Phase space sampling and operator confidence with generative adversarial networks

Mills, Kyle; Tamblyn, Isaac

National Research
Council Canada

Conseil national de
recherches Canada

Canada

Phase space sampling and operator confidence with generative adversarial networks

Kyle Mills*

Department of Physics, University of Ontario Institute of Technology

Isaac Tamblyn†

Department of Physics, University of Ontario Institute of Technology,
University of Ottawa & National Research Council of Canada

(Dated: October 24, 2017)

We demonstrate that a generative adversarial network can be trained to produce Ising model configurations in distinct regions of phase space. In training a generative adversarial network, the discriminator neural network becomes very good a discerning examples from the training set and examples from the testing set. We demonstrate that this ability can be used as an "anomaly detector", producing estimations of operator values along with a confidence in the prediction.

INTRODUCTION

As machine learned potentials come to be used as replacements for conventional force-fields (such as CHARMM, AMBER, etc.) in large-scale and long-time atomistic simulations, it will be important to ensure that these new "model-free" methods provide a level of confidence alongside each of their property predictions.

In the field of self-driving vehicles, it has long been recognized that it is more important to develop algorithms which are constantly aware of their accuracy than models which naively return a prediction irrespective of input. Should a vehicle enter an unfamiliar situation (i.e. one that does not exist within the training set), it is important that it identify the drop in predictive confidence so that the human driver can intervene, or the car can safely come to a stop. Naively continuing to make predictions when an algorithm is making large errors is rarely a good plan, and in some cases would be catastrophic.

A similar problem can occur in a numerical simulation based on a configuration-to-energy or configuration-to-force model. When a user is aware that a model is having difficulty making predictions (perhaps because the system has evolved into a fundamentally new region of configuration space, or is violating a conservation law), it is possible to take action. Such actions could include reducing the integration time step or collecting more training data and producing a new model which includes configurations from the new regime.

Without a confidence metric, there is a risk that a model will become unreliable and begin to return unphysical predictions on new data. For the case of molecular dynamics or a Monte Carlo simulation, these erroneous predictions can result in the system venturing further away from the reference set, exacerbating the problem. The possibility of unphysical predictions outside of the training regime is not a new one in the field of numerical simulation based on fitting. The issue of "transferability" is often discussed in the context of force-fields and pseudo-potential construction.

Model-free machine-learned potentials, however, may return predictions which have errors substantially larger in magnitude than those of a conventional force-field. In a traditional force-field expansion, the internal energy and forces acting on system are generally expressed as a sum over bonded and non-bonded terms. The form of these terms are typically set by simple cases which can either be solved analytically, or where the limits are well known. The CHARMM force-field, for example, contains terms relating to bond angles, dihedral angles, and torsions [1].

The fact that force-fields are usually parameterized in terms of simple physical features such as distance, angle, etc, also make it easier to detect when the simulation is approaching an unfamiliar region of configuration space (bond lengths shrink beyond a threshold, for example).

Any form of numerical fitting procedure will be most reliable when predictions are made within the space of training data (i.e. interpolation). Extrapolation, making predictions for properties which are outside of the manifold where data was collected, will invariably result in higher errors. Force-fields which are based on a simple physical expansion tend to be well behaved when extrapolating, as they implicitly contain information about limiting cases through the choice of expansion. The typical Lennard-Jones (6-12 potential) form used for non-bonded interactions, for example, naturally goes to zero at large distances, and diverges when particles become too close.

While such simple functions have the feature that they require very few training examples to make generally acceptable predictions, the cost of this favourable extrapolation behaviour is that these models typically are much less sophisticated than the underlying physics they are designed to describe, and have an upper bound to their accuracy even within the interpolation region.

Model-free machine learned potentials require substantially more data, and therefore the collection of training data becomes important. The usual concerns with regard to sampling come into play.

In most cases, Nature is able to sample a thermal distribution of states efficiently, although there are notable

exceptions such as the glass transition. *In silico*, sampling the distribution of possible configurations is a very difficult task due to the enormous number of free parameters (e.g. position, spin, charge, etc.) defining a near-infinite configuration space for systems of even a modest number of particles. This "curse of dimensionality", for all but the most trivial systems, precludes directly sampling configuration space at non-zero, finite temperature [2]. Traditionally, Markov Chain Monte Carlo sampling methods have been devised to obtain random samples from an underlying distribution, but these algorithms, such as Metropolis-Hastings [3, 4], depend on the ability to efficiently evaluate both the energy and property of a microstate (E_i) and (O_i), which can in many cases be a very costly computation. Furthermore, this calculation must be carried out repeatedly, many more times than the desired number of final configurations.

Generative adversarial networks

Here we propose the use of a generative adversarial network (GAN) [5] to carry out both the sampling of configuration space and as a means of providing a confidence estimate for predicted properties. Generative models are common approaches to unsupervised machine learning [6]. Generative adversarial networks are typically applied to problems in image processing, such as to image super-resolution [7], image-to-image translation [8], and cross-domain pairings (e.g. pairing shoes with matching handbags) [9]. They have recently been applied to solutions of differential equations involving transport phenomena [10].

Generative models are based on the core premise that in order to synthesize example data, an understanding of the relevant features must be somehow present in the model, and the training procedure attempts to learn these relevant features, usually in the form of optimizing a set of coefficients in a latent variable space. A generative adversarial network is a relatively new unsupervised machine learning technique; it is the combination of two "players", working against each other as adversaries. One player acts as a generator, taking random noise as input and producing examples that fall within a probability distribution. The other player works as a discriminator and learns to tell the difference between examples coming from the generator and true, ground truth examples. These players are trained simultaneously with the generator trying to trick the discriminator, and the discriminator learning how to better tell apart the generator's propositions from the true training examples. A successfully trained generative adversarial network converges to a state where the generator is so good at producing examples that the discriminator cannot tell the generated examples from the ground truth examples. The key to the success of this method is that the generator is pro-

vided with hints about its failure in the form of the gradients, i.e. both the generator and discriminator perform backpropagation using the gradients from the discriminator network. This allows the generator to learn not only whether or not it succeeded in tricking the discrimnator, but effectively gives it access to the reasoning behind its success or failure.

The discriminator can also be provided with relevant labels for the true examples. This acts to help condition the generative adversarial network [11], stabilizing the notoriously sensitive training process [12], but also enabling the trained discriminator to make observable predictions about new data. Some refer to a network trained with such provisions as a Conditional Generative Adversarial Network (cGAN) [10]. In the case where the discriminator is asked to also reproduce the labels (i.e. the cost function includes an error associated with the labels), the discriminator's output can be interpreted as a likelihood that its label prediction is correct, since the discriminator gets exceptionally good at identifying real examples from the distribution. In this way, the discriminator of the trained generative adversarial network can be used as an anomaly detector, providing label predictions alongside a probability that the prediction is correct.

In theory, the generator and discriminator could be any learning algorithm capable of backpropagation. In this work, we use deep convolutional neural networks as their success on the Ising model has been demonstrated in supervised machine learning classification and prediction [13, 14].

The Ising Model

The square two-dimensional Ising model is a well-studied example of a ferromagnetic system of particles [15]. The model consists of an $L \times L$ grid of discrete interacting "particles" which either possess a spin up $(\sigma = 1)$ or spin down $(\sigma = -1)$ moment. The internal energy associated with a given configuration of spins is given by the Hamiltonian $H = -J \sum \sigma_i \sigma_j$ where the sum is computed over all nearest-neighbour pairs $(\langle i, j \rangle)$, and J is the interaction strength. For $J = 1$, the system behaves ferromagnetically; there is an energetic cost of having opposing neighbouring spins, and neighbouring aligned spins are energetically favourable. A measure of "disorder" in the system can be represented by an order parameter known as the "magnetization" M, which is merely the average of all L^2 individual spins. Because both the internal energy and magnetization depend on the discrete spins within the system, both quantities exhibit a discrete distribution. The configuration space of the 8×8 Ising model is of size 2^{8^2}, and thus sampling from all possible configurations is impossible. Given a configuration, while computing properties of interest (e.g. energy) is

generally possible through some theoretical framework, the reverse is not true; it is not possible to obtain a configuration that satisfies a given property. In order to generate examples of a specific energy, one must employ some form of Monte Carlo approach, or devise another clever sampling algorithm [16]. Because of its simplicity and ubiquity, the Ising model has made many recent appearances in machine-learning investigations [2, 13, 17–19]

METHODS

Training datasets

We used a targeted sampling procedure [13] resembling the Metropolis-Hastings algorithm to produce four datasets used for training:

- A low-energy dataset, generated by using a target energy of $-1.3L^2$,

- a high-energy dataset, generated by using a target energy of $+1.3L^2$,

- a bimodal dataset consisting of a equal combination of the previous two datasets, and

- a dataset consisting of an equal number of configurations from each of the 63 possible energy values.

The energy distributions of the first three datasets are displayed as the dashed lines in Fig. 3.

Network

The generator takes an array of random noise z as input and produces examples attempting to mimic the true distribution. Our generator function takes a $2 \times 2 \times 128$ random array sampled from a zero-centered normal distribution with standard deviation of 0.7. It passes the random data through 7 transposed convolution layers with varying kernel, stride, and filter counts arriving at the appropriately shaped 8×8 output, $G(z)$. After every transposed convolutional layer (with the exception of the last two), we include a dropout layer with a 0.7 retention probability, to hopefully prevent the generator from memorizing the training set. Since individual spins of the Ising model can either be 1 or -1, all transposed convolution layers have hyperbolic tangent (tanh) activation, which produces output at an appropriate scale.

The discriminator takes an 8×8 array as input. This can either be $G(z)$ from the generator, or an example x from the training set. Through a series of 9 convolutional layers, and two fully connected layers it reduces the example to a single output D. In the case where we provide labels to the discriminator, the output is of size $1 + N$,

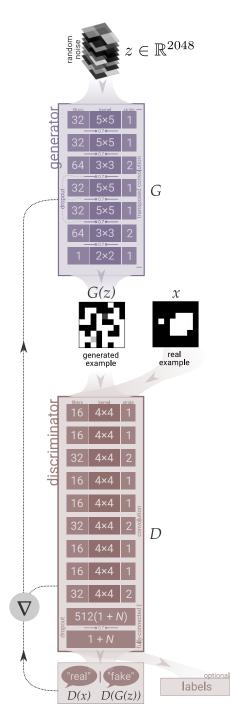

FIG. 1. The generative adversarial network architecture we used. The generator takes random noise, and through a series of transposed convolutions, produces an example. The discriminator takes true examples and "fake" examples from the generator and predicts the probability that each is a real example. The gradients from the discriminator are used by both adversaries to improve their weights.

where N is the number of conditioning labels provided to the discriminator. All layers except for the final layer use rectified linear unit (ReLU) activation. We use a dropout layer with a 0.7 retention probability between the final two fully-connected layers.

We implemented the model in TensorFlow [20]. If trained naively, we discovered that the discriminator learned much more quickly than then generator. When this happened, the discriminator did not provide sufficient feedback (gradients) to the generator and therefore the generator was unable to improve. To remedy this problem, we used a learning rate five times greater for the generator, essentially handicapping the discriminator, and leading to better convergence of both opponents.

The objective of the generator is to minimize the loss function

$$L^{(\mathrm{G})} = -\log(D(G(z)), \tag{1}$$

and the objective of the discriminator is to minimize the loss function

$$L_{\mathrm{GAN}}^{(\mathrm{D})} = \log(1 - D(G(z))) + \log(D(x)), \tag{2}$$

while simultaneously minimizing the mean-squared error of the labels y:

$$L_{\mathrm{LABEL}}^{(\mathrm{D})} = \frac{1}{N_{\mathrm{E}}} \sum_{i=0}^{N_{\mathrm{E}}} \sum_{j=0}^{N_{\mathrm{L}}} \alpha_j \left(y_j^{(\mathrm{predicted})} - y_j^{(\mathrm{true})} \right)^2 \tag{3}$$

where N_{E} is the number of training examples, N_{L} is the number of conditioning labels (in our case $N_{\mathrm{L}} = 2$: energy and magnetization), and α_j is an optional label scaling parameter to account for the relative importance and/or scale when using multiple conditioning labels (for example, the range of Ising energies is always twice that of the Ising magnetizations, so correcly predicting energies would be considered "more important" by the discriminator unless scaled appropriately). The discriminator loss is simply a weighted sum of these two individual losses:

$$L_D = \beta L_{\mathrm{GAN}}^{(\mathrm{D})} + \gamma L_{\mathrm{LABEL}}^{(\mathrm{D})}. \tag{4}$$

We found we obtained the best performance when we initially set $\beta = \gamma = 0.5$. After training for 1000 epochs, we reduced γ to 0.02 and left β unchanged. This permitted the discriminator, which had at this point learned to accurately predict the labels, to "focus" on its ability to differentiate between real and fake outputs, improving anomaly detection.

Contrary to supervised learning, monitoring the loss functions is not an informative method to verify convergence of the model. Both players are simultaneously trying to reduce their own loss functions; a decrease in one leads to an increase in the other. Therefore, during
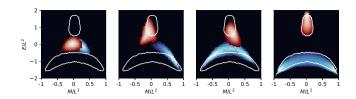

FIG. 2. An example progression of the generator through phase space during the training process for the high energy and low energy target distributions. The generated distributions are represented by the heatmap, and the white boundary represents the target (i.e. training) distribution.

training, we periodically plot the energy and magnetization distributions produced by the generator to verify they are approaching the training distributions.

Training proceeds by passing a batch of 512 images to the discriminator. 256 images are from the training set, and 256 are the output from the generator. Initially, the generator has not learned how to produce realistic looking examples, and outputs nothing more than random noise. We use the Adam optimization method [21] with a learning rate of 0.00002 to update the discriminator's weights. Then it is the generator's turn; we feed a batch of random arrays to the generator. The generator uses its learned filters to process the random information into realistic-looking Ising configurations. Of course, the generator performs very poorly during the first few iterations, but the generator's Adam optimizer receives feedback from the discriminator and is able to improve on its kernels to produce better examples. This process is repeated thousands of times, until the energy and magnetization distributions (Figs. 2 and 3) are deemed sufficiently similar. We used a total of $N_{\mathrm{E}} = 50,000$ training images for each generative adversarial network we trained.

RESULTS

We trained three generative adversarial networks: one on the low-energy distribution, one on the high-energy distribution, and one on the bimodal mixture of high- and low-energy distributions. Shortly after training begins, the generator produces examples composed of random spins, and therefore the distributions of energy and magnetization are mostly centered at the zero-point (Fig. 2). As the generator and discriminator learn from each other, the generator begins to better match the training distributions. In the final frame of Fig. 2, the high-energy distribution (red) matches almost exactly with the high-energy training distribution (white outline), and the low-energy distribution (blue) matches closely with its corresponding training distribution. The third row in Fig. 3 shows the performance of the bimodal dataset. The trained generative adversarial network shown here
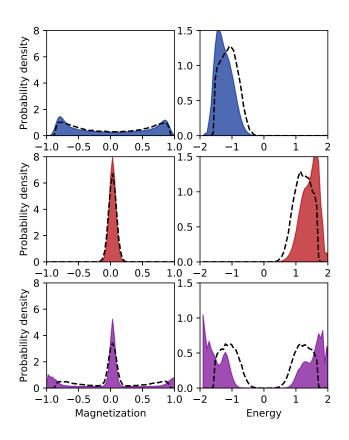
FIG. 3. The energy and magnetization distributions for the training data (dashed lines) and generator output (solid fill). We do not specifically request that the generative adversarial network reproduce these distributions; it must do so automatically. While not matching the exact shape, the generative adversarial network is able to produce examples from the same energy and magnetization ranges, permitting efficient sampling.
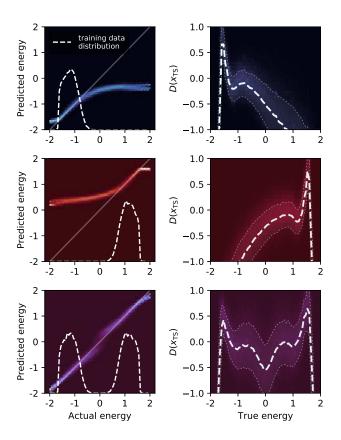

FIG. 4. In addition to its task of opposing the generator, the discriminator can learn to make predictions about operator values if provided with the corresponding labels. Left column: histograms of the predicted vs. true energy for examples from the training set. Right column: when trained using a label, the discriminator's output can be interpreted as a measure of confidence in its prediction. When provided with a uniform distribution of energies across the entire energy range, the discriminator has a much higher average confidence in the region it has not yet seen.

was one of the few training runs that, with moderate success, captured both modes of the distribution. In many training runs, the generator collapsed to either the high-energy or low-energy mode. This "mode-collapse" in generative adversarial networks is a common, and understood phenomena with ongoing research investigating possible solutions [22–25].

In training the generative adversarial network, the discriminator by design becomes very good at identifying configurations which fall outside of the desired distribution, with its output falling roughly on a scale between -1 (the example is suspected to be "fake", i.e. not from the training distribution) and 1 (the example is deemed "real", i.e. likely to be from the training distribution). Provided the discriminator is also able to produce an estimate of the value of an operator (e.g. magnetization and/or energy labels were provided to the discriminator during training), then the output of the discriminator can be interpreted as a confidence of its operator prediction. Thus the generative adversarial network process

produces an anomaly detector, providing both a continuous, real-valued operator evaluation (regression) as well as an indication that the predicted value is correct.

After training the discriminator, we presented it with examples drawn from an even distribution across the energy range. In such an application, one would expect that the discriminator makes predictions both *accurately and confidently* near the region on which it was trained. Arguably more importantly is the ability of the discriminator to output a low confidence when it is unsure of the answer, such as in the regions where training data was not provided. Fig. 4 shows that this is indeed the case; the discriminator outputs the highest confidence in regions where it was provided training examples. In regions devoid of training examples, the discriminator indicates its incompetence in making predictions by outputting a low confidence, precisely what one would desire of a practical prediction engine.

CONCLUSION

We have trained a generative adversarial network to be able to efficiently sample phase space, producing examples from a target distribution without the necessity of *a priori* knowledge of the distribution. A generative adversarial network uses to separate convolutional neural networks, the generator and the discriminator. The generator is tasked with producing examples so realistic that the discriminator cannot tell them apart, and the two networks are trained in tandem. Ultimately, the generator becomes so good at producing realistic examples that the discriminator cannot differentiate between the training examples and the output of the generator. If one provides labelled data to the discriminator, we show that the discriminator can be trained to make accurate predictions as well as indicate its confidence in the predictions, essentially anomaly detection through supervised learning.

The generator of a trained generative adversarial network can be used to produce examples which are larger than those on which it was trained [26]. This has been used to produce textures which have arbitrarily large spatial extent, but are based on a spatially-finite training set. Successful application of this technique to physical systems would be incredibly valuable. As the spatial extent of a physical system increases, the individual features comprising the system do not increase in size, but rather in number (extensivity). The simulation cost for such a system, however, increases dramatically. Therefore, a generative adversarial network trained on a small system which is capable of producing examples from a spatially larger distribution would be incredibly useful, and we propose this as a future application of generative adversarial networks.

Generative adversarial networks are notoriously difficult to train, and the training process can be quite unstable, however ongoing research aimed at stabilizing the training process and making generative adversarial networks more robust will lead to generative adversarial networks as a promising tool for use in the physical sciences.

ACKNOWLEDGEMENTS

---

* kyle.mills@uoit.net
† isaac.tamblyn@nrc.ca

[1] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, Journal of Computational Chemistry **4**, 187 (1983).
[2] J. Carrasquilla and R. G. Melko, Nature Physics **13**, 431 (2017), arXiv:1605.01735.
[3] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, The Journal of Chemical Physics **21**, 1087 (1953).
[4] B. Y. W. K. Hastings, , 97 (1970).
[5] I. J. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, and D. Warde-farley, , 1arXiv:/arxiv.org/abs/1406.2661v1 [https:].
[6] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, (2016).
[7] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, (2016), 10.1109/CVPR.2017.19, arXiv:1609.04802.
[8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, (2017), arXiv:1703.10593.
[9] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, (2017), arXiv:1703.05192.
[10] A. B. Farimani, J. Gomes, and V. S. Pande, **94305** (2017), arXiv:1709.02432.
[11] M. Mirza and S. Osindero, , 1 (2014), arXiv:1411.1784.
[12] T. Salimans, I. Goodfellow, V. Cheung, A. Radford, and X. Chen, , 1 (2016).
[13] K. Mills and I. Tamblyn, (2017), arXiv:1706.09779.
[14] A. Morningstar and R. G. Melko, (2017), arXiv:1708.04622.
[15] L. Onsager, Physical Review **65**, 117 (1944).
[16] N. Portman and I. Tamblyn, Journal of Computational Physics , 43 (2017), arXiv:1611.05891.
[17] I. Luchak, K. Mills, K. Ryczko, A. Domurad, and I. Tamblyn, arXiv (2017), arXiv:arXiv:1708.06686.
[18] L. Wang, Physical Review B **94**, 195105 (2016), arXiv:1606.00318.
[19] A. Tanaka and A. Tomiya, , 1 (2016), arXiv:1609.09087.
[20] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, None **1**, 19 (2016), arXiv:1603.04467.
[21] D. P. Kingma and J. Ba, , 1 (2014), arXiv:1412.6980.
[22] A. Srivastava, L. Valkov, C. Russell, M. Gutmann, and C. Sutton, (2017), arXiv:1705.07761.
[23] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, , 1 (2016), arXiv:1611.02163.
[24] I. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, , 1 (2017), arXiv:1701.02386.
[25] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, , 1 (2016), arXiv:1606.03498.
[26] U. Bergmann, N. Jetchev, and R. Vollgraf, (2017), arXiv:1705.06566.