Extensive deep neural networks

I. Luchak
University of British Columbia

K. Mills and K. Ryczko
Dept. of Physics, University of Ontario Institute of Technology

A. Domurad
University of Waterloo

I. Tamblyn*
National Research Council of Canada and
Dept. of Physics, University of Ontario Institute of Technology & University of Ottawa
(Dated: September 23, 2019)

We present a procedure for training and evaluating a deep neural network which can efficiently infer extensive parameters of arbitrarily large systems, doing so with $\mathcal{O}(N)$ complexity. We use a form of domain decomposition for training and inference, where each sub-domain (tile) is comprised of a non-overlapping focus region surrounded by an overlapping context region. The relative sizes of focus and context are physically motivated and depend on the locality length scale of the problem. Extensive deep neural networks (EDNN) are a formulation of convolutional neural networks which provide a flexible and general approach, based on physical constraints, to describe multi-scale interactions. They are well suited to massively parallel inference, as no inter-thread communication is necessary during evaluation. Example uses for learning simple spin models, Laplacian (derivative) operator, and approximating many-body quantum mechanical operators (within the density functional theory approach) are demonstrated.

INTRODUCTION

Within the past decade, the fields of artificial intelligence, computer vision, and natural language processing have advanced at unprecedented rates. Computerized identification and classification of images, video, audio, and written text have all improved to the extent they are now part of everyday technologies. With the recent advances in hardware acceleration [1], convolutional deep neural networks have been at the forefront of these developments due to their ability to perform "featureless-learning", automatically learning both the features and the mapping between raw data and quantities of interest.

Convolutional deep neural networks are particularly well-suited for applications where the input data structure contains an element of spatial encoding, such as the pixels of an image. Time and time again, convolutional neural networks have outperformed traditional machine learning methods based upon hand-selected features [2] and human-chosen representations [3], and have demonstrated their success at outperforming humans at tasks once thought intractable for computers [4, 5].

Machine learning methods are rapidly being adopted by physicists, chemists, and materials scientists, and have performed well at making predictions in the fields of dynamical mean-field theory, strongly correlated materials, phase classification, and materials exploration and design [6–14]. Deep neural networks have proven their classification power in astronomical applications [15], and particle physics applications [16, 17] but have yet to be widely adopted throughout the physics community for accurate numerical predictions.

In previous work, we showed that deep neural networks have the ability to replace both classical [18] and quantum mechanical operators [19]. In comparison to other machine learning methods, convolutional deep neural networks (DNN) prevailed as the most accurate and best-scaling method for all but the most simple cases. We demonstrated the ability of deep neural networks to learn both the mapping from spin configuration to energy (for the case of the ferromagnetic Ising model and screened Coulomb interaction) as well as magnetization for a classical system. For the case of a confined quantum particle, our deep neural network successfully learned the energy of the ground state, first excited state, and kinetic energy. A similar approach was able to map the structure of two dimensional hexagonal crystal lattice energies computed within the density functional theory framework [20]. All of this was accomplished via "featureless" deep learning, meaning we presented the network with raw, spatial data, without any preliminary attempt at manual feature selection.

Although our previous approach was quite general (i.e. the same basic network architecture was used for all cases), the training process we employed was not transferrable to systems of arbitrary length scales. In practice, this meant that for square, $L \times L$ lattice-gas systems [21], a model trained on 4×4 configurations could not be used on an 8×8 cell (or vice versa). When learning

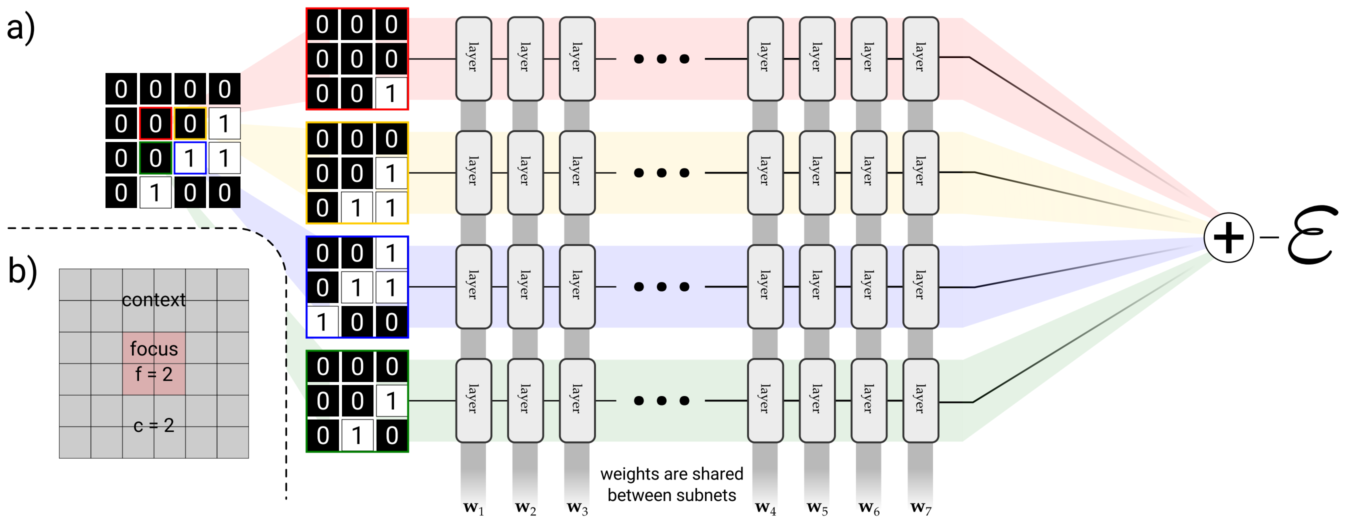

FIG. 1. An input example is decomposed into regularly shaped tiles, with each tile consisting of a focus and context region. a) As a pedagogical example, we expand 4 adjacent tiles comprising a generic binary grid. For this case the focus region, f^2, is 1×1, and the context is unit width ($c = 1$). The tiles are simultaneously passed through identical neural networks with shared weights. The individual outputs are summed, producing an estimate of ε, an extensive quantity. In b), we show an example tile with a focus of $f = 2$, and a context of $c = 2$. The optimal selection of f and c depend on physical contraints determined by the target (learned) function.

the Schrödinger equation, our model could only predict results for potentials described on the same 256×256 numerical grid (40 a.u. $\times$ 40 a.u.) upon which it was trained. Similarly, our model for 2d graphitic systems could only be applied to input structures with dimensions equal to those of the training set. Smaller or larger inputs were not possible. This size limitation was clearly a major shortcoming of our previous approach. Beyond the practical constraints imposed by restriction to a fixed volume or number of particles, from a fundamental standpoint it is unsatisfying to use a model which has no concept of extensivity.

A physical property is extensive if it can be divided among subsystems. A common example is the number of particles in a system. When a sample is divided evenly into two subsystems, the number of particles in each subsystem is halved. This is in contrast to intensive quantities such as temperature, which are unchanged by subdivision or addition of subsystems.

In traditional vision- and audio-based applications of deep learning, maintaining extensivity is not a common requirement as most classification problems (e.g. identification of an animal or shape in an image) are invariant to the physical dimensions of an image (e.g. number of pixels that comprise a cat). Indeed, absolute scale is not normally recorded in a photograph, and therefore scale invariant models are commonplace. Furthermore, photographs, handwriting, and audio recordings too large to be processed by the deep neural network can be resized, cropped, and segmented without destroying the features necessary to make a prediction [22]; there is no absolute spatial scale upon which the label of interest depends. In a physical measurement or simulation, however, the physical scale of a pixel matters critically. Consider the case of an X-ray diffraction experiment where the interference pattern recorded at the detector depends strongly on the wavelength of scattered light and the physical length-scale over which the signal is collected. It is not possible to reconstruct the signal properly unless such parameters are known and are consistent.

In this work, we overcome this limitation, and propose a general method which preserves the extensivity of physical quantities, and also accommodates arbitrary input size. We propose a new organizational structure of a deep neural network which can train and operate on (effectively) arbitrary-sized inputs and length scales while maintaining the physical requirement of extensivity. We call our approach Extensive Deep Neural Networks (EDNN), employing domain decomposition to solve the problem of operator evaluation and learning across length scales. Although domain decomposition techniques have a long history in computer simulation and modelling, here we have taken a new approach, and allow the model itself to identify and self-optimize the overlap of tiles at domain boundaries.

Previous work with deep tensor neural networks and force fields [23] has identified the necessity of extensivity [24], but the use of molecule-specific representations limits its uses. Our method is sufficiently general to allow for applications to any system in which extensivity holds. Furthermore, our method results in a model which can be evaluated on arbitrarily large system sizes.

Extensive deep neural networks

The overall objective of an EDNN is the same as that of our previous work based on convolutional deep neural networks: learn the mapping between an input structure and one or more extensive properties, ε (e.g. total energy, entropy, magnetization, particle number, charge, etc). To date, our input structures have consisted of continuous, regular, real-space grids. We note that some operators may be more conveniently decomposed in one space than another, depending on their locality length-scale (see below), and reciprocal-space representations may be beneficial.

EDNN differ from our previous work based on standard convolutional deep neural networks as they divide the task into a number of tiles (sub-domains) for training and inference. This subdivision is a user-controlled process based on physical constraints discussed in what follows.

EDNN operate on segments of data which we refer to as tiles. Each tile is itself made up of two regions: the focus and context (Fig. 1b). In our implementation, focus regions are orthorhombic structures of dimension f_x, f_y, f_z. In general, there is no requirement to use tiles with orthogonal unit vectors (although the internal details of the convolution operator would need modification if orthogonality is not maintained). For the purposes of discussion, we will assume tiles are cubic (or square for the case of 2d inputs). Surrounding each focus region is a bordering context region of width c.

A trained model takes as input all of tiles simultaneously and returns a value ε corresponding to the extensive quantity of interest. Importantly, we do not assign a fractional contribution to any individual tile. In a heterogeneous system with a complicated (and generally unknown) interaction potential, one should not assign any specific fractional contribution of ε to a fixed volume, region, or, in the case of atomistic systems, species.

For an input structure of size $L \times L$, there will be L^2/f^2 tiles produced during the sub-division process. Overlapping context regions mean that a given feature will appear to the network multiple times within the context of different focus regions. This is by design, and the EDNN will need to learn to compensate for this redundancy. Non-overlapping focus regions mean a given feature will, however, appear in the focus region of only a single tile.

EDNN topology and training

In practice, creating an EDNN requires only slight (albeit fundamental) additions to the network topologies we previously used. In Fig. 1, we show an example EDNN for learning an extensive property present in a 4×4 image. Using both a unit focus and context, we can break such an image into 4 valid tiles (omitting periodic padding in the schematic for simplicity). Each of these 4 tiles is fed *in parallel* to 4 copies of a traditional deep neural network, designed to operate on images of size $(2c+f)^2$, and output a single number corresponding to the contribution to the extensive property associated with the tile. These outputs are summed, imposing the extensivity of the property ε. The critical aspect of this setup is that each subnetwork is identical, i.e. they share the same set of weights. The loss is computed with respect to the final, summed value, and gradients are applied to each of the L^2/f^2 copies of the neural network uniformly and synchronously throughout the training process.

The neural network within the EDNN can be of arbitrary complexity. We have made use of both deep convolutional neural networks as well as shallow fully-connected networks, depending on the complexity of the underlying problem. In fact, there is no strict requirement that any form of neural network be used within the EDNN framework; any machine learning model capable of backpropagation can be used. In our case, we have focused on neural networks. Details of training hyperparameters are provided in Table I. The fully-connected artificial neural network used in training the Ising model is comprised of a fully-connected layer of size $32(f+2c)^2$ (note that $(f+2c)$ is the size of a single tile) fed into a final fully-connected layer of size 1. The convolutional deep neural network used to train the Laplacian and DFT models is constructed from 13 convolutional layers and two fully-connected layers. The first 2 layers are reducing and operate with filter sizes of 3×3 pixels. Each of these reducing layers operates with 64 filters and a stride of 2×2. The next 6 layers are non-reducing, which means they have a unit stride and preserve the resolution of the image. Each of these non-reducing layers operate with 16 filters of size 4×4. The ninth reducing convolutional layer operates with 64 filters of size of 3×3 and a stride of 2×2. The last four of the convolutional layers are non-reducing, and operate with 32 filters of size 3×3. The final convolutional layer is fed into a fully-connected layer of size 1024. This layer feeds into a final fully-connected layer with a single output. The contribution from each parallel tile is summed, and the loss is computed as the mean-squared error between this value and the true value.

We note that parallel branching within neural networks is a common technique. GoogLeNet (a.k.a Inception) [25] uses repeating modules of parallel convolutional layers, and ref. [22] uses multiple preprocessing techniques to feed a variety of data through many branches of a neural network architecture. In both of these methods however, each branch of the neural network has its own set of weights, and learns different features. When a single set of weights are shared between branches [26] the training and inference process can be parallelized efficiently on multiple hardware devices, effectively leading to a multiplicative speedup in training and inference.

Model	Total training examples	Batch size	Network	Optimizer	Learning rate	Loss	Epochs
Ising Model	100,000	2000	ANN	Adam [27]	0.001	MSE	500
Laplacian	10,000	100	DNN	Adam	0.00001	MSE	250
Graphetic DFT	18,515	100	DNN	Adam	0.00001	MSE	500

TABLE I. Parameters used to train the various example models.

Locality length-scale

The spatial extent over which features in the configuration influence the value of an operator is known as locality. In general, operators (such as the Hamiltonian, magnetization, etc.) may be local, semi-local, or fully non-local. For example, the Ising model Hamiltonian considers only nearest-neighbour interactions, but the Coulomb potential, an example of a long-range interaction, is slow to decay.

We define l to be the length scale of an operator's locality. For example, the magnetization of an Ising configuration is fully local ($l = 0$), as computing it requires only knowledge of a single spin at a time. The gradient operator (assuming second-order finite difference method) is a semi-local operator with $l = 2$, as is the Ising Hamiltonian, with $l = 1$. For many systems, such as the Coulomb interaction, there is not a hard cutoff, but typically one expects the importance of a feature to diminish as the distance from the feature increases. For example, in a material, the screening environment (i.e. the importance of many-body effects) has a strong influence on how quickly this attenuation occurs. In metals, it occurs quickly, but in large band-gap insulators the falloff is much more gradual. Even though quantum mechanics involves fully non-local operators, it has been noted that matter is, in practice, near-sighted [28].

The locality of an operator is a crucial parameter when trying to overcome the obstacle of transferability between models of different length scales. For accurate evaluation of a given tile, the EDNN must have access to the data surrounding the primary focus region. This is the motivation for the context region. The context regions are permitted to overlap, and we leave the task of rectifying the inherent double-counting to the deep neural network itself; it must somehow learn to partially ignore the context region.

This of course raises the question of what size focus and context to use in the construction of the EDNN. Size of the focus and context should be chosen with the following considerations in mind:

- $c \geq l$ in order to provide sufficient context to the network
- c should be minimized (while $c \geq l$). When the network is provided with unnecessary context, this is not only inefficient for evaluation, it also makes the weight-optimization process more challenging as the network must learn to ignore a larger fraction of the input signal.
- f should be minimized (while $f \geq c$) as smaller focus regions result in more tiles and consequently a model more amenable to parallelization (each tile can be passed through the network independently).

We note that as the locality length-scale of an operator grows, the requirement that $f \geq c$ means that the EDNN converges to a model identical to our previously published work; there is a single tile, and the context region is simply periodic padding. This is because for fully non-local operators, it is impossible to divide the problem up in the style of EDNN since the full volume is needed to make an inference. Such systems are rare in practice, thankfully. We note that when dealing with operators which have large values of l, it is often useful to recast the problem in reciprocal space, and such an approach could be useful too with EDNN.

Results

As an illustrative example of the method, we have trained EDNN on Ising-like systems, the Laplacian operator, and quantum mechanical simulations (within the density functional theory framework) for different values of focus and context.

Example: The Ising model

The Ising model is a two-state spin (σ) model with $\sigma = \pm 1$ with a total energy Hamiltonian

$$\hat{H} = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j \tag{1}$$

where J is the interaction strength, and the summation is computed over nearest-neighbour pairs ($\langle i, j \rangle$). For $J = 1$ the system is ferromagnetic; it is favourable for neighbouring spins to align. Application of EDNN to the Ising model is particularly instructive because the locality length scale of the Hamiltonian operator is known explicitly; as previously discussed, it is an $l = 1$ operator. The nearest-neighbour interaction means that $c > 1$ provides no additional information to the EDNN. Indeed, including this data makes the task more difficult, as the network must learn to completely ignore these features. In Fig. 2 we report the loss for an EDNN trained to reproduce the total energy of 8×8 Ising model configurations for different values of f and c. As the size of the context region grows beyond the locality length scale of the operator, the learning process is less efficient. Our optimal EDNN trained on the Ising model achieves a MAE (median absolute error) of 0.00148 J on the testing set. In comparison, the Ising model decomposed using $f = 8$ and $c = 1$ produces an error of $0.512J$, sufficiently accurate [18] to reproduce the finite temperature phase transition for which the model is so well-known.

Example: Laplacian Operator

Here we demonstrate the ability of an EDNN to learn an operator *not* associated with discrete spins or atomistic systems, based upon the Laplacian of images.

We define a value $\mathcal{L}$ to be

$$\mathcal{L} = \sum_{i,j} |\nabla^2 X_{ij}|, \tag{2}$$

i.e. the sum of the pixel intensity of the Laplacian of an image X. For the image source, we used the dataset of randomly generated images originating from our work on the Schrödinger equation [19], although the operator could be applied to any smooth scalar field. Since this involves a second order derivative ($l = 2$), computing $\mathcal{L}$ requires additional context. Our trained EDNN with focus of $f = 10$ and context of $c = 2$ trained on this operator achieves a MAE of less than 0.1%.

Example: Density functional theory

For comparison with previous work, we reuse a previously reported training set [20] of 2d crystalline structures. This training set consists of 26,449 structures of crystalline and defect (missing atom) graphene surfaces. We represent atoms using a pseudopotential of the form

$$V(x,y) = \textstyle\sum_{i=1}^{N} Z_i \exp\left(-\frac{[(x-x_i)^2+(y-y_i)^2]}{2\gamma^2}\right). \tag{3}$$

Here x_i, y_i are the coordinates of the i^{th} atom with atomic number Z_i. $\gamma = 0.2$ Å was used as the width of the atomic potential wells (Fig. 3). These pseudopotentials are evaluated on a 256×256 grid representing a 12.5 Å $\times$ 13 Å unit cell containing 60 atoms arranged in a hexagonal lattice. The dataset contains both graphene and hexagonal boron nitride.

For these quantum mechanical systems (computed within the generalized gradient approximation of the density functional theory [29, 30] framework), it is not possible to determine a specific value of l. Within the hierarchy of approximations, the method we used to compute the total energy (PBE [31]) is considered to be a semi-local approximation, since the exchange-correlation potential includes only gradients of the charge density. Other terms within the total energy are fully non-local (although they are subject to screening by the electron gas). Nonetheless, total energy is an extensive quantity and again our EDNN performs favourably compared to previously reported (non-EDNN) results.

Using $f = 64$ and $c = 32$, we achieve a MAE of 0.825 meV/atom on our test set after 500 training epochs, notably better than the MAE of 1.10 meV/atom on a traditional (non-EDNN) network. For this choice of (f, c) our input vector is divided into 16 tiles, enabling an inference speedup of $16\times$ due to the ability to calculate the contribution from each tile in parallel. This is on top of the inherent speedup of evaluating the EDNN compared to DFT. A conservative estimate for this latter speedup is on the order of 1 million in terms of CPU-hours.

The ability of an EDNN to learn to ignore, or compensate for redundant context can be explored by measuring the performance of the model when information is partially obfuscated through the application of a Gaussian blur to select regions of the input. In Fig. 5, we plot the performance of the network when random noise is added within the context region (edge) during inference. As expected, when the noise is added at the periphery of the context region, the network reports very similar values for ε as when there is no noise present, as the blurring encroaches on more context, the predictions become poorer; the neural network is evidently learning to ignore the context region. When a small area in the center of the focus region is blurred, the neural network is able to make accurate predictions, (likely due to the limited amount of information being lost), but as the extent of the blur increases within the focus region, the accuracy of inference suffers greatly.

Transferability to arbitrary input size

EDNN have the capability of making predictions on input configurations of arbitrary size so long as the physical length scale (i.e. the real space extent of a pixel in the input representation), is preserved, and the input size remains an integer multiple of the focus region. To demonstrate this, we test the DFT model trained on a 12.5 Å$\times$13 Å, 256×256 grid ($N = 60$ atoms) on multiple larger domains up to 62.6 Å $\times$ 65.1 Å (1024×1024 grid, $N = 1500$ atoms). The EDNN performs similarly well predicting energies of configurations larger than those in the training set, as seen in Fig. 6, and does so many orders of magnitude faster than conventional numerical methods. This is a powerful feature of EDNN, as one can generate a testing set of many training examples for

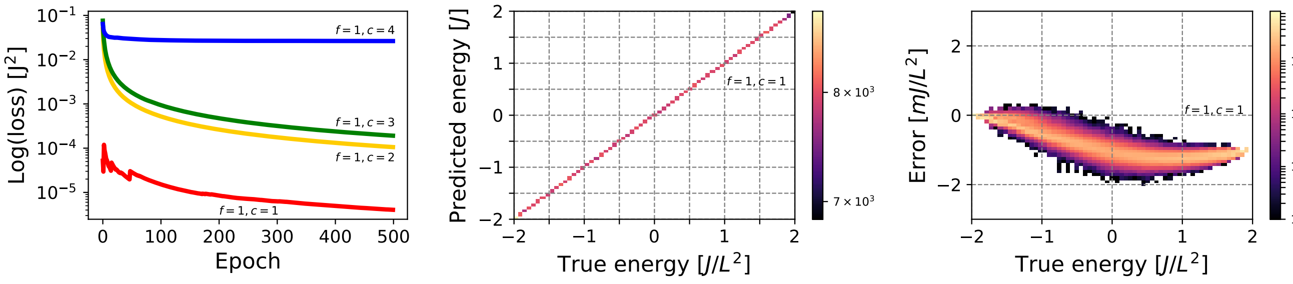

FIG. 2. Left: Loss vs. epoch during training for an EDNN tasked with learning the energy E operator for the 8×8 Ising model using different sized contexts c. Since E is semi-local ($l = 1$), additional information in the form of a larger context region, $c > l$ does not help the network predict values, and in fact makes the training more difficult, as the network must learn to ignore a significant amount of information. Middle: Predicted vs. true energies (per spin) for optimal model. Right: Error (predicted − true energy) vs. true energy for optimal EDNN model ($f = c = 1$).

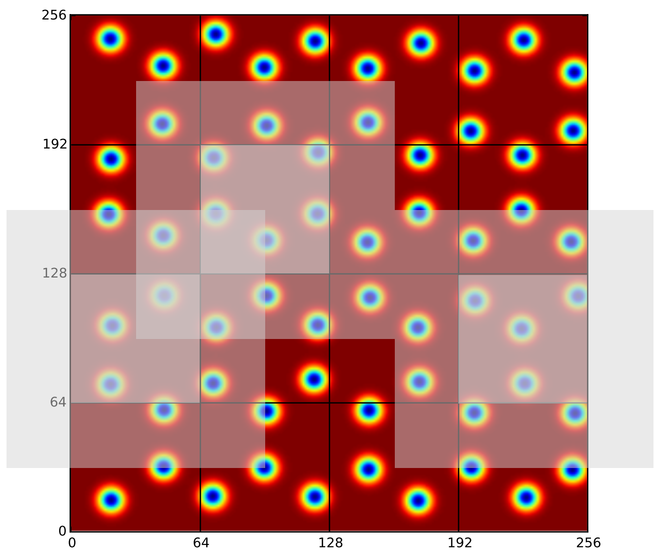

FIG. 3. Decomposition of input image for the quantum mechanical density functional theory calculation using $f = 64$ and $c = 32$. Three tiles consisting of a focus region and context region are highlighted. Overlap in the context region is by design and the EDNN must learn to ignore this overlap in the final reduction of the extensive quantity.

significantly less computational expense, and then apply it to larger systems without the $\mathcal{O}(N^3)$ (or worse) scaling inherent in density functional theory. The evaluation of the EDNN scales as $\mathcal{O}(N)$.

Multi-scale interactions

By construction, the EDNN topology that we have outlined in this work has a limited range of length scales for which predictions are possible; they cannot be aware of features which occur on length scales larger than the tile width. This is because EDNN are tuned so as to maximize the number of minimally sized tiles (for parallel performance) based on the locality length-scale of the operator. In certain cases, however, there may be multiple interactions which give rise to a particular quantity such as the total energy. Take for example proteins in a solvent in the presence of an external electric field. The total energy of a structure is due to a combination of nearsighted covalent interactions, solvent-protein interactions, long-range (screened) dispersive protein-protein interactions, and finally the interactions with the external field. Typically, long-range interactions are less sensitive to local geometries (relative to covalent bonds for example), therefore it is possible to create tiles which are more coarse in their description of the environment. Here, it is advisable to resolve the longer range structures at a lower resolution than is used for the shorter range tiles. Very long-range interactions may also be more efficiently described in an inverse spatial representation (i.e. reciprocal space).

EDNN can treat such systems by modifying the structure of the final output, as shown in Fig. 7. Instead of reporting the output of a single EDNN, multiple EDNN are used as input to a fully-connected layer, where the relative weights of the individual EDNN are learned automatically during the training process. The size of focus region and context are all interaction specific, and tiles may be represented in different (e.g. reciprocal) spaces. In this sense, this multi-level EDNN is itself a reminiscent of a traditional CNN topology.

EDNN advantages

We note that EDNN are particularly well suited for massive parallelization, particularly during inference, since subnets and data can be distributed across hardware - there is no need for communication between them until the final layer. Beyond scalability, EDNN have the feature that they can operate on inputs of arbitrary shape

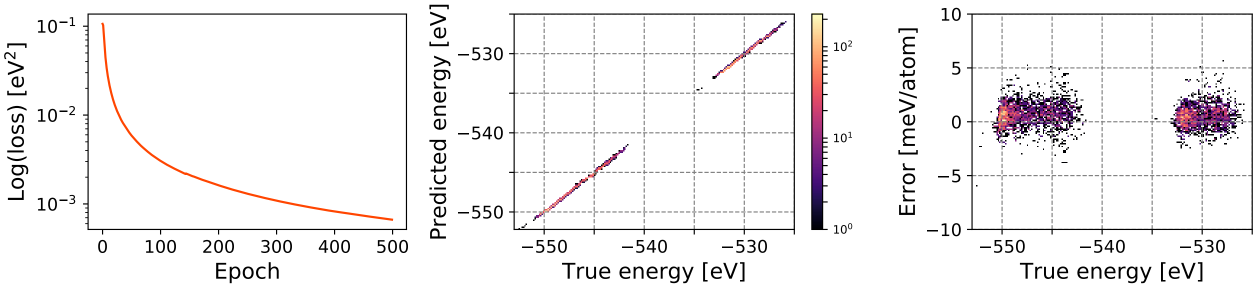

FIG. 4. Left: Training loss monitored during training. Middle: Performance of our EDNN model on a testing set (predicted vs true). Right: Error (predicted − true) vs. true for the EDNN model trained on the total energy as calculated through the density functional theory framework .

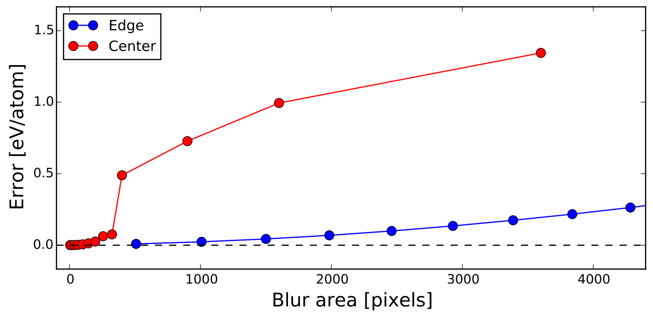

FIG. 5. Resiliance of an EDNN to the addition of obfuscating Gaussian blur. When a constant amount of information (constant area) within the tiles is blurred, examples which had context area blurred result in more accurate inference than examples which had focus blurred. This is evidence that the EDNN is learning to ignore the context regions.

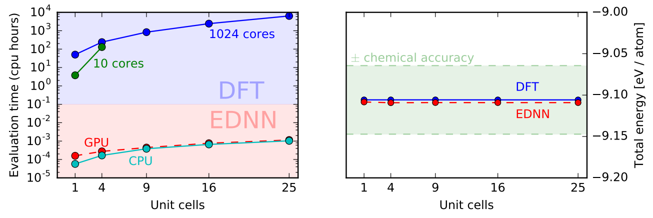

FIG. 6. A single EDNN was trained on a 12.5 Å × 13 Å unit cell. This trained model was used to make accurate predictions on larger unit cells. The inference time for large systems was over 1 million times smaller than the equivalent density functional theory approach (left), and the resulting energy predictions are consistent within chemical accuracy of 1 kcal/mol (right).

and size (to within integer multiples of f). This is particularly useful for treating large-scale mesoscopic structures *in silico*.

In this regard, EDNN can be thought of as the ultimate generalization of a complex, many-body force-field. Features are learned automatically by the EDNN, and domain decomposition is handled intrinsically. Traditional force-fields assume that extensive properties such as the total energy can be expressed as a many-body sum over interacting particles, and in some cases, an implicit solvation environment. While many different models exist, almost all share the common feature of expanding the total energy in terms of neighbour interactions, typically within a fixed cutoff radius. Bond lengths, angles, and partial charges are used as features, and the coefficients of the relative terms are trained against a fixed set of examples. Even methods designed for metallic environments (e.g. the embedded atom method) make use of structural "features" (e.g. the density of neighbours) which are then fed into a feature based decision algorithm. When used for atomistic modelling, EDNN accomplish the same task as a force-field without the requirement hand selecting features. They are also extremely straightforward to implement in parallel, and do not require complex calculations of angles, dihedrals, feature vectors, or neighbor lists for efficient parallelization.

Conclusion

We have demonstrated a new form of deep neural network which can operate on arbitrary sized input and physical length scales while maintaining the extensivity of properties inferred by the network. Networks of this form are particularly well suited to large-scale parallel inference, as the individual components of the input data can be computed independently of one another. Although we have chosen to demonstrate three specific examples, techniques and arguments that we present are quite general, and naturally apply to many problems in physics and image processing.

The network architecture is designed for the task of learning physically extensive properties such as mass, total energy, etc. The method naturally learns the locality length scale of a problem, and is able to optimize the boundaries between sub-domains. EDNN could be made to simultaneously operate at multiple length scales and different spaces if the particular problem demands it.

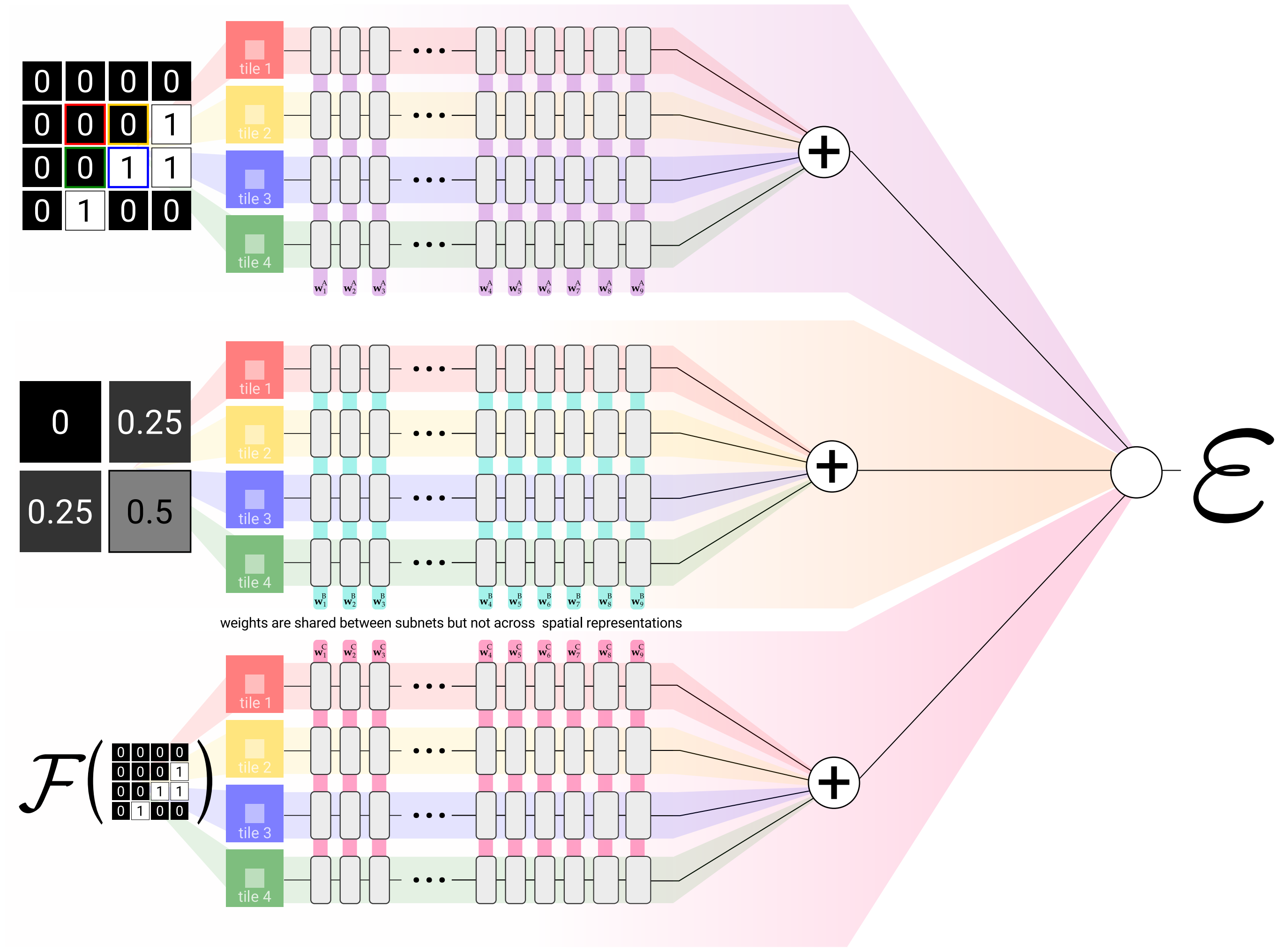
tile 1
tile 2
tile 3
tile 4
weights are shared between subnets but not across spatial representations
0.25
0.5
$\mathcal{F}$
$\mathcal{E}$

FIG. 7. By creating multi-scale EDNN, it is possible to learn operators which act at multiple length scales and coarseness. This form of EDNN is particularly well suited for describing extensive properties in biological systems, where an extensive property may be sensitive to multiple interactions spanning many length-scales.

ACKNOWLEDGEMENTS

The authors acknowledge funding from NSERC and SOSCIP. Compute resources were provided by SOSCIP, National Research Council of Canada, and an NVIDIA Faculty Hardware Grant.

* isaac.tamblyn@nrc.ca

[1] S. Chetlur and C. Woolley, arXiv , 1 (2014), arXiv:1410.0759.
[2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, arXiv , 675 (2014), arXiv:1408.5093.
[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Proceedings of the IEEE **86**, 2278 (1998), arXiv:1102.0183.
[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Nature **529**, 484 (2016).
[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. a. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, Nature **518**, 529 (2015).
[6] O. S. Ovchinnikov, S. Jesse, P. Bintacchit, S. Trolier-Mckinstry, and S. V. Kalinin, Physical Review Letters **103**, 2 (2009).
[7] A. G. Kusne, T. Gao, A. Mehta, L. Ke, M. C. Nguyen, K. Ho, V. Antropov, C.-Z. Wang, M. J. Kramer, C. Long, and I. Takeuchi, Scientific Reports **4**, 6367 (2015).
[8] S. Jesse, M. Chi, A. Belianinov, C. Beekman, S. V. Kalinin, A. Y. Borisevich, and A. R. Lupini, Scientific Reports **6**, 26348 (2016).
[9] P. V. Balachandran, D. Xue, J. Theiler, J. Hogden, and T. Lookman, Scientific Reports **6**, 19660 (2016).
[10] G. Carleo and M. Troyer, Science **355**, 602 (2017).

[11] L. F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, Physical Review B **90**, 155136 (2014), arXiv:1408.1143.

[12] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, arXiv , 1 (2016), arXiv:1609.02552.

[13] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nature Physics **13**, 435 (2017), arXiv:1610.02048.

[14] N. Artrith and A. Urban, Computational Materials Science **114**, 135 (2016).

[15] E. J. Kim and R. J. Brunner, Mnras **000**, 1 (2016), arXiv:1608.04369.

[16] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, and P. Vahle, Journal of Instrumentation **11**, P09001 (2016), arXiv:1604.01444.

[17] R. Acciarri, C. Adams, R. An, J. Asaadi, M. Auger, L. Bagby, B. Baller, G. Barr, M. Bass, F. Bay, M. Bishai, A. Blake, T. Bolton, L. Bugel, L. Camilleri, D. Caratelli, B. Carls, R. C. Fernandez, F. Cavanna, H. Chen, E. Church, D. Cianci, G. Collin, J. Conrad, M. Convery, J. Crespo-Anadón, M. Del Tutto, D. Devitt, S. Dytman, B. Eberly, A. Ereditato, L. E. Sanchez, J. Esquivel, B. Fleming, W. Foreman, A. Furmanski, G. Garvey, V. Genty, D. Goeldi, S. Gollapinni, N. Graf, E. Gramellini, H. Greenlee, R. Grosso, R. Guenette, A. Hackenburg, P. Hamilton, O. Hen, J. Hewes, C. Hill, J. Ho, G. Horton-Smith, C. James, J. J. de Vries, C.-M. Jen, L. Jiang, R. Johnson, B. Jones, J. Joshi, H. Jostlein, D. Kaleko, G. Karagiorgi, W. Ketchum, B. Kirby, M. Kirby, T. Kobilarcik, I. Kreslo, A. Laube, Y. Li, A. Lister, B. Littlejohn, S. Lockwitz, D. Lorca, W. Louis, M. Luethi, B. Lundberg, X. Luo, A. Marchionni, C. Mariani, J. Marshall, D. M. Caicedo, V. Meddage, T. Miceli, G. Mills, J. Moon, M. Mooney, C. Moore, J. Mousseau, R. Murrells, D. Naples, P. Nienaber, J. Nowak, O. Palamara, V. Paolone, V. Papavassiliou, S. Pate, Z. Pavlovic, D. Porzio, G. Pulliam, X. Qian, J. Raaf, A. Rafique, L. Rochester, C. R. von Rohr, B. Russell, D. Schmitz, A. Schukraft, W. Seligman, M. Shaevitz, J. Sinclair, E. Snider, M. Soderberg, S. Söldner-Rembold, S. Soleti, P. Spentzouris, J. Spitz, J. St. John, T. Strauss, A. Szelc, N. Tagg, K. Terao, M. Thomson, M. Toups, Y.-T. Tsai, S. Tufanli, T. Usher, R. Van de Water, B. Viren, M. Weber, J. Weston, D. Wickremasinghe, S. Wolbers, T. Wongjirad, K. Woodruff, T. Yang, G. Zeller, J. Zennamo, and C. Zhang, Journal of Instrumentation **12**, P03011 (2017), arXiv:1611.05531.

[18] K. Mills and I. Tamblyn, (2017), arXiv:1706.09779v1.

[19] K. Mills, M. Spanner, and I. Tamblyn, (2017), arXiv:1702.01361.

[20] K. Ryczko, K. Mills, I. Luchak, C. Homenick, and I. Tamblyn, , 1 (2017), arXiv:1706.09496.

[21] N. Portman and I. Tamblyn, arXiv , 43 (2016), arXiv:1611.05891.

[22] D. Ciresan, U. Meier, and J. Schmidhuber, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on , 3642 (2012), arXiv:1202.2745.

[23] A. K. Rappe, C. J. Casewit, and K. S. Colwell, Journal of the American Chemical Society **2**, 10024 (1992).

[24] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Nature Communications **8**, 13890 (2017).

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 07-12-June (IEEE, 2015) pp. 1–9, arXiv:1409.4842.

[26] D. C. Cirean, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, Procedia Technology **15**, 474 (2011), arXiv:1102.0183.

[27] D. P. Kingma and J. Ba, , 1 (2014), arXiv:1412.6980.

[28] W. Kohn, in *Lindau Nobel Laureate Meetings* (2014).

[29] P. Hohenberg and W. Kohn, Physical Review **136**, B864 (1964).

[30] W. Kohn and L. J. Sham, Physical Review **140** (1965), 10.1103/PhysRev.140.A1133, arXiv:PhysRev.140.A1133 [10.1103].

[31] J. P. J. Perdew, K. Burke, and M. Ernzerhof, Physical Review Letters **77**, 3865 (1996).