Adversarial generation of mesoscale surfaces from small scale chemical motifs

Kyle Mills*
Vector Institute for Artificial Intelligence
Toronto, Ontario, Canada
Ontario Tech University
Oshawa, Ontario, Canada
kyle.mills@ontariotechu.net

Corneel Casert*
Ghent University
Ghent, Belgium
corneel.casert@ugent.be

Isaac Tamblyn
National Research Council Canada
Ottawa, Ontario, Canada
isaac.tamblyn@nrc.ca

Introduction – In materials science and materials physics, first-principles theoretical investigation of large and meso scale phenomena is often intractable in part due to the difficulty of sampling configurations from a near-infinite set of microstates. Obtaining valid, large scale, low energy microstates that are likely to occur in a macroscopic ensemble can be a time consuming sampling task [1]. In most cases, Nature is able to sample such distributions 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 number of microstates 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 (or at least the difference in these properties between two states) 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 microstates.

We use our previously reported technique [5], Regressive Upscaling Generative Adversarial Network (RUGAN) (Figure 1a) that can generate unique microstates from the distribution of possible microstates after observing only a very small subset. By conditioning the GAN on an associated quantity, such as the total energy of the microstate, we can "request" that the generated configuration be of a specific energy. Most importantly, our RUGAN can transfer the knowledge learned by observing small scale microstates to generate arbitrarily large scale states beyond that used in training (Figure 1c); it is not limited to small scale generation [6]. This technique enables one to access large scale microstates while only running expensive sampling methods on a small number of small systems.

Methods – We demonstrate the technique on a data set of porous graphene sheets, previously presented in [7]. The study of such systems could be useful in predicting large scale material properties, such as how the strength of a material depends on hole size or hole density (for example), but acquiring a sufficient number of relevant, large scale microstates so as to compute statistics is prohibitively expensive. The sheets are approximately 35 Å $\times$ 35 Å with a random number of randomly-sized holes introduced. To represent these structures compactly, we one-hot encode pairs of atoms and vacancies as a 4-state "spin" (Figure 1b), on a 14×16 lattice, similar to the encoding used in [8].

*The authors acknowledge that both KM and CC contributed equally toward this work

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

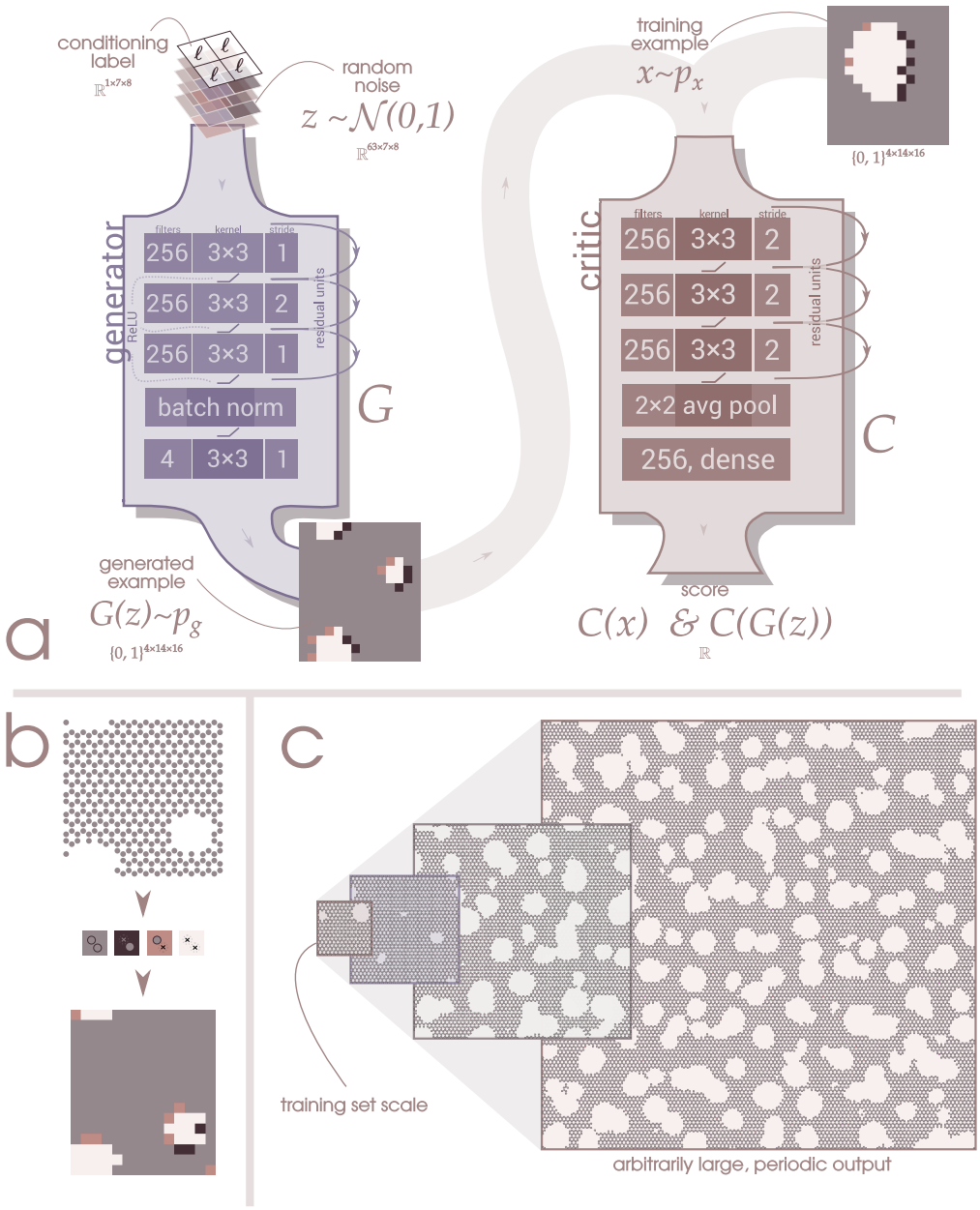

Figure 1: a) Schematic representation of the regressive upscaling generative adversarial network (RUGAN) used in this work. The generator G takes a latent vector as input (concatenated with a conditioning label channel) and, using translationally invariant convolutional layers, produces an output encoding of a microstate. The critic C takes the proposed microstates from the generator and microstates from a training set and learns to assign a score, differentiating whether the input came from G or the training set. b) The encoding used to represent the hexagonal lattice on a 2d rectangular grid. c) Through the adversarial training procedure, the generator of the RUGAN is able to learn relevant features from small scale training examples and extend that knowledge to large scale microstate generation. Since the generator uses only translationally invariant convolutional layers, increasing the size of the input latent vector consequently increases the spatial scale of the output microstate. Importantly, large scale generated microstates respect periodic boundary conditions so they can be easily used with standard electronic structure approaches common in materials simulation.

Our new generative adversarial network [9] is based a conditional Wasserstein GAN [10, 11]. The generator $G(z)$ in our work maps small latent samples to new configurations, a technique inspired by texture synthesis [12]. It is comprised of three residual convolutional layers [13], one batch normalization layer and a final 2d convolutional layer. All convolution operations are implemented with periodic padding. The generator takes as input a block of noise sampled from a Gaussian distribution $z_R \in \mathcal{N}(0,1)^{63\times7\times8}$ concatenated with a label channel $L = \{\ell\}^{1\times7\times8}$ where ℓ is the conditioning value, for a full latent input of $z \in \mathbb{R}^{64\times7\times8}$. Through its several layers, the generator transforms this input into an output of $G(z) \in \{0,1\}^{4\times14\times16}$. Softmax activation is used along the first dimension to collapse the probability distribution to a single, valid encoding of a sheet. The critic C receives encoded configurations x from the training set distribution p_x, as well as the examples $G(z)$ originating from the generator's output distribution p_g. Through a series of layers shown in Figure 1a, it outputs a single scalar value. Through training, the critic is optimized to calculate the Wasserstein distance between p_x and p_g, essentially differentiating between "true" ($x \sim p_x$) and "artificial" ($G(z) \sim p_z$) examples. Through the training process, the generator also learns how to improve its capability, with p_g improving towards matching p_x.

We train the RUGAN for 1000 epochs on a dataset of 65536 porous graphene sheets and their corresponding density functional theory energy, computed using the extensive deep neural network trained and validated in [7]. We use the Adam optimizer [14] with a learning rate of $\alpha = 10^{-4}$ and hyperparameters $\beta_1 = 0, \beta_2 = 0.9$, and $\epsilon = 10^{-8}$ to minimize the WGAN loss function [15, 16] with regularizing parameters $\lambda_1 = 10$ and $\lambda_2 = 2$ for the gradient penalty and consistency term respectively. To stabilize the prediction of the Wasserstein distance, we perform 10 weight updates (10 batches) of the critic for every update of the generator. After 1000 epochs, the generator has learned to approximate the data distribution, and we can use $G(z)$ to generate examples that appear to come from p_x. Furthermore, since G was conditioned on the energies of the microstates, we can request microstates of a specific energy.

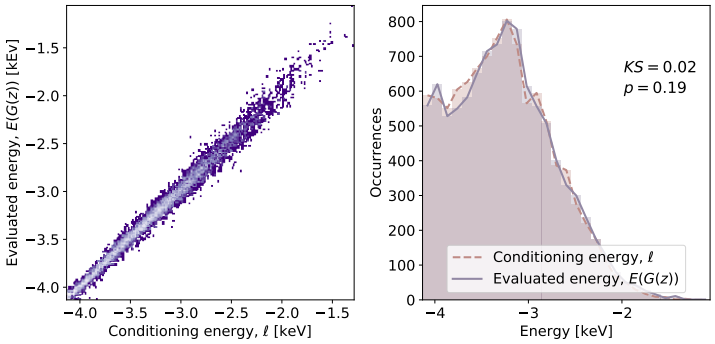

Figure 2: On the left we plot a heatmap of the energy of the generated microstates from the RUGAN (computed using the extensive deep neural network of [7]) against the energy requested by means of the conditioning label. The color indicates the density of each point, with brighter colors occurring more often. On the right, we plot a histogram of the two distributions. The diagonal trend on the left and the closely matching distributions on the right confirm that the generator has indeed learned to produce configurations that match the requested energy values at the same length scale as the training data.

Upscaling – In the generator, we intentionally use only translationally invariant layers (e.g. convolutional layers) as well as periodic padding. Doing so enables the generation of larger scale microstates that adhere to the periodic boundary conditions merely by changing the size of the input random noise. For example, if we feed a $z \in \mathbb{R}^{64\times16\times14}$ noise block into the generator, $G(z)$ will produce a microstate $G(z) \in \{0,1\}^{4\times28\times32}$, representing a 70 Å $\times$ 70 Å sheet, four times as large as the states on which it was trained.

Results – We train the RUGAN on the dataset of porous graphene sheets from [7], conditioning the generator on the provided total energy (computed under the density functional theory framework), normalized by the surface area. Once trained, the job of the critic is complete and we focus our attention on what the generator has achieved. We feed 10 000 randomly generated latent blocks z_R

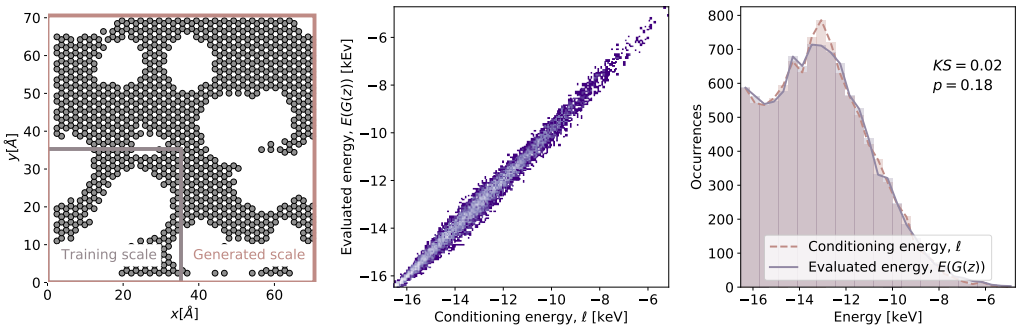

Figure 3: On the left we show an example large scale configuration, superimposed with an indicator of the size of the training set. In the center we plot a heatmap of the energy of the larger microstates produced by the RUGAN generator (computed using the extensive deep neural network of [7]) against the energy requested by means of the conditioning label. The color indicates the density of each point, with brighter colors occurring more often. On the right, we plot a histogram of the two distributions. The diagonal trend on the left and the closely matching distributions on the right confirm that the generator has indeed learned to produce configurations that match the requested energy values.

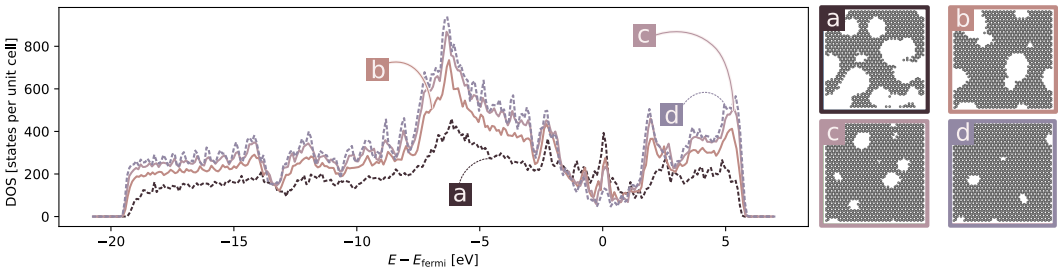

Figure 4: Our periodic configurations are fully compatible with standard materials simluation protocols. Here we plot the density of states for four of the large scale distributions that were created using the generator trained on small scale structures.

and conditioning values ℓ randomly sampled from the training data into the trained generator and receive 10 000 encoded microstates. To verify that the generated examples do indeed represent the energy requested through the conditioning, we used the extensive deep neural network, E, trained independently of this work [7] to evaluate the energy of each generated microstate. Thus we can compare the distribution of ℓ to the distribution of $E(G(z))$ to investigate the hypothesis that the generator has successfully learned the training distribution p_x. We plot the two distributions in Figure 2. In doing a KS test on the two distributions we get a KS score of 0.02 with a p-value of 0.19 and can conclude that the two distributions match.

Next, we repeat the process, feeding in larger random blocks to the generator, which in turn produces spatially larger output configurations. We again use the extensive deep neural network to evaluate the "DFT-energy" of the generated configurations and compare it to the energies requested through conditioning. Similar to before, the distributions match (KS score 0.02, p-value 0.18, Figure 3) and we can conclude that the generator is successful at producing output at scales larger than the scale on which it was trained.

We can use the generated microstates as starting configurations for further density functional theory calculations. For example, in Figure 4 we plot the density of states for four different configurations generated through the RUGAN approach.

A trained RUGAN enables the fast and accurate generation of energetically relevant microstates after being provided a small number of training examples, enabling rapid sampling of configuration space. Furthermore, the RUGAN, with its translationally invariant and periodic design empowers

one to sample the configuration space of large scale structures, a task that is traditionally infeasible, providing a basis for the investigation of large scale structures.

Acknowledgements
We would like to thank Jannes Nys and Tom Vieijra for useful discussions. We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan V GPU used for this research.

References

[1] Frank Noe, S Olsson, J Kohler, and H Wu. Boltzmann generators: Sampling equilibrium states ofmany-body systems with deep learning. *Science*, 365(September):1147, 2019.

[2] Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, feb 2017.

[3] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, jun 1953.

[4] B Y W K Hastings. Monte Carlo sampling methods using Markov chains and their applications. pages 97–109, 1970.

[5] Corneel Casert, Kyle Mills, Tom Vieijra, Jan Ryckebusch, and Isaac Tamblyn. Generative adversarial deep learning for augmenting large scale ultracold atom experiments. *Submitted*, 2019.

[6] Yuan Dong, Dawei Li, Chi Zhang, Chuhan Wu, Hong Wang, Ming Xin, Jianlin Cheng, and Jian Lin. Inverse Structural Design of Graphene/Boron Nitride Hybrids by Regressional GAN. 1, aug 2019.

[7] Kyle Mills, Kevin Ryczko, Iryna Luchak, Adam Domurad, Chris Beeler, and Isaac Tamblyn. Extensive deep neural networks for transferring small scale learning to large scale systems. *Chemical Science*, 10(15):4129–4140, aug 2019.

[8] Yuan Dong, Chuhan Wu, Chi Zhang, Yingda Liu, Jianlin Cheng, and Jian Lin. Bandgap prediction by deep learning in configurationally hybridized graphene and boron nitride. *npj Computational Materials*, 5(1):26, dec 2019.

[9] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. pages 1–9, jun 2014.

[10] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. pages 1–7, nov 2014.

[11] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[12] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv:1611.08207*, 2016.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein GANs. *Advances in Neural Information Processing Systems*, 2017-December:5768–5778, 2017.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. pages 1–15, dec 2014.

[15] Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect. *arXiv:1803.01541 [cs, stat]*, March 2018. arXiv: 1803.01541.

[16] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved Training of Wasserstein GANs. *arXiv:1704.00028 [cs, stat]*, March 2017. arXiv: 1704.00028.