,j\rangle$ will be located near the minima of $-J\cos(\theta_i-\theta_j)$. Performing a harmonic approximation of the overall potential we get:

$$Z_0=\int_{-\infty}^{\infty}d\boldsymbol{\theta}e^{-\beta E_0(\boldsymbol{\theta})}\ ,\qquad E_0(\boldsymbol{\theta})=\frac{J}{2}\sum_{\langle i,j\rangle}(\theta_i-\theta_j)^2+\frac{H}{2}\sum_i\theta_i^2+\mathrm{const}\ . \tag{S19}$$

Here, we extended the variables θ_i from being angles to unconstrained real numbers. Accordingly, microstates of the full system X satisfy a multivariate normal distribution

$$p(\boldsymbol{\theta})=\frac{e^{-\frac{1}{2}\boldsymbol{\theta}^TM\boldsymbol{\theta}}}{Z_0}\ ,\qquad \text{with}\qquad M_{ij}=\frac{H+zJ}{T}\delta_{ij}-\frac{J}{T}\delta_{\langle i,j\rangle}\ . \tag{S20}$$

M is the system's Hessian, a $N\times N$ matrix where N is the number of sites in the system X. Here z is the coordination number ($z=4$ for a square lattice) and $\delta_{\langle i,j\rangle}=1$ if i and j are neighbors and 0 otherwise. The entropy of a multivariate Gaussian is well known:

$$S(X)=\frac{N}{2}\log 2\pi e-\frac{1}{2}\log\det M. \tag{S21}$$

For a single spin in a magnetic field, for example, this gives $S=\log\left(\sqrt{2\pi eT/H}\right)$ which is valid as long as the variance of θ, $(T/H)^2$, is sufficiently small compared to $(2\pi)^2$.

The key object required for the calculation of the $\mathcal{M}$ is the marginal probability for a subsystem A. It is obtained by integrating $p(\boldsymbol{\theta})$ over all degrees of freedom $\boldsymbol{\theta}_B\in B$,

$$p_A(\boldsymbol{\theta}_A)=\frac{1}{Z}\int_0^{2\pi}d\boldsymbol{\theta}_Be^{-\beta E(\boldsymbol{\theta}_A,\boldsymbol{\theta}_B)}. \tag{S22}$$

To perform the Gaussian integral we decompose the matrix M as

$$M=\begin{pmatrix}M_{AA} & M_{AB}\\ M_{BA} & M_{BB}\end{pmatrix}, \tag{S23}$$

where, M_{AA} is an $N^A\times N^A$ matrix acting only on the N^A degrees of freedom in A, and similarly for M_{BB}. The off-diagonal blocks $M_{AB}=M_{BA}^T$ couple the two subsystems. Thus,

$$p_A(\boldsymbol{\theta}_A)=e^{-\frac{1}{2}\boldsymbol{\theta}_A^TM_{AA}\boldsymbol{\theta}_A}\int d\theta_B\exp\left[-\frac{1}{2}\boldsymbol{\theta}_B^TM_{BB}\boldsymbol{\theta}_B-\boldsymbol{\theta}_A^TM_{AB}\boldsymbol{\theta}_B\right]\ . \tag{S24}$$

Performing the Gaussian integral over $\boldsymbol{\theta}_B$ gives

$$P(\boldsymbol{\theta}_A)=\left((2\pi)^{N_B}\det M_{BB}\right)^{1/2}\exp\left[-\frac{1}{2}\boldsymbol{\theta}_A^TM_{AA}\boldsymbol{\theta}_A-\frac{1}{2}\boldsymbol{\theta}_A^T\left(M_{AB}M_{BB}^{-1}M_{BA}\right)\boldsymbol{\theta}_A\right]\ . \tag{S25}$$

Since the marginal distribution is also Gaussian, its entropy is given by Eq. (S21), with the effective Hessian (covariance matrix) of A given by Eq. (S24),

$$M_A^{\mathrm{eff}}=M_{AA}-M_{AB}M_{BB}^{-1}M_{BA}\ . \tag{S26}$$

$M_A^{\rm eff}$ contains direct interactions inside A, as well as new interactions $M_{AB}M_{BB}^{-1}M_{BA}$ generated by tracing out the environment B. We thus have

$$\mathcal{M} = \frac{1}{2}\log\frac{\det M_X}{\det M_A^{\rm eff}\det M_B^{\rm eff}}. \tag{S27}$$

Note that this expression gives the $T \to 0$ limit of $\mathcal{M}$ and is independent of T. Finite temperature corrections are not present in the harmonic approximation and start to appear when the variance of spins becomes of order 2π and deviations from the Gaussian distribution are sampled.

For the system described in the main text $\mathcal{M}$ was computed by evaluating the determinant in Eq. (S21) numerically using the effective covariance matrix Eq. (S26).

5. $\mathcal{M}$ between two XY-spins in a magnetic field

It is instructive to contrast the result in the main text for the $\mathcal{M}$ of the XY model with that for a system consisting of only two spins. This can be calculated exactly, and is shown in Fig. S5. At high temperature $\mathcal{M}$ decreases like $\mathcal{M} \to \frac{1}{4}\left(\frac{J}{T}\right)^2$, indicated by a dashed line in the right panel, as predicted by Eq. (S17). As $T \to 0$, we can see in the central panel a logarithmic divergence with T which is cut-of when $T \approx H$.

Indeed it is easy to derive from Eqs. (S20), (S26) and (S27) the zero temperature limit of $\mathcal{M}$,

$$\lim_{T\to 0} \mathcal{M}_{\text{two spins}} = \log\frac{H+J}{\sqrt{H(H+J)}}. \tag{S28}$$

As H increases, the cutoff of the logarithmic divergence occurs at higher temperatures, and the peak thus shifts to higher temperatures. Thus,the peak itself, as well as its H-dependence features, are already present in a two-spin system.

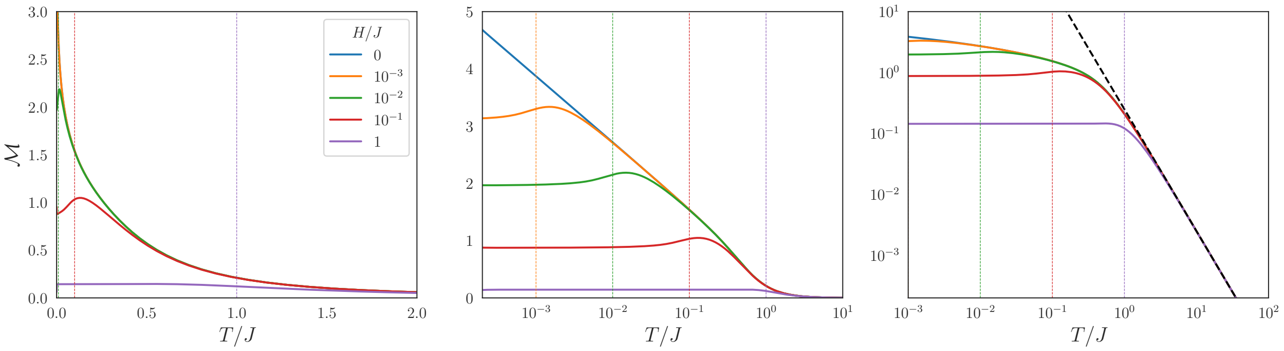


Fig. S5. Exact calculation of $\mathcal{M}$ for two XY spins ($J = 1$) in the presence of external field (H). The same data are shown in linear-linear, log-linear and log-log scales (some data of the middle panel appears also in the main text). Colored vertical dashed lines show $T = H$ with the color code corresponding to H as in the legend. The dashed black line in the right panel is the high temperature expansion limit of Eq. (S17).

6. Simulations of the soft sphere system

The system is an equimolar system of larger and smaller spheres. We choose the units such that the diameter of the smaller sphere is unity, and the radius of the larger one is 1.4. The dynamics were simulated using a fast inertial relaxation engine algorithm (5) in a square box of size 150 with periodic boundary conditions. 100 realizations were generated for each ϕ, ranging between 14,000 to 17,000 particles.

7. Discrete vs. differential entropy

As discussed in the main text, the system of bidisprese sphere is a continuous system, parameterized by a continuous vector $\boldsymbol{x} \in \mathbb{R}^{2N}$ where N is the number of particles in the system. However, the state of the system is provided to the ANN as a binary image, which is a discrete variable. Here we discuss the subtleties of comparing the discrete and continuous defintions of entropy (Eq. (1) and (7) of the main text, respectively).

Let us denote $p(\boldsymbol{x})$ the probability density of observing the configuration $\boldsymbol{x}$. The discretization is a mapping of the continuous vector $\boldsymbol{x}$ to an image $I(\boldsymbol{x})$ where I takes one of a finite set of values which we denote $I_1, I_2, \ldots$. Each I_i is associated with its pre-image Ω_i, observation probability p_i and phase-space volume v_i, defined as follows:

$$\Omega_i \equiv \{\boldsymbol{x} \mid I(\boldsymbol{x}) = I_i\}\ , \qquad p_i \equiv \int_{\Omega_i} p(\boldsymbol{x})d\boldsymbol{x}\ , \qquad v_i \equiv \int_{\Omega_i} 1\, d\boldsymbol{x}\ . \tag{S29}$$

In the limit of very fine discretization, i.e. $\max_i\{v_i\} \to 0$, and assuming $p(x)$ is not ill-behaved, the second definition can be approximated as

$$p_i \approx p(\boldsymbol{x}_i)v_i \ , \tag{S30}$$

where $\boldsymbol{x}_i$ is any point in Ω_i. This approximation is accurate when the discretization is fine enough such that p doesn't change considerably across Ω_i, i.e. when all configurations that are mapped to the same image are roughly equiprobable. When this happens, the differential entropy $\tilde{S}$ can be approximated by a Riemman sum:

$$\begin{aligned}\tilde{S} &= -\int p(\boldsymbol{x})\log p(\boldsymbol{x})d\boldsymbol{x} \approx -\sum_i \Big(p(\boldsymbol{x}_i)\log p(\boldsymbol{x}_i)\Big)\cdot v_i \\ &\approx -\sum_i \left(\frac{p_i}{v_i}\log\left(\frac{p_i}{v_i}\right)\right)\cdot v_i = \sum_i\left(-p_i\log p_i + p_i \log v_i\right) = S + \sum_i p_i \log v_i \ .\end{aligned} \tag{S31}$$

We see that $\tilde{S}$ differs from S by a term logarithmic in the resolution size. This term, however, cancels out when computing $\mathcal{M}$ rather than S.

To see this, let's say $\boldsymbol{x}$ and $\boldsymbol{y}$ are random variables, with the joint probability density $p(\boldsymbol{x},\boldsymbol{y})$ and marginal densities $p^x(\boldsymbol{x}) = \int p(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y}$ and $p^y(\boldsymbol{y}) = \int p(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}$. In addition, we have two discretization schemes $I^x(\boldsymbol{x})$ and $I^y(\boldsymbol{y})$ that map each observation to some finite set. We define, in analogy to Eq. (S29),

$$\begin{aligned}
&\Omega_{ij} \equiv \left\{(\boldsymbol{x},\boldsymbol{y}) \mid I^x(\boldsymbol{x}) = I^x_i \text{ and } I^y(\boldsymbol{x}) = I^y_j\right\} \ , && p_{ij} \equiv \int_{\Omega_{ij}} p(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x} \ , && v_{ij} \equiv \int_{\Omega_{ij}} 1\, d\boldsymbol{x}d\boldsymbol{y} \ , \\
&\Omega^x_i \equiv \{\boldsymbol{x} \mid I^x(\boldsymbol{x}) = I^x_i\} \ , && p^x_i \equiv \int_{\Omega^x_i} p^x(\boldsymbol{x})d\boldsymbol{x} \ , && v^x_i \equiv \int_{\Omega^x_i} 1\, d\boldsymbol{x} \ , \\
&\Omega^y_j \equiv \left\{\boldsymbol{y} \mid I^y(\boldsymbol{y}) = I^y_j\right\} \ , && p^y_j \equiv \int_{\Omega^y_j} p^y(\boldsymbol{y})d\boldsymbol{y} \ , && v^y_j \equiv \int_{\Omega^y_j} 1\, d\boldsymbol{y} \ .
\end{aligned}$$

Eqs. (1)-(2) of the main text can be combined to represent the mutual information as

$$\mathcal{M} = \int p(\boldsymbol{x},\boldsymbol{y})\log\left(\frac{p(\boldsymbol{x},\boldsymbol{y})}{p^x(\boldsymbol{x})p^y(\boldsymbol{y})}\right)d\boldsymbol{x}\, d\boldsymbol{y} \tag{S32}$$

Since $v_{ij} = v^x_i v^y_j$, the analog of Eq. (S30) is

$$p_{ij} \approx p(\boldsymbol{x}_i,\boldsymbol{y}_j)v^x_i v^y_j \ , \qquad p^x_i \approx p^x(\boldsymbol{x}_i)v^x_i \ , \qquad p^y_j \approx p^y(\boldsymbol{y}_j)v^y_j \ . \tag{S33}$$

Finally, combining all the above we get

$$\mathcal{M} \approx \sum_{i,j} p(\boldsymbol{x}_i,\boldsymbol{y}_j)\log\left(\frac{p(\boldsymbol{x}_i,\boldsymbol{y}_j)}{p^x(\boldsymbol{x}_i)p^y(\boldsymbol{y}_j)}\right)v^x_i v^y_j \approx \sum_{i,j} p_{ij}\log\left(\frac{p_{ij}}{p^x_i p^y_j}\right) \ , \tag{S34}$$

which identifies with the discrete defintion of $\mathcal{M}$.

As an aside, we note that Eq. (S31) has an intuitive interpretation: $\log v_i$ is exactly the entropy of a uniform distribution over Ω_i (whose probability density is $p = 1/v_i$). Therefore, the differential entropy $\tilde{S}$ measures the uncertainty (=entropy) associated with knowing in which Ω_i the observation $\boldsymbol{x}$ resides, plus the average uncertainty (=entropy) associated with knowing where does $\boldsymbol{x}_i$ resides within Ω_i. The latter cancels out when computing $\mathcal{M}$.

8. Derivation of Eq. (10) of the main text

Eq. (4) of the main text starts with a system X_0 of a given volume V_0 and looks at smaller and smaller subsystems (i.e. larger m). For the purposes of Eq. (10) of the main text we want to explore the other direction – assuming that X_0 is by itself a part of a much larger system and extrapolating from X_0 to the system size. To comply with the notation of the main text, where larger m's correspond to smaller subsystems X_m, we consider subsystems which are formally indexed by negative integers. Also, it will be useful to consider Eq. (3) of the main text normalized per unit volume. For any k we have

$$S(X_{k-1}) = 2S(X_k) - \mathcal{M}(X_k) \qquad \Rightarrow \qquad s(X_{k-1}) \equiv \frac{S(X_{k-1})}{V_{k-1}} = s(X_k) - \frac{\mathcal{M}(X_k)}{2V_k} \ , \tag{S35}$$

where we used the fact that $V_{k-1} = 2V_k$. Using this relation recursively we get

$$\begin{aligned} s(X_{-1}) &= s(X_0) - \frac{\mathcal{M}(X_0)}{2V_0} \\ s(X_{-2}) = s(X_{-1}) - \frac{\mathcal{M}(X_{-1})}{2V_{-1}} &= s(X_0) - \frac{\mathcal{M}(X_0)}{2V_0} - \frac{\mathcal{M}(X_{-1})}{4V_0} \\ s(X_{-3}) = s(X_{-2}) - \frac{\mathcal{M}(X_{-2})}{2V_{-2}} &= s(X_0) - \frac{\mathcal{M}(X_0)}{2V_0} - \frac{\mathcal{M}(X_{-1})}{4V_0} - \frac{\mathcal{M}(X_{-2})}{8V_0} \\ &\vdots \\ s(X_{-m}) &= s(X_0) - \frac{1}{2V_0}\sum_{k=0}^{m-1}\frac{\mathcal{M}(X_{-k})}{2^k} \end{aligned} \quad [S36]$$

We now assume that for subsystems larger than X_0 the mutual information is extensive, so by Eq. (9) of the main text we have $\mathcal{M}(X_{-k}) = (\ell_{-k}/\ell_0)\mathcal{M}(X_0)$. For our choice of selecting subsystems, we also have $\ell_{-k}/\ell_0 = 2^{\lfloor \frac{k+1}{2} \rfloor}$, where $\lfloor \cdot \rfloor$ is the floor function. We assume that X_0 is a square subsystem (subsystems alternate between square and rectangular, cf. Fig. 1 of the main text). Putting all this together we get

$$S(X_{-m}) = s(X_0) - \frac{\mathcal{M}_0}{2V_0}\sum_{k=0}^{m-1} 2^{\lfloor \frac{k+1}{2} \rfloor - k} \, . \quad [S37]$$

One can easily verify that in the limit $m \to \infty$ the sum in the last equation approaches 4. We conclude that

$$s(X_{-m}) = s(X_0) - 2\frac{\mathcal{M}_0}{V_0} \, . \quad [S38]$$

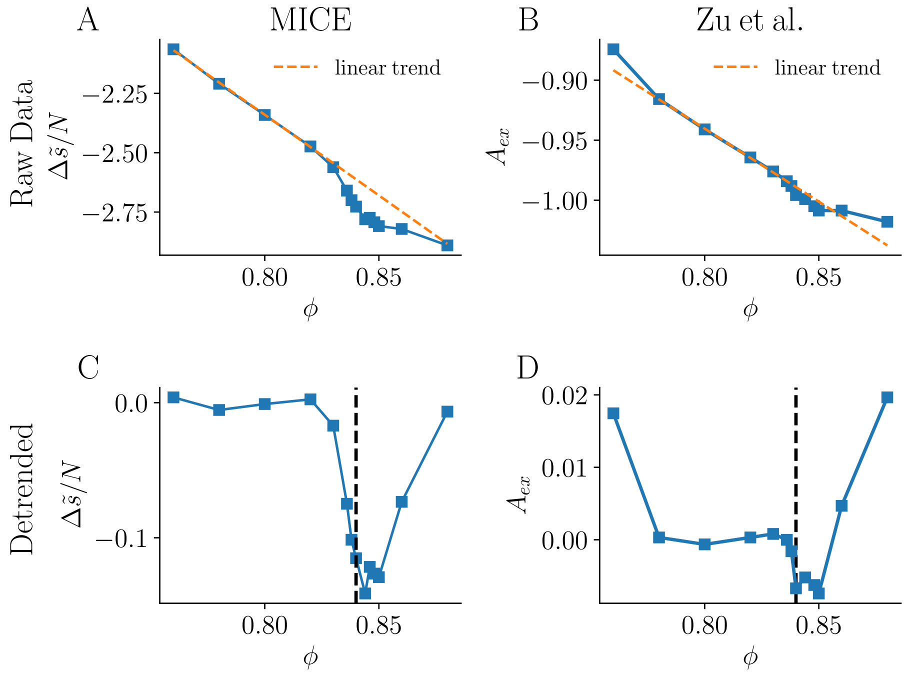


Fig. S6. Entropy estimation of the bidisperse soft sphere mixture, using two different methods, see text of Sec. 9 for a description. The dashed black line represents the theoretical jamming transition point.

9. Comparing *MICE* and the results of Zu et. al.

In the main text we claimed that *MICE* outperforms the compression method used by Zu et. al. (6) in detecting the jamming point. This was based on their statements that their Computable Information Density (CID) estimates do not show a minimum near the jamming point (see Sec. 3.5 of their paper).

In Fig. S6 we show a direct comparison between our data (left column, the same data appear as Fig. 4F and its inset in the main text) and theirs (right column, taken from Figure 6A of (6)).

The top row shows the estimation of the "excess entropy", i.e. the difference in entropy from some baseline behavior: With *MICE* this is achieved by omitting the entropic contribution of the smallest scales, cf. Eq. (8) of the main text. Zu et. al. do this by subtracting the information density of an ideal gas (see Sec. 2.3.2 of Zu et. al. (6)). These two baselines are conceptually similar but quantitatively different and therefore the absolute numbers differ somewhat between the methods. The trend, however, is informative.

To better visualize the signature of the transition, in the bottom row we plot the same data as the top row, with a linear trend (shown in dashed orange in the top row) subtracted. It is seen that the deviations from linearity are very pronounced when measured with *MICE*, but the CID estimation shows small deviations compared to the overall effect.

References

1. Paszke A, et al. (2019) Pytorch: An imperative style, high-performance deep learning library in *Advances in Neural Information Processing Systems 32*, eds. Wallach H, et al. (Curran Associates, Inc.), pp. 8024–8035.
2. Belghazi I, Rajeswar S, Baratin A, Hjelm RD, Courville AC (2018) MINE: mutual information neural estimation. *CoRR* abs/1801.04062.
3. Kent-Dobias J, Sethna JP (2018) Cluster representations and the wolff algorithm in arbitrary external fields. *Phys. Rev. E* 98(6):063306.
4. Katsinis D, Pastras G (2020) An inverse mass expansion for the mutual information in free scalar qft at finite temperature. *Journal of High Energy Physics* 2020(2):1–60.
5. Bitzek E, Koskinen P, Gähler F, Moseler M, Gumbsch P (2006) Structural relaxation made simple. *Phys. Rev. Lett.* 97(17):170201.
6. Zu M, Bupathy A, Frenkel D, Sastry S (2020) Information density, structure and entropy in equilibrium and non-equilibrium systems. *Journal of Statistical Mechanics: Theory and Experiment* 2020:023204.

4 Conclusions

This work introduces *MICE*, a method for estimating arbitrary physical systems' entropy. The introduced method proved to be both accurate and efficient. We demonstrated that *MICE* naturally allows us to decompose the entropy into contributions from different scales, providing an insightful diagnostic for physical systems' thermodynamics.

The decomposition of the system might be helpful when studying correlation lengths of a physical system. For example, we showed that using *MICE* we can identify the bidisperse mixture's relevant scale. Theoretically, we could create a physical system with controllable correlation lengths and show that using *MICE* these could be identified.

We have examined the capabilities of *MICE* on some canonical equilibrium physical systems - the 2D ferromagnetic and anti-ferromagnetic Ising models and the XY model. We have seen that *MICE* achieves a state of the art accuracy in entropy estimation.

This means that using *MICE* we might be able to examine some essential physical systems, such as the configurational entropy of amorphous solids [69], the entropy crisis of glassy systems [59], the entropy of active matter [25].

Although being promising, the currently proposed method has its limits. Since *MICE* is based on neural networks, it requires a reasonably powerful computer, with at least a decent GPU on it to run efficiently.

Physical simulations often have an extremely high resolution or many particles. ANNs in general, and as a result *MICE* as well, can handle inputs of limited sizes. It is important to remember that the computational bottleneck of *MICE* is the minimal system size (volume in 3D) for which the area law (cf. (2.2)) is observed. This determines the largest input on which an ANN should be trained. Recent studies have shown successful implementations of ANNs on a 3D input of size $128 \times 128 \times 128$ [70], but this is a limiting factor of the proposed method.

The current models on which *MICE* was tested had short-range correlations. These models are well fitted for convolutional neural networks, such as the one presented in this article. However, it is not clear how our model would perform on a system with long-range interactions.

Many studies have shown that some architectures allow neural networks to deal with "long-range interaction"-like problems. These include natural language processing, where neural networks need to understand the relationship between words that are spread in text [71], or scene description where the network needs to identify the relation between pixels far from each other in images [72]. Inspired by these studies, I believe it would be fascinating to examine *MICE* with new architectures on physical systems with long-range interactions. I believe that on these systems, *MICE* could outperform other methods, as the compression-based methods, in a significant manner.

I believe that *MICE* offers more than an efficient tool for entropy estimation. This research showed that using mutual information could identify phase transitions in the 2D Ising model. Wilms et al. detected the phase transition in the many-body quantum Lipkin-Meshkov-Glick model using mutual information [73]. Thus, it should be interesting to examine the behavior of mutual information for systems with complex phase transition, and *MICE* might offer a new insightful tool for studying physical systems.

Inspired by Wilms et al., I believe it should be possible and valuable to fit *MICE* for quantum systems. In quantum systems, the mutual information is fundamentally related to the entanglement of quantum states [74]. A relevant direction could be the extraction of entropy from quantum Monte Carlo simulations.

Lastly, I believe that we can benefit from the fact that *MICE* is based on a neural network model. These models are widely investigated, and many methods were introduced to receive insights from these models.

For example, in this research, we showed that using the transfer learning method, the run time of *MICE* is reduced by a factor of 10 and becomes around a few hours for an entire system. This implies that the network learns the core physical features of the systems. Although not being thoroughly researched in this work, it might be interesting to look at the network's feature maps as a function of temperature or other system properties.

References

[1] Ram Avinery, Micha Kornreich, and Roy Beck. Universal and accessible entropy estimation using a compression algorithm. *Phys. Rev. Lett.*, 123(17), Oct 2019.

[2] Hiromu Yakura, Shinnosuke Shinozaki, Reon Nishimura, Yoshihiro Oyama, and Jun Sakuma. Malware analysis of imaged binary samples by convolutional neural network with attention mechanism. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, CODASPY '18, page 127–134, New York, NY, USA, 2018. Association for Computing Machinery.

[3] Amit Nir, Eran Sela, Roy Beck, and Yohai Bar-Sinai. Machine-learning iterative calculation of entropy for physical systems. *Proceedings of the National Academy of Sciences*, 117(48):30234–30240, 2020.

[4] Josiah Willard Gibbs. A method of geometrical representation of the thermodynamic properties by means of surfaces. *Transactions of Connecticut Academy of Arts and Sciences*, pages 382–404, 1873.

[5] Lev D Landau and Evgeny M Lifshitz. *Statistical Physics: Volume 5*, volume 5. Elsevier, 2013.

[6] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(4):623–656, oct 1948.

[7] E. T. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):227–241, 1968.

[8] A. Kolmogorov. New metric invariant of transitive dynamical systems and endomorphisms of lebesgue spaces. *Doklady of Russian Academy of Sciences*, 119 (5):861–864, 1958.

[9] Ming Li and Paul M.B. Vitnyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 3 edition, 2008.

[10] Peter Grünwald and Paul M. B. Vitányi. Shannon information and kolmogorov complexity. *CoRR*, cs.IT/0410002, 2004.

[11] O. Melchert and A. K. Hartmann. A computational mechanics approach to estimate entropy and (approximate) complexity for the dynamics of the 2d ising ferromagnet, 2012.

[12] Mengjie Zu, Arunkumar Bupathy, Daan Frenkel, and Srikanth Sastry. Information density, structure and entropy in equilibrium and non-equilibrium systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2020:023204, 02 2020.

[13] H. A. Kramers and G. H. Wannier. Statistics of the two-dimensional ferromagnet. part i. *Phys. Rev.*, 60:252–262, Aug 1941.

[14] Lars Onsager. Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition. *Physical Review*, 65(3-4):117–149, feb 1944.

[15] Masahide Matsuura, Haruhiko Yao, Kazutoshi Gouhara, Ichiro Hatta, and Norio Kato. Heat capacity in $\alpha - \beta$ phase transition of quartz. *Journal of the Physical Society of Japan*, 54(2):625–629, 1985.

[16] DC Ginnings and RJ Corruccini. Enthalpy, specific heat, and entropy of aluminum oxide from 0 degrees to 900 degrees c. *Journal of research of the National Bureau of Standards*, 38(6):593–600, 1947.

[17] Stefano Martiniani, Paul M. Chaikin, and Dov Levine. Quantifying hidden order out of equilibrium. *Phys. Rev. X*, 9:011031, Feb 2019.

[18] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, may 1977.

[19] Ludwig Boltzmann. *Further Studies on the Thermal Equilibrium of Gas Molecules*, volume 1, pages 262–349. 07 2003.

[20] Christopher Jarzynski. Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.*, 78(14):2690, 1997.

[21] J. Meixner. *Entropy and Entropy Production*, pages 129–142. Macmillan Education UK, London, 1973.

[22] Georgy Lebon, David Jou, and José Casas-Vázquez. *Understanding non-equilibrium thermodynamics*, volume 295. Springer, 2008.

[23] Rosa Velasco, Leopoldo García-Colín, and Francisco Uribe. Entropy production: Its role in non-equilibrium thermodynamics. *Entropy*, 13, 12 2011.

[24] Gil Ariel and Haim Diamant. Inferring entropy from structure. *Phys. Rev. E*, 102:022110, Aug 2020.

[25] Cesare Nardini, Étienne Fodor, Elsen Tjhung, Frédéric van Wijland, Julien Tailleur, and Michael E. Cates. Entropy production in field theories without time-reversal symmetry: Quantifying the non-equilibrium character of active matter. *Phys. Rev. X*, 7:021007, Apr 2017.

[26] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[27] Ishmael Belghazi, Sai Rajeswar, Aristide Baratin, R. Devon Hjelm, and Aaron C. Courville. MINE: mutual information neural estimation. *CoRR*, abs/1801.04062, 2018.

[28] Gabriel Moreno-Hagelsieb. Operons across prokaryotes: Genomic analyses and predictions 300+ genomes later. *Current Genomics*, 7:163–170, 05 2006.

[29] Brian Swingle. Mutual information and the structure of entanglement in quantum field theory, 2010.

[30] Maciej Koch-Janusz and Zohar Ringel. Mutual information, neural networks and the renormalization group. *Nature Physics*, 14(6):578–582, Mar 2018.

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

[32] Raffaele Parisi, Elio D Di Claudio, G Lucarelli, and G Orlandi. Car plate recognition by neural networks and image processing. In *ISCAS'98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (Cat. No. 98CH36187)*, volume 3, pages 195–198. IEEE, 1998.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[34] Yao Tang, Fei Gao, Jufu Feng, and Yuhang Liu. Fingernet: An unified deep network for fingerprint minutiae extraction. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 108–116. IEEE, 2017.

[35] Dong-Kyum Kim, Youngkyoung Bae, Sangyun Lee, and Hawoong Jeong. Learning entropy production via neural networks. *Phys. Rev. Lett.*, 125:140604, Oct 2020.

[36] Askery Canabarro, Felipe Fernandes Fanchini, André Luiz Malvezzi, Rodrigo Pereira, and Rafael Chaves. Unveiling phase transitions with machine learning. *Phys. Rev. B*, 100(4), Jul 2019.

[37] Alexey Uvarov, Andrey Kardashin, and Jacob Biamonte. Machine learning phase transitions with a quantum processor, 2019.

[38] Rosenblatt. *The perceptron, a perceiving and recognizing automaton, Project Para*. Cornell Aeronautical Laboratory, 1957.

[39] Yann A. LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. *Efficient BackProp*, pages 9–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[40] Rajat Raina, Anand Madhavan, and Andrew Y. Ng. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 873–880, New York, NY, USA, 2009. Association for Computing Machinery.

[41] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, December 1989.

[42] Yann LeCun and Yoshua Bengio. *Convolutional Networks for Images, Speech, and Time Series*, page 255–258. MIT Press, Cambridge, MA, USA, 1998.

[43] Warren S Sarle. Stopped training and other remedies for overfitting. *Computing science and statistics*, pages 352–360, 1996.

[44] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017.

[45] David B Parker. Learnins logic. *Technical Report*, 1985.

[46] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[47] Alekh Agarwal, Peter L Bartlett, Pradeep Ravikumar, and Martin J Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012.

[48] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. arxiv:1412.6980.

[49] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010.

[50] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20(5):14, 2015.

[51] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.

[52] G. H. Wannier. Antiferromagnetism. the triangular ising net. *Phys. Rev.*, 79:357–364, Jul 1950.

[53] J M Kosterlitz and D J Thouless. Ordering, metastability and phase transitions in two-dimensional systems. *Journal of Physics C: Solid State Physics*, 6(7):1181–1203, apr 1973.

[54] Ji-Feng Yu, Zhiyuan Xie, and Tao Xiang. Two-dimensional classical XY model by HOTRG. In *APS March Meeting Abstracts*, volume 2013 of *APS Meeting Abstracts*, page Y29.004, March 2013.

[55] Glenn Agnolet, DF McQueeney, and JD Reppy. Kosterlitz-thouless transition in helium films. *Phys. Rev. B*, 39(13):8934, 1989.

[56] DJ Resnick, JC Garland, JT Boyd, S Shoemaker, and RS Newrock. Kosterlitz-thouless transition in proximity-coupled superconducting arrays. *Phys. Rev. Lett.*, 47(21):1542, 1981.

[57] Andrea J Liu and Sidney R Nagel. The jamming transition and the marginally jammed solid. *Annu. Rev. Condens. Matter Phys.*, 1(1):347–369, 2010.

[58] Corey S. O'Hern, Leonardo E. Silbert, Andrea J. Liu, and Sidney R. Nagel. Jamming at zero temperature and zero applied stress: The epitome of disorder. *Phys. Rev. E*, 68:011306, Jul 2003.

[59] Andrea Cavagna. Supercooled liquids for pedestrians. *Physics Reports*, 476(4-6):51–124, 2009.

[60] Rémi Monasson. Structural glass transition and the entropy of the metastable states. *Phys. Rev. Lett.*, 75(15):2847, 1995.

[61] D. Koeze, D. Vågberg, B. Tjoa, and B. Tighe. Mapping the jamming transition of bidisperse mixtures. *EPL (Europhysics Letters)*, 113:54001, 03 2016.

[62] Daniel Vågberg, Daniel Valdez-Balderas, M. A. Moore, Peter Olsson, and S. Teitel. Finite-size scaling at the jamming transition: Corrections to scaling and the correlation-length critical exponent. *Phys. Rev. E*, 83(3), Mar 2011.

[63] Randall D. Kamien and Andrea J. Liu. Why is random close packing reproducible? *Phys. Rev. Lett.*, 99:155501, Oct 2007.

[64] Jorge Kurchan and Dov Levine. Order in glassy systems. *Journal of Physics A: Mathematical and Theoretical*, 44(3):035001, dec 2010.

[65] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[66] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970.

[67] Ulli Wolff. Collective monte carlo updating for spin systems. *Phys. Rev. Lett.*, 62:361–364, Jan 1989.

[68] Jaron Kent-Dobias and James P. Sethna. Cluster representations and the wolff algorithm in arbitrary external fields. *Phys. Rev. E*, 98:063306, Dec 2018.

[69] Eran Bouchbinder, JS Langer, and Itamar Procaccia. Athermal shear-transformation-zone theory of amorphous plastic deformation. i. basic principles. *Phys. Rev. E*, 75(3):036107, 2007.

[70] Jun Young Gwak. 3d model classification using convolutional neural network.

[71] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA, 2008. Association for Computing Machinery.

[72] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 487–495. Curran Associates, Inc., 2014.

[73] Johannes Wilms, Julien Vidal, Frank Verstraete, and Sébastien Dusuel. Finite-temperature mutual information in a simple phase transition. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(01):P01023, Jan 2012.

[74] Luigi Amico, Rosario Fazio, Andreas Osterloh, and Vlatko Vedral. Entanglement in many-body systems. *Reviews of modern physics*, 80(2):517, 2008.

תקציר

אנטרופיה היא אחד הגדלים הבסיסיים בחקר של מערכות פיזיקליות. היא מקושרת לרמת הסדר במערכת ומשחקת תפקיד משמעותי בחקר של מעברי פאזה, היווצרות של תבניות, קיפול חלבונים ועוד. אנטרופיה נחקרת רבות גם במסגרת תורת האינפורמציה, שם היא מוגדרת ככמות המידע שקיים במערכת נתונים כלשהי.

למרות זאת, שיטות קיימות להערכת אנטרופיה לוקות בעלות חישובית גבוהה, התאמה למערכות מסוימות בלבד, דיוק נמוך וחוסר יכולת להתמודד עם מערכות מורכבות בעלות אינטראקציות חזקות. למשל, חישוב ישיר של אנטרופיה מתוך מעבר על כל מצבי המערכת נהיה לא ישים ברמה החישובית כבר עבור מערכות בינאריות קטנות. נוסף על כך, שיטות רבות שעובדות עבור מערכות בשיווי משקל לא מתאימות למערכות מחוץ לשיווי משקל.

בהשראת רעיונות מתורת האינפורמציה, מחקר זה מראה שניתן לחשב אנטרופיה על ידי חלוקה חוזרת ונשנית של מערכת לתתי מערכות קטנות יותר, והערכה של האינפורמציה ההדדית בין כל זוג חצאים של תתי מערכות. הערכת האינפורמציה ההדדית נעשית באמצעות שיטה מבוססת למידת מכונה שהוצעה לאחרונה. שיטה זו מתאימה לכל ארכיטקטורה של רשת, כך שניתן לבחור ארכיטקטורה שמתאימה למבנה ולסימטריות של המערכת הנתונה. בניגוד לשיטות נוספות שהוצעו לאחרונה, רשתות נוירונים מותאמות היטב לעבודה עם מערכות בשניים ושלושה ממדים. מחקר זה מצטרף לרבים אחרים שמראים שחקר הפיזיקה יכול להפיק רבות מההתפתחויות האחרונות באלגוריתמים במדעי המחשב בכלל, ובלמידת מכונה בפרט.

השיטה המוצעת לעיל מאפשרת חישוב מדויק של אנטרופיה לשלל מערכות, תרמיות וא-תרמיות, בדיוק גבוה במיוחד. ספציפית, השיטה המוצעת נבחנה באמצעות מספר מערכות ספין קלאסיות, ושימשה לזיהוי נקודת הדחיסות (jamming) של מערכת דו רכיבית של דסקאות רכות. פרט לכך, באמצעות שיטת העברת הלמידה (transfer learning), השיטה המוצעת פועלת בזמן חישוב מהיר יחסית.

לבסוף, מועלה האפשרות שפרט לשימושיות בחישוב אנטרופיה, האינפורמציה ההדדית עצמה יכול לספק תובנות משמעותיות בחקר של מערכות פיסיקליות. מכיוון שהאינפורמציה ההדדית כוללת בתוכה את כל המידע הרלוונטי אודות המערכת הפיסיקלית, ניתן להשתמש בה על מנת לזהות מעברי פאזה ולחשב אורכי קורלציה.

אוניברסיטת תל אביב

בית הספר לפיסיקה ואסטרונומיה ע"ש ריימונד ובברלי סאקלר

חישוב איטרטיבי של אנטרופיה מבוסס למידת מכונה

חיבור זה הוגש כעבודת מחקר לקראת התואר "מוסמך פיסיקה" באוניברסיטה על ידי

עמית ניר

העבודה נעשתה בבית הספר לפיסיקה ואסטרונומיה

בהנחיית פרופ' רועי בק

תשפ"א