Machine Learning: Science and Technology

LETTER • OPEN ACCESS

Watch and learn—a generalized approach for transferrable learning in deep neural networks via physical principles

To cite this article: Kyle Sprague *et al* 2021 *Mach. Learn.: Sci. Technol.* **2** 02LT02

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 02LT02 https://doi.org/10.1088/2632-2153/abc81b

MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
11 March 2020

REVISED
7 October 2020

ACCEPTED FOR PUBLICATION
5 November 2020

PUBLISHED
3 February 2021

LETTER

Watch and learn—a generalized approach for transferrable learning in deep neural networks via physical principles

Kyle Sprague[1], Juan Carrasquilla[2,3], Stephen Whitelam[4] and Isaac Tamblyn[1,2,5]

[1] Department of Physics, University of Ottawa, Ottawa, Ontario, Canada
[2] Vector Institute, Toronto, Ontario, Canada
[3] Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario, Canada
[4] Molecular Foundry, Lawrence Berkeley National Laboratory, Berkeley, CA, United States of America
[5] National Research Council of Canada, Ottawa, Canada

E-mail: isaac.tamblyn@nrc-cnrc.gc.ca

Keywords: deep learning, statistical mechanics, inverse problem, Autoregressive model, Extrapolation

Abstract

Transfer learning refers to the use of knowledge gained while solving a machine learning task and applying it to the solution of a closely related problem. Such an approach has enabled scientific breakthroughs in computer vision and natural language processing where the weights learned in state-of-the-art models can be used to initialize models for other tasks which dramatically improve their performance and save computational time. Here we demonstrate an unsupervised learning approach augmented with basic physical principles that achieves fully transferrable learning for problems in statistical physics across different physical regimes. By coupling a sequence model based on a recurrent neural network to an extensive deep neural network, we are able to learn the equilibrium probability distributions and inter-particle interaction models of classical statistical mechanical systems. Our approach, distribution-consistent learning, DCL, is a general strategy that works for a variety of canonical statistical mechanical models (Ising and Potts) as well as disordered interaction potentials. Using data collected from a single set of observation conditions, DCL successfully extrapolates across all temperatures, thermodynamic phases, and can be applied to different length-scales. This constitutes a fully transferrable physics-based learning in a generalizable approach.

1. Introduction

Machine learning has emerged as a powerful tool in the physical sciences, seeing both rapid adoption and experimentation in recent years. Already, there have been demonstrations of learning operators [1], detecting unseen patterns within data [2], ‘discovering’ physical equations [3], and predicting trends within the scientific literature [4]. Within the field of statistical mechanics, machine learning was recently used to estimate the value of the partition function [5], solve canonical models [6], and generative models conditioned on the Boltzmann distribution have been shown to be efficient at sampling statistical mechanical ensembles [7]. The connection between physics and machine learning continues to strengthen, with new results appearing daily.

Despite these and other successes, machine learning has severe limitations. A major obstacle to the more widespread use of machine learning in the physical sciences is that typically, learned models tend to exhibit poor transferability. Using a model outside of the training or parameterization set can be unreliable [8]. This, along with the lack of well defined and well-behaved error estimates can result in erroneous results, the magnitude of which are often uncontrolled. While transferability can be somewhat improved through techniques such as regularization [9], there is currently no general approach that can guarantee transferability to new conditions or ensure reliability of a model when it is presented with previously unseen data.

Here we demonstrate a new approach that overcomes the transferability problem by coupling machine learning concepts with physical principles in a new way. The approach, which we call distribution-consistent learning (DCL) enables fully transferrable learning with a minimal number of observations: using it, we can collect observations at a single set of conditions, yet make accurate predictions for all others (including in different physical phases and across phase boundaries). Using DCL, it is possible to extrapolate observations over a wide range of conditions, including those far from the training set. This is achieved through the straightforward yet rigorous application of the physical concept of equilibrium and the postulate of the uniformity of physical law.

To explain DCL, we will first focus on simple classical spin models (ferromagnetic Ising and Potts models with coupling constant $J = 1$) for the purposes of illustration and validation. For such simple cases, the application of statistical methods to 'invert' observations are not new [10, 11].

Conceptually, DCL is valid when ensemble probabilities are governed by a known statistics (e.g. Boltzmann or Fermi distributions, which describe equilibrium and some near-equilibrium processes) and the interaction energies do not depend on the control parameters. Finally, we demonstrate the generalizability of DCL explicitly by applying it, without modification, to an unsolved inversion problem: a semi-local disordered system where all couplings between neighbours are selected randomly from a Gaussian distribution ($\mu = 0$, $\sigma = 1$).

2. Probabilities at equilibrium

When a system with N degrees of freedom is at equilibrium, its macroscopic properties can be interpreted as weighted averages over a large number of micro-states, $\sigma = \{\sigma_1 ... \sigma_N\}$. For an example such as a polymer, such micro-states may be different molecular conformations, whereas for a spin system they are the 2^N different configurations of a collection of N spins ($\sigma_i \ = \uparrow, \downarrow$) on a lattice. Classically, such micro-states are visited with a probability given by the Boltzmann distribution. For the case of the canonical ensemble in particular (constant particle number, N, volume, V, and temperature, T), the ensemble probability of visiting a particular micro-state is given by

$$P(\sigma) = \exp(-\beta \mathcal{H}(\sigma))/Z, \tag{1}$$

where $\mathcal{H}$ is the classical Hamiltonian that specifies the energy of the configuration σ, i.e. $E = \mathcal{H}(\sigma)$, β is the inverse temperature, $\frac{1}{k_B T}$, and Z is the partition function of the system. The Hamiltonian of the two-dimensional (2D) Ising model is given by

$$\mathcal{H}(\sigma) = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j. \tag{2}$$

Likewise, the 2D Potts model Hamiltonian is given by

$$\mathcal{H}(\sigma) = -J \sum_{\langle i,j \rangle} \cos(\theta_i - \theta_j), \tag{3}$$

where $\langle i,j \rangle$ are nearest neighbours on a 2D lattice and $\theta_n = \frac{2\pi n}{Q}$ is the spin angle for a Q-spin Potts model. Here we set $J = 1$ for both Ising and Potts models.

We note that in principle if one were able to observe an equilibrium system long enough, it would be possible to estimate the ensemble probabilities of each micro-state simply by counting how often the system visits a particular micro-state and divide by the number of observed events. Of course, for all but trivial state spaces, this is impractical. The probability of visiting micro-states above the ground state at finite T is exponentially small in E, and the probability of visiting them multiple times within an observation period (which would be necessary in order to collect reliable estimates) is even smaller. Micro-states visited during a macroscopic experiment represent only a vanishingly small fraction of the possible configurations which can be realized. Watching and counting is not an option.

3. Essence of the approach

We note that equation (1), combined with the observation temperature (β_O^{-1}), is sufficient information to determine the energy differences between any two micro-states σ', σ:

$$\Delta E(\sigma', \sigma) = \beta_O^{-1} \ln \frac{P(\sigma)}{P(\sigma')}. \tag{4}$$

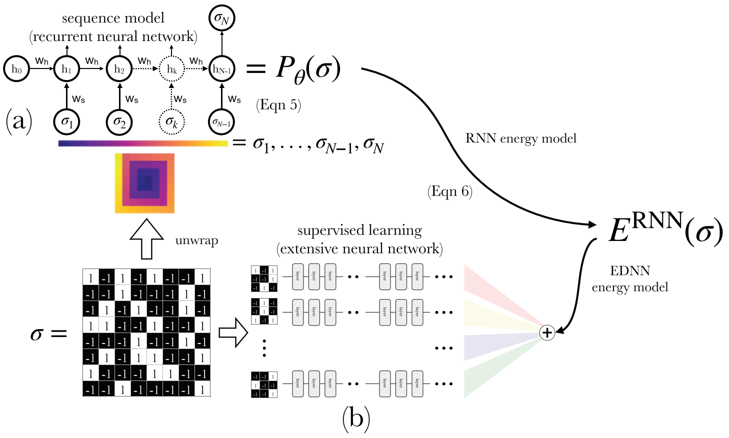

Figure 1. Distribution-consistent learning. (a) We conduct numerical experiments at a single set of 'experimental' conditions and collect a large dataset of visited micro-states σ. We train a sequence model based on an RNN on the collected dataset and learn a model $P_\theta(\sigma)$. (b) The RNN probability predictions can train a second neural network, via supervised learning. The estimate of $P_\theta(\sigma)$ is obtained from the RNN through a chain of conditional probabilities (equation 5). Applying the same process to a second configuration σ' results in an estimate of the energy difference of the two, ΔE.

Thus, if one could accurately estimate $P(\sigma)$, it would be possible to create a model Hamiltonian capable of computing properties under different conditions. The connection between probabilities and energies is core to the DCL approach.

Rather than attempt to infer micro-state visitation probabilities from counting, we instead characterize them with a probabilistic model $P_\theta(\sigma)$ with parameters θ by exploiting the underlying the structural properties of the system induced by its Hamiltonian and temperature. It is generally known that interactions between spins can give rise to correlations on a range of scales, and that emergence of such correlations are temperature dependent. We posit that these correlations are similar to patterns that emerge in language due to the rules of grammar: we can 'unwrap' the variables σ into a sequence of variables σ_i (figure 1) and analyze them using language-based sequential machine learning methods. While many such sequence models exist, recurrent neural networks (RNNs) are a particularly powerful deep learning technique which have seen significant use in recent years. RNN differ from the more common feed-forward neural networks as they reuse part of their output signal as input each time they are executed. RNN possess a hidden state, h, which represents a low dimensional space that encodes signals previously observed by the network (i.e. the previous values of a sequence). Like a feed-forward network, RNN also contain a learned set of weights and biases w. Collectively, they, along with the typology, uniquely define the model P_θ.

Primarily, RNN have been used in natural language processing (e.g. predictive text and translation), as well as predicting time-series data such as stock markets and weather. More recently, autoregressive RNN have been adapted to spatially structured inputs such as images [12]. A crucial feature of RNN (from the perspective of DCL) is that they have the ability to directly estimate $P(\sigma)$. We also note that normalizing flows [13] could be used with a continuous version of DCL. For a detailed review of the ability of RNN to estimate P, we refer the reader to section II A of Ref [14].

Here we first train an RNN probabilistic model $P_\theta(\sigma)$ on the micro-states observed at a fixed set of experimental temperature conditions, β_O^{-1} (for this study we used $\beta_O = \beta_c \approx 0.4407$) as DCL requires only data collected at a single temperature. Formally, the precise temperature of observation does not matter, so long as the system is approximately ergodic. We tested this explicitly by increasing the temperature to $\beta_O = 0.2500$, then $\beta_O = 0.1250$. This did not change our results qualitatively. Since DCL assumes de-correlated observations at low temperatures, predictive accuracy can be improved with sub-sampling techniques (see below).

Our training set consists of observations σ as well as the thermodynamic conditions upon which they were collected (i.e. the observation temperature β_O^{-1}). Importantly, the Hamiltonian operator and values of the energy are *not included* in our training data. This is analogous to what occurs experimentally—one does not generally have knowledge of the interaction potential between particles, but can always conduct an experiment at fixed conditions and make observations of which micro-states are explored during such an experiment.

To train the RNN, we provided it with examples of spin micro-states as unwrapped configurations (figure 1). We experimented with different 'unwrapping' techniques (which we called 'snake' or 'spiral' respectively); tests confirm our results are not sensitive to this choice, so long as the procedure is used

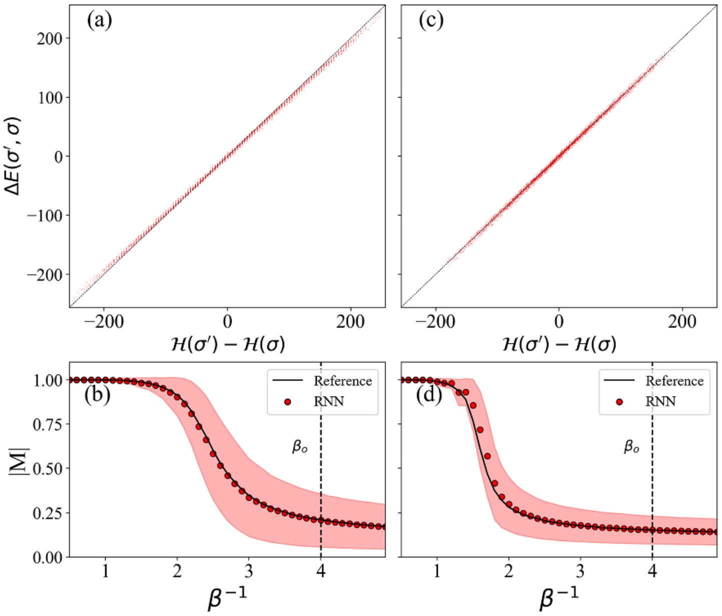

Figure 2. Performance of RNN energy models for the cases of an 8 × 8 Ising (left) and 3-state Potts (right) lattice. In (a) and (c) the predicted energy differences $\Delta E(\sigma, \sigma')$ are compared to their true values $\mathcal{H}(\sigma') - \mathcal{H}(\sigma)$. The black diagonal line represents the true energy differences and serves as a guide to the eye. Using these energy predictions within a Markov chain Monte Carlo simulation, it is possible to estimate observables (e.g. magnetization $|M| = \frac{1}{N}\sum_i |\mathbb{E}[\sigma_i]|$) over a wide range of temperatures (bottom row, (b) and (d)). We note that RNN energy models are trained only on samples which are recorded at β_O (denoted as a vertical dashed line), yet can be used predictions observables across a wide range of physical conditions, including regimes where the systems exhibit phase transitions in the $N \to \infty$. The red shaded area indicates the standard deviation of the MCMC simulation and the reference line indicates a numerical simulation using the true Hamiltonian.

consistently. Our neural network training procedure and architecture follow standard techniques and are reported in the appendix A.

Once the RNN has been trained, we can use it to estimate a particular $P(\sigma) \approx P_\theta(\sigma)$ by computing a series of conditional probabilities on our unwrapping $\sigma_1, \ldots, \sigma_N$ of the spin values in σ. To do this, we initialize the RNN with the first spin value of σ_1 and record its conditional probability prediction $P_\theta(\sigma_2|\sigma_1)$.

The total probability of $P_\theta(\sigma)$ is found from the multiplicative product of many conditional probabilities:

$$P_\theta(\sigma) = P_\theta(\sigma_1)\prod_{i=2}^{N} P_\theta(\sigma_i|\sigma_{i-1}\ldots\sigma_1). \tag{5}$$

Using equation (5), we can estimate the probability of any micro-state, $P_\theta(\sigma)$, and define its energy:

$$E^{RNN}(\sigma) = -\beta_O^{-1}\ln[P_\theta(\sigma)] + C, \tag{6}$$

where C is some reference energy. When calculating the energy difference between two micro-states σ', σ, one can simply take $E^{RNN}(\sigma') - E^{RNN}(\sigma)$ and obtain equation (4). We are able to compute such an estimate for any pair, including configurations which were never visited during the training process. Henceforth, we refer to this process of estimating E^{RNN} as the RNN energy model.

4. Using the RNN energy model

The top row of figures 2((a),(c)) shows the performance of RNN energy models across a range of energies for both the Ising and Potts ($Q = 3$) models. In both cases, the RNN energy model does an excellent job estimating the energy difference between any two micro-states. We note that it is exactly this quantity which is needed to perform finite temperature Markov chain Monte Carlo (MCMC) simulations to predict the thermodynamic properties of a spin system.

Using these models, we carry out such numerical simulations under conditions which are far away from the training set. When we compare results for the phase transitions generated using the true Hamiltonian with those generated with the RNN energy models, we see excellent agreement for both Ising and Potts models (figures 2(b),(d), bottom row). This confirms that the RNN model errors in the prediction of $\Delta E(\sigma', \sigma)$ are small enough that they do not alter the essential physics of systems under study.

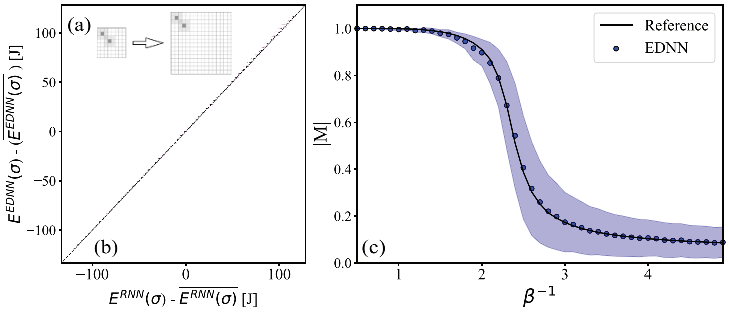

Figure 3. (a) Schematic of an EDNN learning. By training on small scale examples of a problem (8 × 8 Ising model), EDNN can learn the locality length-scale of an operator. (b) Performance of an EDNN energy model trained on labels produced by the RNN energy model. (c) Larger-scale (16 × 16) simulations are now possible thanks to the interaction length-scale learning of the EDNN energy model. The model was trained on data collected at a single temperature of a small lattice (8 × 8) yet can be used to accurately predict the full phase behaviour of a larger lattice across all temperatures. This constitutes fully transferrable learning across conditions and length-scales. The blue shaded area indicates the standard deviation of the MCMC simulation and the reference line indicates a numerical simulation using the true Hamiltonian.

Since the RNN already allows us to estimate ensemble probabilities (and thus energy labels from observation), we might think that nothing more can be extracted from our initial observations. It turns out, however, we can incorporate more physics knowledge through the use of a second neural network. We will demonstrate that incorporating this second network will both overcome some limitations of the RNN energy model and improve the accuracy of our predictions *without any new observations or labels*.

One of the most obvious disadvantages of the RNN energy model is that it has no concept of locality with respect to energy updates. Whenever we wish to know the difference in energy between σ' and σ, we must reevaluate all of the conditional probabilities which make up $P_\theta(\sigma')$ and $P_\theta(\sigma)$. For interactions which are short-ranged (as is the case for both the Ising and Potts models), this is very inefficient. In conventional simulations of such systems, it is customary to reevaluate only interactions which have changed as a result of the MC trial move. With the RNN energy model, there is no general way to achieve such 'local' energy updates. This is because for spin flips at some locations in the lattice one may need to recompute only some conditionals, but in general, one has to recompute an extensive number of them

Another limitation is that the RNN energy model is only able to make predictions for system sizes which are the same as those within the training set. Ideally, one would like to be able to observe an $L \times L$ system, learn something from it, and then make predictions for a larger $M \times M$ case.

5. The EDNN energy model

In previous work [15], we showed that with the proper construction, neural networks have the ability to directly learn the locality length-scale, l, of operators such as the Hamiltonian. By locality length-scale, we mean the amount of information in the neighbourhood of a focus region, f, which is necessary to compute extensive properties. Magnetization, for example, is a fully local ($l = 0$) operator. It is possible to divide the task of computing magnetism for every site in the system, record the value, and sum all at the end (since it is an extensive quantity). For a nearest neighbour spin model, additional context, c, is needed in order to determine site energies, i.e. the values of the spins in the neighbourhood. Using an extensive deep neural network (EDNN), these locality scales can be learned directly from the data through hyperparameter optimization of c.

Initially, our motivation for using an EDNN in this investigation was to be able to learn from small scale systems (and 8 × 8 spin model in this case) and apply that learning to a larger system (e.g. 16 × 16). Through the use of energy labels predicted by the RNN on a set of spin configurations, we can train the EDNN model to predict energies via regression using a mean-square error loss [15]. As expected, the EDNN is able to take the small scale examples and transfer that learning to larger systems (figure 3 shows the performance of the EDNN energy model). Creating an EDNN energy model had another unforeseen benefit however, as noted below.

By construction, EDNN topologies require that physical laws are the same everywhere. They are designed to learn a function which, when applied across a configuration, maps the sum of outputs to an extensive quantity such as the internal energy. Interestingly, in this case, we find that this physics-based network design requirement results in improved performance in predicting the underlying interactions even when the labels used in training have noise introduced by the imperfect RNN energy model.

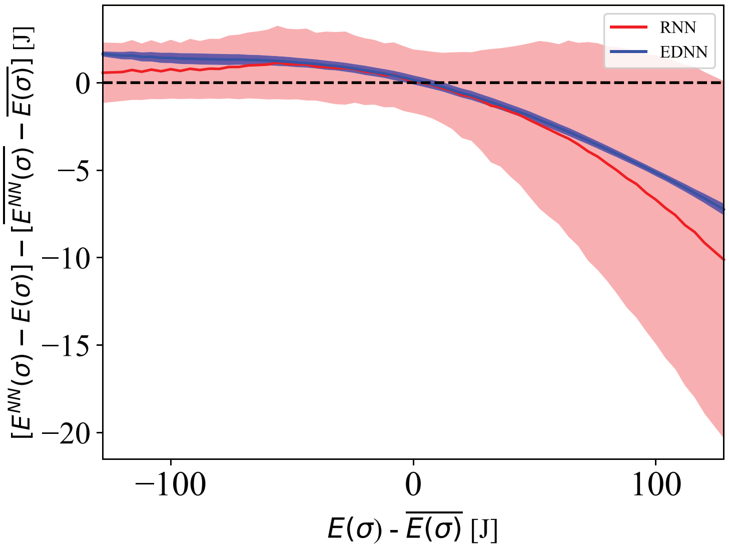

Figure 4. Comparison of RNN and EDNN energy models and their standard deviation relative to true values of the energy over a wide range. The EDNN model has lower variance and lower RMSE relative to true data, even though it has never seen those labels. Results for the 8 × 8 Ising model are shown. Shared area indicates variance over a sample of 8000 configurations per energy window.

As discussed above, we first train an RNN to predict the energy E^{RNN} (where C is chosen such that average energy $\overline{E^{RNN}(\sigma)}$ over a set of random micro-states is zero). In the second step, we train an EDNN to reproduce predictions from the RNN. The EDNN has never seen labels other than those estimated by the RNN. Despite this, when we compare EDNN predictions relative to the true values—it has better performance than the RNN itself. We surmise that the physics-based construction of the EDNN enables it to see through noise introduced by the imperfect RNN and achieve a better estimate of the underlying operator (the root mean squared error, RMSE, of the RNN energy model for Ising is 5.69 J compared with 2.75 J for the EDNN, figure 4). EDNN enforces the postulate of uniformity of physical law into our training procedure and the performance improves as a result.

6. Sampling and observables

In order to generate micro-states at a fixed set of temperature conditions β_O, we used the Metropolis–Hastings algorithm (a MCMC method). Proposed samples σ' were generated by flipping a single spin in the previous sample σ and were accepted with probability

$$p(\sigma'|\sigma) = \min(1, \exp[-\beta\Delta E(\sigma', \sigma)]). \tag{7}$$

In the limit of a long Markov chain, this provides samples from the Boltzmann distribution p which can then be used to train RNN models as well as estimate observables like the internal energy

$$U = \mathbb{E}_{\sigma\sim p}\left[\mathcal{H}(\sigma)\right] \tag{8}$$

and the absolute magnetization

$$|M| = \mathbb{E}_{\sigma\sim p}\left|\frac{1}{N}\sum_i \sigma_i\right|, \tag{9}$$

where $\mathbb{E}_{\sigma\sim p}$ denotes the expectation value with respect to the Boltzmann distribution p.

When generating micro-states at low temperatures, the low acceptance rate will cause Markov chains produced to be highly correlated (with micro-states often repeating). This can create biases when training an RNN as it may favour micro-states correlated with the training data over equiprobable states. For the temperatures we considered, we computed the autocorrelation function (in time) to estimate the correlation time via an exponential fit. These values are listed in table 1

Table 1. Correlation times τ at our sampling temperatures and system sizes. Correlation times are in terms of MCMC steps using the single spin-flip algorithm mentioned above.

Name	System size	β	Correlation time
τ_a	(8×8)	0.1250	$4.7 \pm 0.2\ (\times 10^1)$
τ_b	(8×8)	0.2500	$1.7 \pm 0.2\ (\times 10^2)$
τ_c	(8×8)	0.4407	$1.7 \pm 0.1\ (\times 10^4)$
τ_d	(16×16)	0.4407	$3.1 \pm 0.4\ (\times 10^5)$
τ_e	(24×24)	0.4407	$1.58 \pm 0.08\ (\times 10^6)$

Table 2. Comparison of RNN and EDNN energy models trained under different conditions. RMSE results are obtained by evaluating predictions on our 8×8 Ising test data (see appendix A1). Since multiple RNN models were trained on each data-set, each model contributed equally to mean-squared error. Seeds correspond to the number of independent trajectories used in MCMC train-data generation and sampling period is in MCMC steps.

β	Sampling period	Seeds	Model	RMSE
(A) 0.1250	$\tau_a/(4.7 \times 10^1)$	2	RNN	8.813
	$\tau_a/(4.7 \times 10^1)$	2	EDNN	5.046
	$\tau_a/(4.7 \times 10^0)$	32	RNN	1.673
	$\tau_a/(4.7 \times 10^0)$	32	EDNN	0.651
(B) 0.2500	$\tau_b/(1.7 \times 10^2)$	2	RNN	5.694
	$\tau_b/(1.7 \times 10^2)$	2	EDNN	2.747
	$\tau_b/(1.7 \times 10^1)$	32	RNN	1.431
	$\tau_b/(1.7 \times 10^1)$	32	EDNN	1.231
(C) 0.4407	$\tau_c/(1.7 \times 10^4)$	2	RNN	17.619
	$\tau_c/(1.7 \times 10^4)$	2	EDNN	12.489
	$\tau_c/(1.7 \times 10^3)$	32	RNN	4.472
	$\tau_c/(1.7 \times 10^3)$	32	EDNN	1.585
	$\tau_c/(1.7 \times 10^1)$	32	RNN	2.863

Table 3. Comparison of DCL (our work) with VAN [16] and NIS [17] on a 24×24 and 16×16 lattice (trained at β_c). For DCL, the Hamiltonian is estimated with an EDNN trained with RNN C (de-correlated, table A1) and given a reference ground state of -2 J per spin. Internal energy and absolute magnetization are estimated with MCMC using the EDNN Hamiltonian and averaged over 16 runs. Free energy and entropy are estimated by directly sampling from the probability distribution generated by our RNN and averaged over three RNN models. RNN models were trained with 32 seeds and a sampling period of 10 steps per observation (correlation times are given in table 1). Since evaluation of the free energy requires both probability and energy labels, both the EDNN and RNN are used in its estimate. It is important to note that the *only* energy labels given in DCL are the ground states. Reference values for entropy and free energy are given by the exact analytical solutions [18] whereas for internal energy and absolute magnetization, they are given by MCMC simulations using the true Hamiltonian.

Lattice	Sampler	$\frac{\vert U\vert}{L^2}$	$\frac{\vert M\vert}{L^2}$	$\frac{S}{L^2}$	$\frac{F}{L^2}$
(24×24)	VAN [17]	−1.5058 (0.0001)	0.7829 (0.0001)	0.26 505 (0.00 004)	−2.107 250 (0.000 001)
	NIS [17]	−1.43 (0.02)	0.67 (0.03)	0.299 (0.007)	−2.1128 (0.0008)
	DCL	−1.449 (0.001)	0.697 (0.003)	0.328 (0.009)	−2.074, (0.009)
Reference		−1.440 (0.002)	0.678 (0.003)	0.296	−2.112
(16×16)	VAN [17]	−1.4764 (0.0002)	0.7478 (0.0002)	0.28 081 (0.00 007)	−2.11 363 (0.00 001)
	NIS [17]	−1.4533 (0.0003)	0.71 363 (0.00 004)	0.2917 (0.0002)	−2.11 529 (0.00 001)
	DCL	−1.459 (0.001)	0.725 (0.002)	0.32 (0.02)	−2.07 (0.04)
Reference		−1.453 (0.001)	0.714 (0.002)	0.292	−2.115

In order to combat this and de-correlate the training data, two measures are taken. First, multiple trajectories are sampled using independent seeds during MCMC train-data generation, then the sampling period is increased by including only one of every n micro-states in train-data (while running the simulation for n times the length). The relative performance of RNN (and their corresponding EDNN) models with their de-correlation details can be found in table 2.

We note that while the EDNN provides an energy model, the RNN model estimates energy by first computing the probability of a given micro-state. Due to this, an RNN can be used to estimate observables which involve the partition function such as entropy

$$S \approx -\mathbb{E}_{\sigma \sim P_\theta}\left[\ln(P_\theta(\sigma))\right] \tag{10}$$

and free energy

$$F \approx \mathbb{E}_{\sigma \sim P_\theta}\left[\mathcal{H}(\sigma) + \beta^{-1}\ln(P_\theta(\sigma))\right] \tag{11}$$

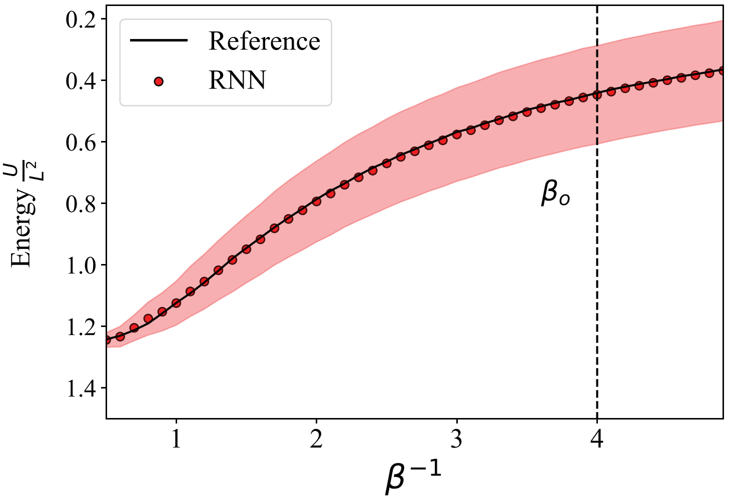

Figure 5. Predicting the finite temperature properties of a disordered system. Observations of micro-states are made at a single fixed temperature, whereas predictions can be made across a wide range of temperatures. The shaded region denotes the standard deviation of the MCMC simulation. The reference line indicates a numerical simulation using the true Hamiltonian.

so long as the RNN is trained at the appropriate inverse-temperature. It is important to note that any observation involving energy (internal or free) must either be expressed in the form of energy difference, or be given some reference energy (like a ground state of −2 J per lattice site in the Ising model) when using DCL.

Table 3 details DCL performance in estimating observables on a 16 × 16 and 24 × 24 lattice and compares it to other sampling methods [16, 17].

7. A general method

As a demonstration of the generality of DCL, we now consider the case of a disordered system where interactions between neighbours are sampled from a random distribution (the Edwards–Anderson model). Even with a much more complex and rich Hamiltonian, the RNN is able to learn only from observations at a fixed temperature, yet can be used to make accurate predictions across a wide range of conditions. Again, only unlabelled observations at a single fixed temperature are required to determine the behaviour of the system under unseen conditions, figure 5 (see appendix for another example of random couplings).

By applying DCL to a disordered system, we have demonstrated, for the first time, the ability to effectively learn the Hamiltonian operator directly without ever knowing its form, symmetries, or seeing directly labelled examples. This is the first time such an inversion has been demonstrated for disordered systems.

8. Limitations of DCL

Distribution consistent learning is conducted in two steps. We first train an RNN probabilistic model $P_\theta(\sigma)$ to extract information about an $L \times L$ system. Then, we train an EDNN energy model and generalize it to the $M \times M$ case. While the energy model can be generalized, the probabilistic model cannot and is only valid under the conditions seen in training. This can become problematic when estimating observables which require a probabilistic model such as free energy and entropy. As an example, if one wanted to know the free energy of a 24 × 24 lattice at the critical temperature β_c but only had access to an 8 × 8 lattice at some β_O, they would need to first train an RNN on the 8 × 8 lattice at β_O, generalize the RNN energy model to 24 × 24 using an EDNN, then use said EDNN energy model as well as MCMC to finally train a probabilistic model on the 24 × 24 lattice at β_c.

Additionally, while scaling up to larger lattice sizes is possible using an EDNN, creating a probabilistic model using an RNN becomes difficult. When training on larger lattice sizes (16 × 16 and above on the Ising model), high energy states were labelled with a probability of zero. Due to this, the only way to estimate observables using the RNN was to sample directly from $P_\theta(\sigma)$. Training an EDNN energy model using $P_\theta(\sigma)$ is also an issue since a probability of zero corresponds to infinite energy.

9. Conclusion

With knowledge only of the physical constraints (i.e. the statistical ensemble) and a statistically significant number of observations from only a *single* set of thermodynamic conditions, DCL is able to extract enough information about a physical system to produce an energy model capable of predicting its behaviour over a wide range of unseen conditions and across different length-scales, including systems that exhibit phase transitions in the thermodynamic limit $N \to \infty$. The equilibrium properties of physical systems are described by their Hamiltonian and thermodynamic ensemble. The Hamiltonian maps microstate-to-energy, and the temperature defines the selection probability based on the degeneracy of the microstate and its energy relative to the rest. Additionally, classical interactions are nearsighted—they depend on local features, and therefore the energy of larger scale structures can be seen as a sum over local environments. DCL exploits these basic principles, taking advantage of modern machine learning techniques which can estimate a probability distribution from observations and their ability to learn the locality (e.g. screening length) present within a dataset. DCL can predict the relative energy of micro-states at sufficient accuracy that it can be used to reproduce the energetic cost of excitations between states in a size consistent manner. The model consists of two deep neural network topologies which able to learn co-operatively. By using a combination of deep learning and physical constraints, we have shown that full transferability is possible. We expect that there will be many applications of this new method, including optical lattices, the growth of molecules on surfaces, among others.

Data availability statement

The data that support the findings of this study will be openly available following an embargo at the following URL/DOI: http://clean.energyscience.ca/codes. Data will be available from 21 December 2020.

Acknowledgments

The authors acknowledge useful discussions with M. Fraser. IT and JC acknowledge support from the Natural Sciences and Engineering Research Council of Canada (NSERC). JC acknowledges support from the Shared Hierarchical Academic Research Computing Network (SHARCNET), Compute Canada, Google Quantum Research Award, and the Canadian Institute for Advanced Research (CIFAR) AI chair program. This research was supported in part by the National Science Foundation under Grant No. NSF PHY-1748958. Work carried out at NRC was done under the auspices of the AI4D Program. Work by SW was performed as part of a user project at the Molecular Foundry, Lawrence Berkeley National Laboratory, supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02–05CH11231.

Appendix A

Details of neural network topologies, training data, as well as additional tests and examples are outlined below.

A.1. Neural network and training details

We used 5.12×10^7 configurations in our training data. For test data, we sampled all possible energies (8000 samples from each). For the 8×8 Ising model, this is $\approx 5 \times 10^5$. At the extremes, this is equivalent to reselecting certain configurations over and over again (consistent with what occurs physically). Test and train-data were chosen such that they did not overlap.

For the case of the Ising and disordered models, the input to the RNN was a sequence of integer values of the spins (0 or 1), fed one at a time, and the output (label) was the value of the next spin in the sequence. For the Potts model ($Q = 3$), we input vectors of length Q, one at a time as input, and the network was trained to predict the next vector in the sequence.

Our RNN are very simple, consisting of a single gated recurrent unit [19] (we also tried up to four stacked units, which gave only a modest improvement in our results). The size of the hidden state was between 378 and 512 neurons. All RNN models converged quickly—results reported here are based on only 30 epoch (learning rates between 1 and 5×10^{-4} and batch sizes between 800 and 8000, where batch size and learning rate were reduced as system sizes increased. For all models, we used a dropout rate of 0.9. All of our networks are implemented in TensorFlow.

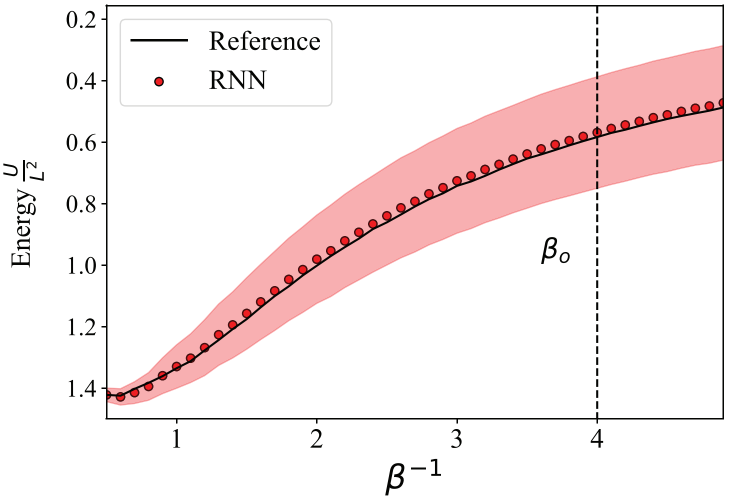

Figure A1. Internal energy per spin as a function temperature for another disordered system (i.e. a different set of random couplings).

Table A1. Entropy predictions of RNN models under different experimental conditions. RNN models were trained on an 8 × 8 lattice. Correlated models were trained with 2 seeds and a sampling period of 1 step per observation, whereas de-correlated models were trained with 32 seeds and a sampling period of 10 steps per observation. Values for correlation time τ are given in table 1. Entropy, $S \approx -\mathbb{E}_{\sigma\sim P_\theta}[\ln(P_\theta(\sigma))]$ was estimated by training six RNN models under identical conditions, sampling 1×10^4 micro-states from each RNN model, then averaging across models with standard deviation included in parenthesis. Reference values were calculated using the exact solution for a finite-sized Ising lattice [18].

Entropy S/L^2	$\beta = 0.1250$ (A)	$\beta = 0.2500$ (B)	$\beta = 0.4407$ (C)
Correlated	0.677 (0.001)	0.621 (0.001)	0.30 (0.03)
De-correlated	0.6769 (0.0003)	0.622 (0.001)	0.28 (0.01)
Reference	0.67 690	0.61 951	0.28 236

For the EDNN, we used 1.92×10^6 random spin configurations for training data and achieved a converged result within only 60 epochs for all models. EDNN models trained using a set of RNN models were trained with 1.92×10^6 configurations per RNN model. As such, the number of epochs was reduced in order to train for the same number of steps. The EDNN was built using a previously reported architecture [15] ($f = 2, c = 1$ and 2 fully connected layers with 32 and 64 neurons with respectively). Throughout, we used rectified linear units as activation functions. Our goal was to use a simple and consistent set of parameters and training; it is very likely that there exist better choices of hyper-parameters than those presented here.

We also note that the RNN we have used here are very simple in form. Recently, attention mechanisms [20, 21] have been shown to improve the performance of sequence models significantly (i.e. reducing the number of needed samples need to achieve fixed fidelity). We expect that more advanced sequence models, including attention only 'Transformer' [22] networks and related models could also be of benefit here, particularly with experimental data.

A.2. Performance under different experimental conditions

In order to determine the importance of training temperature and de-correlation, six data-sets were created on an 8 × 8 lattice with different temperature and correlation assignments. For each data-set, six RNN models were trained. Table A1 details relative performance of RNN models trained on both correlated and de-correlated train-data in predicting entropy. RMSE values for each data-set can be found in table 2.

Additionally, an EDNN corresponding to each data-set was trained for 10 epochs (1.92×10^6 micro-states per epoch) on labels produced by each RNN model (a total of 60 epochs).

A.3. Additional runtime details

In figures 2, 3, 5, A1, and A2; observables were estimated at each temperature with a single MCMC run where 1.6×10^5 steps per spin were used to initially equilibrate the system, after which 4×10^4 steps per spin were recorded. The shaded area was calculated using the standard deviation of over the set of recorded micro-states. In table 3, internal energy and absolute magnetisation were estimated by averaging 16 MCMC runs. Each run first had 1×10^5 equilibration steps per spin, then 4×10^5 steps (per spin) with one in every ten steps recorded. Free energy and Entropy were estimated using 1×10^4 micro-states sampled directly from

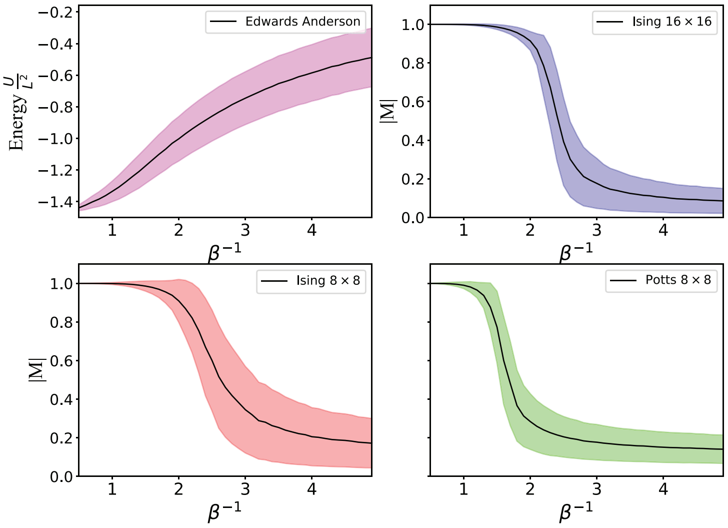

Figure A2. Reference energy and magnetization curves as a function of temperature along with the standard deviation of the MCMC simulation.

Table A2. Average performance of EDNN models under different simulated conditions. EDNN models are labelled according to their corresponding data-set shown in table A1. Observables corresponding to the EDNN models were estimated by running MCMC for 2×10^4 steps/spin (keeping the last 4×10^3 micro-states). These results were averaged over 16 runs (again with standard deviation in parenthesis). Reference values were calculated similarly with the true Hamiltonian and the same number of steps/spin. Samplers labelled with (Corr) were trained with 2 seeds and a sampling period of 1 step per observation. The rest were trained with 32 seeds and a sampling period of 10 steps per observation. Values for correlation time τ are given in table 1. The inverse-temperatures used were $\beta_1 = 0.3407$, $\beta_c = 0.4407$, and $\beta_2 = 0.5407$.

Lattice	Sampler	$\frac{\|M\|}{L^2}$ (β_1)	$\frac{\|M\|}{L^2}$ (β_c)	$\frac{\|M\|}{L^2}$ (β_2)	$\frac{\Delta U_{\beta_2,\beta_1}}{L^2}$
(24×24)	A (Corr)	0.132 (0.005)	0.77 (0.01)	0.947(0.002)	−1.00(0.02)
	A	0.123 (0.008)	0.67 (0.05)	0.945(0.001)	−0.979(0.004)
	B (Corr)	0.121 (0.005)	0.69 (0.05)	0.946(0.001)	−0.990(0.004)
	B	0.120 (0.004)	0.67 (0.03)	0.942(0.002)	−0.98(0.01)
	C (Corr)	0.131 (0.005)	0.71 (0.01)	0.931(0.004)	−0.98(0.02)
	C	0.121 (0.006)	0.69 (0.04)	0.9498(0.0009)	−1.004(0.004)
Reference		0.119 (0.007)	0.68 (0.03)	0.948(0.001)	−0.992(0.004)
(16×16)	A (Corr)	0.192 (0.008)	0.77 (0.01)	0.947(0.002)	−1.00(0.02)
	A	0.180 (0.009)	0.71 (0.03)	0.945(0.002)	−0.980(0.005)
	B (Corr)	0.181 (0.008)	0.72 (0.03)	0.946(0.002)	−0.990(0.005)
	B	0.180 (0.006)	0.69 (0.04)	0.942(0.003)	−0.97(0.01)
	C (Corr)	0.180 (0.007)	0.72 (0.02)	0.930(0.005)	−0.98(0.03)
	C	0.183 (0.007)	0.73 (0.02)	0.950(0.002)	−1.003(0.006)
Reference		0.180 (0.006)	0.71 (0.02)	0.948(0.001)	−0.994(0.005)

an RNN model and averaged over three RNN models. In table A1, entropy was estimated the same way except with six RNN models.

ORCID iD

Isaac Tamblyn https://orcid.org/0000-0002-8146-6667

References

[1] Mills K and Tamblyn I 2018 *Phys. Rev.* E **97** 032119
[2] Chng K, Carrasquilla J, Melko R G and Khatami E 2017 *Phys. Rev.* **7** 031038
[3] Wang C, Zhai H and You Y-Z 2019 *Sci. Bull.* **64** 1228
[4] Tshitoyan V, Dagdelen J, Weston L, Dunn A, Rong Z, Kononova O, Persson K A, Ceder G and Jain A 2019 *Nature* **571** 95
[5] Desgranges C and Delhommelle J 2018 *J. Chem. Phys.* **149** 044118
[6] Wu D, Wang L and Zhang P 2019a *Phys. Rev. Lett.* **122** 080602
[7] Noé F, Olsson S, Köhler J and Wu H 2019 *Science* **365** eaaw1147
[8] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
[9] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 *J. Mach. Learn. Res.* **15** 1929
[10] Nguyen H C, Zecchina R and Berg J 2017 *Adv. Phys.* **66** 197

[11] Valleti S M P, Vlcek L, Vasudevan R K and Kalinin S V 2019 Inversion of lattice models from the observations of microscopic degrees of freedom: parameter estimation with uncertainty quantification (arXiv: 1909.09244)

[12] van den Oord A, Kalchbrenner N and Kavukcuoglu K 2016 Pixel recurrent neural networks (arXiv: 1601.06759 [cs.CV])

[13] Kobyzev I, Prince S and Brubaker M 2020 *IEEE Trans. Pattern Anal. Mach. Intell.* 1

[14] Hibat-Allah M, Ganahl M, Hayward L E, Melko R G and Carrasquilla J 2020 *Phys. Rev. Res.* **2** 023358

[15] Mills K, Ryczko K, Luchak I, Domurad A, Beeler C and Tamblyn I 2019 *J Chem. Sci.* **10** 4129

[16] Wu D, Wang L and Zhang P 2019 *Phys. Rev. Lett.* **122** 2–3

[17] Nicoli K A, Nakajima S, Strodthoff N, Samek W, Müller K-R and Kessel P 2020 *Phys. Rev. E* **101** 5–6

[18] Ferdinand A E and Fisher M E 1969 *Phys. Rev.* **185** 832

[19] Cho K, van Merrienboer B, Bahdanau D and Bengio Y 2014 On the properties of neural machine translation: encoder-decoder approaches (arXiv: 1409.1259 [cs.CL])

[20] Bahdanau D, Cho K and Bengio Y 2014 Neural machine translation by jointly learning to align and translate (arXiv: 1409.0473 [cs.CL])

[21] Kim Y, Denton C, Hoang L and Rush A M 2017 Structured attention networks (arXiv: 1702.00887 [cs.CL])

[22] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 Attention is all you need (arXiv: 1706.03762 [cs.CL])