## Optical lattice experiments at unobserved conditions with generative adversarial deep learning

Corneel Casert,<sup>1,\*</sup> Kyle Mills<sup>(a)</sup>,<sup>2,3</sup> Tom Vieijra,<sup>1</sup> Jan Ryckebusch,<sup>1</sup> and Isaac Tamblyn<sup>(a)</sup>,<sup>2,3,4,†</sup>

<sup>1</sup>Department of Physics and Astronomy, Ghent University, 9000 Ghent, Belgium <sup>2</sup>Vector Institute for Artificial Intelligence, Toronto, Ontario M5G 1M1, Canada <sup>3</sup>Ontario Tech University, Oshawa, Ontario L1G 0C5, Canada

<sup>4</sup>Department of Physics, University of Ottawa, Ontario K1N 6N5, Canada

(Received 22 September 2020; accepted 5 August 2021; published 20 September 2021)

Optical lattice experiments with ultracold atoms allow for the experimental realization of contemporary problems in many-body physics. Yet, devising models that faithfully describe experimental observables is often difficult and problem dependent; there is currently no theoretical method which accounts for all experimental observations. Leveraging the large data volume and presence of strong correlations, machine learning provides a novel avenue for the study of such systems. It has recently been proven successful in analyzing properties of experimental data of ultracold quantum gases. Here we show that generative deep learning succeeds in the challenging task of modeling such an experimental data distribution. Our method is able to produce synthetic experimental snapshots of a doped two-dimensional Fermi-Hubbard model that are indistinguishable from previously reported experimental realizations. We demonstrate how our generative model interprets physical conditions such as temperature at the level of individual configurations. We use our approach to predict snapshots at conditions and scales which are currently experimentally inaccessible, mapping the large-scale behavior of optical lattices at unseen conditions.

DOI: 10.1103/PhysRevResearch.3.033267

# I. INTRODUCTION

Ultracold atoms provide a controlled environment for the study of emergent phenomena in many-body physical systems-including high-temperature superconductivity, many-body localization, and topological quantum phasesand also have applications in fields such as cosmology and quantum chemistry [1–4]. Finding a unifying theoretical or numerical model able to create configurations with statistics that conform to experimental observations allows one to make predictions for conditions that currently cannot be experimentally realized. Here we propose using generative deep learning to create such a model, which learns to produce configurations indistinguishable from those experimentally obtained and can also make predictions for configurations at larger scale or at unobserved control parameter values. The capability of *discriminative* machine learning to analyze physical systems has been well established, both for data obtained through numerical simulations [5-12], and for experimental observations through electronic quantum matter visualization [13], quantum gas microscopy [14,15], or momentum-space density images [16]. In these machine learning applications, a neural network is trained to predict properties y of configurations **x**, i.e., learn the conditional probability  $p(\mathbf{y}|\mathbf{x})$ . Examples include the characterization of phases of matter, and efficiently calculating properties of individual microstates. The use of generative machine learning is relatively unexplored for experimental data; it is a more complex problem as it requires the modeling and sampling of a probability distribution  $p(\mathbf{x}, \mathbf{y})$  not known *a priori*. Yet, generative learning provides a particularly attractive approach as it relies on automatic pattern recognition-and hence does not focus on the reproduction of a specific physical quantity. The latter can introduce bias, or require prior knowledge about the system. Recently, Boltzmann generators [17] were trained on the energy functional of many-body systems to directly generate low-energy equilibrium configurations and were shown to overcome rare-event sampling problems in simulations. A particular class of generative models, namely restricted Boltzmann machines, has seen use as efficient variational ansätze for quantum many-body wave functions [18–24].

Here we show that generative deep learning can be used to represent and sample the distribution of snapshots of an experimental realization of the Fermi-Hubbard model with ultracold atoms in an optical lattice. We show that our generative model is able to generalize in two ways: (i) it can create microstates with properties for which no training data is available, and (ii) it is able to create samples at a much larger scale (or "upscale") than the training examples. The former is relevant for systems where obtaining configurations is only numerically or experimentally feasible for a limited set of system properties. As only small-scale samples are required for training, the latter generalization enables

<sup>\*</sup>corneel.casert@ugent.be

<sup>&</sup>lt;sup>†</sup>isaac.tamblyn@uottawa.ca

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

efficient sampling of configurations at scales inaccessible to traditional methods, either due to excessive computational cost or experimental limitations on the imageable system size.

In Sec. II we describe the Fermi-Hubbard model and theoretical frameworks to which we compare our predictions. The developed generative model is introduced in Sec. III. Sec. IV contains the results obtained with this method, where we focus on its generalization properties. A conclusion and perspective is offered in Sec. V. Appendix A contains details on the architecture and optimization of our generative model, where we also discuss some of its limitations. Additional results on the experimental Fermi-Hubbard data set are provided in Appendix B; we also demonstrate how our approach works for simulated data in Appendix C. Appendix D describes the theoretical frameworks to which we compare our results.

# **II. FERMI-HUBBARD MODEL**

The Fermi-Hubbard model is of particular interest as it is suggested to hold the key to understanding high-temperature superconductivity [3,27,28]. The Fermi-Hubbard model is described by the Hamiltonian

$$\widehat{\mathcal{H}} = -t \sum_{\langle \mathbf{i}, \mathbf{j} \rangle, \sigma} (\hat{c}^{\dagger}_{\mathbf{i}, \sigma} \hat{c}_{\mathbf{j}, \sigma} + \text{H.c.}) + U \sum_{\mathbf{i}} \hat{c}^{\dagger}_{\mathbf{i}, \uparrow} \hat{c}_{\mathbf{i}, \uparrow} \hat{c}^{\dagger}_{\mathbf{i}, \downarrow} \hat{c}_{\mathbf{i}, \downarrow}, \quad (1)$$

where  $\hat{c}^{\dagger}_{\mathbf{i},\sigma}(\hat{c}_{\mathbf{i},\sigma})$  is the creation (annihilation) operator of a spin  $\sigma \in \{\uparrow, \downarrow\}$  on site **i**. The first term corresponds to tunneling between neighboring lattice sites i and j. The second term accounts for the onsite interaction between fermions with opposite spin. Here we consider the strongly correlated regime, where  $U/t \gg 1$ . The Fermi-Hubbard model can be experimentally realized with ultracold atoms trapped in an optical lattice [Fig. 1(a)] [1–3,29,30]. Quantum gas microscopy provides site-resolved snapshots of these quantum states, imaging either the total atom distribution or that of a single spin species. Holes and doubly occupied sites (doublons) are detected as empty sites [Fig. 1(b)]. Recently, the use of discriminative machine learning for classifying snapshots of ultracold atomic gases has been investigated [14,15]; we use the same data set [31] as in Ref. [14] to train our generative model. We first only consider the low-temperature data, which are obtained at temperature  $T = (0.65 \pm 0.04)J$ , where J = $4t^2/U$ , and the ratio U/t = 8.1(2). Here, a mixture of the two lowest hyperfine states of <sup>6</sup>Li is trapped in a two-dimensional optical lattice. Site-resolved measurements of the occupation in the optical lattice are obtained with high-resolution quantum gas microscopy [32]. The experimental snapshots of the atomic distributions consist of a circular region of 80 sites. In total, 8 822 images of the atomic distributions with both spin components present and 17 233 with one spin component removed are available. We augment the data set by also including these samples rotated by multiples of  $90^{\circ}$ .

At half filling, the Fermi-Hubbard model is theoretically relatively well understood and maps to the Heisenberg model with superexchange coupling  $J = 4t^2/U$ . In this case, long-range antiferromagnetic (AFM) correlations are present throughout finite-size systems for temperatures  $T \ll J$ . The Fermi-Hubbard model is not as well understood when straying



FIG. 1. (a) Experimental realization of the Fermi-Hubbard model with ultracold atoms in a two-dimensional square optical lattice. (b) Our generative model is trained on site-resolved snapshots of quantum states, obtained at a fixed temperature and for a range of hole doping values  $\delta$ . (c) Sign-corrected spin correlations  $C_s(r)$  of Eq. (2) as a function of the hole doping  $\delta$  for nearest neighbors (r = 1). The correlations obtained with our generative model "RUGAN" (blue squares, 1000 snapshots generated for each doping as to obtain statistics similar to the experiment) are consistent with experiment (orange circles). We also show results obtained through sprinkled holes (dashed line) and geometric string theory (solid line). Our generative model is not optimized on data corresponding to  $\delta \gtrsim 0.24$  (shaded area). The calculation of this observable is described in Appendix D. (d) Same as in panel (c) but now for next-nearest neighbors ( $r = \sqrt{2}$ ).

from the half-filling regime through the addition of holes. The motion of these holes displaces strings of spins and hence hides the AFM order observed at half filling; this has recently been experimentally observed [26]. These observations can be partially explained in the framework of geometric string theory, in which strings of displaced spins are added to a background of experimental snapshots produced at half filling [33] (see Appendix D for a detailed definition of the geometric strings, and the inset of Fig. 2(d) for a graphical depiction).

Note that considering the motion of the holes is crucial to describe experimentally observed hidden order, as this is not correctly accounted for by randomly adding holes to a state obtained at half filling ("sprinkled holes").



FIG. 2. (a) Through a series of residual convolutional layers, the generative neural network transforms a latent sample to a new snapshot of a single spin species at a prescribed value of the doping  $\delta$ . (b) The distribution of snapshots created by the RUGAN (blue, color online) cannot be distinguished from the experimental snapshots (orange) with other unsupervised machine learning methods. Here, we show a dimensionality reduction with the UMAP algorithm [25] of an equal number of synthetic and experimental snapshots for three different values of  $\delta$ . (c) The number of strings exceeding a length of 2 per system site and (d) the average length (sites) of the string patterns detected by the algorithm described in Ref. [26], as a function of the doping  $\delta$ . The string statistics obtained with RUGAN (with which 1000 snapshots were created for each doping value) are consistent with experimental observations. The intermediate doping values  $0.09 \leq \delta \leq 0.23$  in the shaded area are not included during training of the generative model. (Inset) Illustration of the string pattern formation due to a hole moving through an AFM ordered state, which leaves behind a trail of displaced spins. (e) and (f) Same as in (c) and (d) but now the RUGAN is not provided with data corresponding to large doping values  $\delta \geq 0.24$  during training and is required to extrapolate to unseen values of  $\delta$ . For the extrapolation regime, we show the statistics obtained by two independently trained RUGANs.

#### **III. GENERATIVE MODELING**

Here, we develop a generative model which allows for efficient augmentation of the experimental data set at requested doping values  $\delta$  and offers predictions for observables at conditions where no experimental data is available. Our generative approach, called "regressive upscaling generative adversarial network" (RUGAN), allows for the direct generation of new microstates at any given scale and with desired properties after being shown a training data set of small-scale configurations. RUGAN is based on generative adversarial networks (GANs) [34-36], which are the combination of two neural networks competing against each other as adversaries. One network, G, acts as a generator, taking samples z randomly drawn from a latent space as input and transforming these to create new samples with distribution  $\mathbb{P}_{g} \sim G(\mathbf{z})$ . The other network works as a critic, learning to discern between samples coming from the generator and the example data set with distribution  $\mathbb{P}_r$ . These networks are trained simultaneously such that the distance between  $\mathbb{P}_r$  and  $\mathbb{P}_{g}$  is minimized, at which point the critic cannot tell the generated samples from the reference training set. The generator and critic of a GAN can be conditioned on additional information such as known properties of individual samples [35]. This allows one to control the region of configuration space from which the generator produces new configurations. GANs are typically used in image processing such as superresolution [37] or cross-domain pairings [38]. They have also been applied to the Ising model [39,40], scalar field theories [41,42], inverse molecular design [43], in particle physics [44] and in studies of transport phenomena [45]. To allow the RUGAN to create samples of arbitrary size, our generator is designed such that it consists solely of translationally equivariant operations that can be applied to latent inputs of any size. Along with technical details on the design, training and upscaling capability of the RUGAN in Appendix A, we give a simple illustration and results on classical spin models in Appendix C.

## **IV. RESULTS**

#### A. Antiferromagnetic Correlations

We now demonstrate to the use of generative deep learning to tackle the challenging task of modeling an experimentally obtained sample distribution. To this aim, we first train a RU-GAN on experimental snapshots of a doped two-dimensional Fermi-Hubbard model on a square lattice, and condition it on the doping ratio  $\delta$ . The output of RUGAN is a series of synthetic snapshots at prescribed doping values [Fig. 2(a)]. Alternate unsupervised machine learning methods, such as dimensionality reduction, are not capable of discriminating between the experimental samples and those created by RUGAN [see Fig. 2(b)]. We then apply the same analysis procedure to these samples as the experimentally obtained snapshots [26]. We first consider the decay of AFM correlations upon increasing the doping value at low temperatures [26,32,46,47] in snapshots created by our model by evaluating the sign-corrected spin correlation for sites at relative displacement **r** with  $r = ||\mathbf{r}||_2$  and  $|| \cdot ||_p$  denoting the *p*-norm:

$$C_{s}(r) = 4(-1)^{\|\mathbf{r}\|_{1}} \left( \left\langle \hat{S}_{\mathbf{i}}^{z} \hat{S}_{\mathbf{i}+\mathbf{r}}^{z} \right\rangle - \left\langle \hat{S}_{\mathbf{i}}^{z} \right\rangle \left\langle \hat{S}_{\mathbf{i}+\mathbf{r}}^{z} \right\rangle \right).$$
(2)

Correlations between nearest neighbors (r = 1) and nextnearest neighbors  $(r = \sqrt{2})$  measured on snapshots created by our generative model are shown in Figs. 1(c) and 1(d) along with the theoretical predictions by both the geometric string theory [33] and sprinkled holes [26] (longer-range correlations for  $r > \sqrt{2}$  are discussed in Appendix B). A RUGAN can produce synthetic snapshots with high AFM spin correlations  $C_s(r)$  at small hole doping values  $\delta$ , and correctly models the decay of  $C_s(r)$  in the snapshots upon increasing  $\delta$ . Unlike these theoretical models, our generative model is able to accurately capture correlations across all hole-doping values—even at values for which it is not optimized.

#### **B.** String statistics

We now show that the synthetic configurations generated by a RUGAN also capture the more intricate hidden order present in the experimental snapshots, quantified by the number and average length of strings of spins displaced by hole motion as a function of  $\delta$ . More information on the determination of these observables can be found in Appendix D. These statistics are shown in Fig. 2 for the synthetic samples, along with the experimentally determined values, and the predictions made by the theoretical frameworks developed in Refs. [26,33]. Importantly, RUGAN is not tasked with reproducing specific experimental observables during training, but rather only attempts to minimize the distance between the experimental and synthetic data distributions, without our intervention. Hence, we do not introduce any bias during the optimization of our model, which explains why RUGAN generally succeeds in providing excellent agreement with all experimental observables shown here; rather than only a restricted set of observables as for the theoretical frameworks. Remarkably, the RUGAN is able to accurately generate snapshots that exhibit the correct string statistics even at values of  $\delta$  on which it is not optimized. In Figs. 2(c) and 2(d) we show that when the RUGAN is trained on a subset of the experimental snapshots restricted to the extrema of experimentally available doping values, it still succeeds in generating configurations at intermediate  $\delta$  with string statistics matching closely with those observed in experiment. Exploiting this feature dramatically reduces the small number of experimental observations required to train the RUGAN. Excluding the largest  $\delta$  from the training set allows us to assess the RUGAN's ability to extrapolate its learned knowledge of configurations with smaller  $\delta$ . The observation in Figs. 2(e) and 2(f) that the string statistics obtained with this extrapolation procedure again match closely with experimental results, showcases the RUGAN's capability to predict complex correlation patterns, and indicates that the synthetic configurations can serve as a benchmark for quantitative comparison with theoretical developments at conditions where no experimental data is available. We stress that, though still performing better than theoretical frameworks for doping values not included in training, the reliability of extrapolated predictions does of course eventually decrease as predictions are made at doping values further away from those used during training. In Figs. 2(e) and 2(f), we show an example of the variation in extrapolated values that occurs far from training conditions. We give a more detailed analysis of the extent to which extrapolation is found to work in Appendix A, and provide additional results in Appendix B.

#### C. Temperature conditioning

Additionally, we investigate how our generative neural network models the physical conditioning values. By keeping the random latent channels fixed, the effect of the conditioning label on the generation of configurations can be singled out and visualized. We use this feature to explore how a RUGAN models the transition between an AFM ordered and disordered phase by training it on data obtained at half filling ( $\delta = 0$ ) for a range of temperatures while conditioning it on the temperature T. After training, we synthesize new configurations at these temperatures and calculate the AFM correlations to verify that the crossover between both phases is correctly represented [Figs. 3(a) and 3(b)]. Then, selecting a fixed noise configuration  $\mathbf{z}$  while varying the temperature conditioning label allows us to visualize how our RUGAN distinguishes the effect of temperature on the level of individual configurations. In Figs. 3(c) - 3(f), we show that it achieves this by altering a small number of disconnected sites, as indicated by the shading-which causes the AFM correlations to correctly decrease upon raising the temperature-rather than performing large-scale changes.

#### **D.** Upscaling

A current limitation in experiments with quantum gas microscopy on ultracold atoms is the number of sites that can be imaged. Experimental snapshots of the optical lattices currently typically consist of less than 100 sites. The upscaling ability of the RUGAN provides a useful precursor of what could be observed at large scale while experimental realizations containing more lattice sites are still unavailable. Given the success in performing this upscaling for classical spin models (see Appendix C), and the excellent agreement with experimental data for small-scale samples, these configurations can serve as a benchmark for future experimental observations of larger optical lattices. In Fig. 4(a), we show such configurations obtained by our RUGAN consisting of approximately four times as many sites as the experimental examples. As RUGAN enables us to synthesize a distribution of large-scale snapshots, we can use these to predict the behavior of physical observables at scales that are experimentally inaccessible. In Fig. 4(b) we show the string count per site and average string length as a function of doping  $\delta$ , measured on snapshots created by the same RUGAN as in Figs. 2(e) and 2(f). The limitations of the upscaling procedure are discussed in Appendix A.

### **V. CONCLUSION**

In this paper, we have demonstrated the great potential of generative learning with RUGAN on experimental snapshots of a doped Fermi-Hubbard model. Whereas current theoretical frameworks of this model often focus on the description of a number of specified observables, such as spin-spin corre-



FIG. 3. (a) Decay of sign-corrected spin correlations  $C_s(r)$  of Eq. (2) with temperature *T* for nearest neighbors (r = 1) for configurations obtained through experiment and with RUGAN. (b) Same as panel (a) but now for next-nearest neighbors  $(r = \sqrt{2})$ . (c–f) Snapshots generated at the temperatures indicated in panel (a). Each row is created with the same noise instance **z**, and the marked sites indicate the change in a configuration upon increasing the temperature.

lators or hidden order, the power of generative learning lies in its unbiased learning procedure. Especially at large doping values, the synthetic snapshots created by RUGAN provide a better match with experimental observations than alternate theoretical predictions. On top of that, the RUGAN provides the ability to sample snapshots at experimentally unobserved doping values or at larger spatial scales, and thus opens the door for quantitative testing of new phenomenological, analytical, and numerical models on synthetic data under conditions where no experimental data is available. We can also visualize how RUGAN chooses to represent concepts such as temperature. Due to its expressive power, we plan to apply RUGAN to even more complex strongly correlated systems where theoretical descriptions pose real challenges and are less developed than for the Fermi-Hubbard model.



FIG. 4. (a) Example snapshots of a doped Fermi-Hubbard model created at a scale that has approximately four times as many sites as the training examples (orange line) in Fig. 1(b). (b) The number of strings exceeding a length of 2 per system site (circles) and the average length (sites) of the strings (squares), for large-scale snapshots created with the RUGAN. For each value of  $\delta$ , we use the RUGAN of Figs. 2(e) and 2(f) to create 1000 large-scale snapshots.

## ACKNOWLEDGMENTS

We thank Christie S. Chiu, Phil de Luna, Dennis Klug, Roger Melko, Jannes Nys, and Michael Schuurman for helpful discussions. Work at NRC was carried out under the auspices of the MCF and AI4D Program. The computational resources and services used in this work were partly provided by the VSC and the Flemish Government—Department EWI. T.V. is supported as an "FWO-aspirant" under Contract No. FWO18/ASP/279. K.M. and I.T. acknowledge support from NSERC. The Titan V used for this research was donated by the NVIDIA Corporation. Computer code for reproducing results of this paper is available at Ref. [57].

# APPENDIX A: MODEL AND OPTIMIZATION

#### 1. Generative adversarial networks

A generative adversarial network [34] consists of a generator, which maps latent samples to new configurations, and a critic, which measures the distance between the distributions of the real samples  $\mathbb{P}_r$  and the generated samples  $\mathbb{P}_g$ . The distance metric used by the critic in our work is the Wasserstein-1 or Earth-Mover distance (WGAN), which allows for stable training [36]. The Wasserstein-1 distance W between  $\mathbb{P}_r$  and  $\mathbb{P}_g$  is given by

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} [\|\mathbf{x} - \mathbf{y}\|_2], \qquad (A1)$$

with  $\Pi(\mathbb{P}_r, \mathbb{P}_g)$  the set of joint distributions whose marginals are  $\mathbb{P}_r$  and  $\mathbb{P}_g$ . As the infimum in Eq. (A1) is intractable, the

Kantorovich-Rubinstein duality is used to write

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leqslant 1} \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[f(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[f(\tilde{\mathbf{x}})], \quad (A2)$$

with the supremum taken over all 1-Lipschitz continuous functions. The training objective can now be formulated as a minimax game between two neural networks, the critic C and generator G:

$$\min_{G} \max_{\|C\|_{L} \leqslant 1} \mathcal{L}(C, G), \tag{A3}$$

where

$$\mathcal{L}(C,G) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[C(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[C(\tilde{\mathbf{x}})].$$
(A4)

During training, the critic is optimized to maximize  $\mathcal{L}$ , and thus finds an estimate for  $W(\mathbb{P}_r, \mathbb{P}_g)$ . The generator then learns to minimize this distance, so that its sampled distribution  $\mathbb{P}_g$  is similar to  $\mathbb{P}_r$  [Fig. 8(a)].

The critic in the WGAN construction needs to be 1-Lipschitz continuous over the whole domain to find a correct estimate for the Wasserstein distance. While enforcing this constraint everywhere is impracticable, a good approximation can be obtained by adding two regularizing terms to the loss function of Eq. (A4):

$$\mathcal{L}' = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[C(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[C(\tilde{\mathbf{x}})]$$
(A5a)

$$+\lambda_1 \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} C(\hat{\mathbf{x}})\|_2 - 1)^2]$$
(A5b)

$$+\lambda_2 \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[\|C(\mathbf{x} + \boldsymbol{\delta}_1) - C(\mathbf{x} + \boldsymbol{\delta}_2)\|_2], \quad (A5c)$$

and as new objective,

$$\min_{G} \max_{C} \mathcal{L}'(C, G).$$
(A6)

A differentiable function is 1-Lipschitz continuous if it has gradients with at most unit norm over the whole domain. Hence, the first of these regularizing terms [Eq. (A5b)] penalizes the critic such that the norm of the gradient equals one for samples  $\hat{\mathbf{x}}$  sampled from  $\mathbb{P}_{\hat{\mathbf{x}}}$  [48]. As enforcing this over the whole support domain is intractable, the distribution  $\mathbb{P}_{\hat{\mathbf{x}}}$  is sampled uniformly on straight lines between data points in the training distribution  $\mathbb{P}_r$  and generated distribution  $\mathbb{P}_g$ .

The limitation on how  $\mathbb{P}_{\hat{\mathbf{x}}}$  is sampled leaves much of the domain unconstrained. In particular, Lipschitz continuity over the manifold that supports the training distribution  $\mathbb{P}_r$  is not properly enforced until the distribution  $\mathbb{P}_g$  lies close to  $\mathbb{P}_r$ . To alleviate the lack of Lipschitz-constraint on the training manifold, the term [Eq. (A5c)] is added to the loss function that explicitly enforces Lipschitz continuity close to it [49]. Lipschitz continuity requires that for two points  $\mathbf{x}'$  and  $\mathbf{x}''$  close to one another, the distance  $||C(\mathbf{x}') - C(\mathbf{x}'')||_2$  is bounded by a constant. Enforcing this criterion is accomplished by perturbing every training data point twice, with small random perturbations ( $\delta_1$ ,  $\delta_2$ ), and minimizing the distance between the critic output of these configurations. In practice, the perturbation of samples is achieved by adding dropout [50], which disables nodes in a layer with a specified probability, to several layers of the critic and feeding it the same data point twice.

### 2. Conditioning and upscaling

To condition the generator on system properties, we provide it with both a random sample from the latent space and labels describing the desired properties (e.g., the energy, magnetization, or doping of configurations) as input [35]. Meanwhile, the critic is shown this same label for the generated configurations, while receiving the exact label for real samples. The critic uses this additional label during its estimation of the Wasserstein distance, prompting the generator to adapt by creating configurations that have features accurately described by their label. Note that since the critic and generator find efficient internal representations of these quantities through training, they are never explicitly evaluated during training. This implies that upon conditioning the generation on expensive operators, the majority of computational effort is devoted to the generation of training data for the small-scale samples.

A generator built with translationally equivariant layers can be applied to latent samples of any size and hence allows for upscaling by applying it to larger inputs. Motivated by the locality of the interactions in the models studied here, the generator is designed as a convolutional network. By construction convolutional neural networks have a limited receptive field defined by the size of the convolutional kernels and the depth of the network, and can not account for arbitrary-range interactions. Hence, the network depth required to accurately model a physical data set is a proxy for the typical correlation lengths present in the individual configurations. However, long-range interactions could also be efficiently included by making use of attention layers [51]. Once optimized on the training data, such a generator can then efficiently create configurations containing a substantially larger number of sites. The creation of new large configurations only requires a single pass through a convolutional neural network and hence comes at a small cost. The validity of the upscaling procedure is heavily dependent on whether the physical length scales of the upscaled configurations and the smaller training samples are identical. This also means that the failure of creating configurations at a larger scale (i.e., resulting in different statistics than computationally or experimentally obtained) is indicative of the appearance of physics at a new, larger length scale-a typical example is the divergence of the correlation length at a critical point which cannot easily be captured by RUGAN. Another advantage of the fully convolutional design of the RUGAN is that it provides an optimized starting point for the training of larger systems. Additionally, making generative models interpretable, i.e., understanding how it models the interactions between the degrees of freedom [52–54], would lead to new insight for further theoretical developments.

## 3. Architecture and training

With the eye on stable training [55], both the critic and generator are implemented as deep residual convolutional neural networks, where residual functions with respect to the layer inputs are learned. We add a hyperbolic tangent activation function to the last layer of the generator to obtain a valid output representation. As we want to generate configurations with an approximately circular shape for the optical lattice data set, we manually apply a mask that sets the borders to zero after creating a square configuration with the generative network. For the hyperparameters in our WGAN implementation, we use  $\lambda_1 = 10$  and  $\lambda_2 = 2$  in Eq. (A5) [48,49]. The



FIG. 5. Relative error on the average string length for RUGANs trained on different subsets of the data. For the green line, the doping values on which we train are cut off to only include the 12 lowest values [same as in Fig. 2(f)]; this is 9 and 7 for the blue and orange line, respectively.

dropout rate is set to 25% for two layers in the critic to evaluate Eq. (A5c). The weights of the neural networks are optimized with the ADAM optimizer [56], with learning rate  $\alpha = 10^{-4}$  and the exponential decay rates for the first and second moment estimates to  $\beta_1 = 0$  and  $\beta_2 = 0.9$ . To gain a more reliable estimate of the Wasserstein distance before updating the generator's weights, we train the critic on 20 batches for every generator training iteration. For the classical spin models we use a 10:1 ratio of critic updates to generator updates. Each model is trained for 2000 epochs.

### 4. Model selection

Once training is complete, we use each model epoch to generate a number of configurations for every conditioning label. As the Wasserstein distance quickly stabilizes (after a couple of epochs) during training, we resort to a different selection criterion to decide on which model is ultimately deployed. Here, we evaluate the relative squared deviation for the four experimental observables shown in the main text (weighted by the experimental error) and select the model that minimizes this deviation. Naturally, when data corresponding to certain conditioning labels is not shown during training (i.e., for the demonstration of interpolation and extrapolation), data generated at these labels is also not used for model selection, and we select the model that performs best on the training regime. The results obtained when interpolating between the lowest and highest doping values [Figs. 2(c) and 2(d) of the main text] consistently match well with experimental values across the model epochs. As anticipated, extrapolation to doping values substantially larger than those used during training leads to less robust results [see Figs. 2(e) and 2(f)]. We also demonstrate this in Fig. 5, where we provide the RU-GAN with even smaller subsets of the available doping values than in Figs. 2(e) and 2(f). Though again more accurate than theoretical predictions, the string statistics measured on the synthetic configurations start to deviate from the experimental



FIG. 6. Sign-corrected spin correlations  $C_s(r)$  as in Figs. 1(c) and 1(d), but now for (a) r = 2 and (b) r = 4.

observations for the highest doping values, even for the model epochs that perform best on the training set. Details on the network architectures (e.g., number of layers and channels) can be found in our open-source implementation at Ref. [57].

## **APPENDIX B: ADDITIONAL RESULTS**

In Fig. 6, we show the AFM correlations  $C_s(r)$ , similar to Figs. 1(c) and 1(d), but now for larger distances r = 2 and r = 4. This indicates that RUGAN is also able to capture longer-range correlations. In Fig. 7, we have trained a



FIG. 7. (a) The average length (sites) of the string patterns, and (b) the number of strings exceeding a length of 2 per system site as a function of the doping  $\delta$ . The RUGAN results are obtained with 1000 generated spin configurations for each  $\delta$ . The doping values  $\delta \leq 0.09$  in the shaded area are not included during training of the generative model.



FIG. 8. (a) The setup of the RUGAN method. The generator *G* (blue) transforms latent samples of size *l*, and optionally labels  $\{\lambda\}$ , to proposed configurations of size 4*l*. The critic *C* (pink) is used to measure the distance between the distributions of generated and real configurations. This information is used to update both neural networks. (b) The generator consists of translationally equivariant convolutional operations and therefore can be applied to inputs of arbitrary size, allowing for the creation of configurations at different spatial scales. (c) The *l*th hidden layer of the residual convolutional networks transforms an input  $\mathbf{x}_l$  into  $\mathbf{x}_{l+1}$ , which can have a different spatial scale than the input. When reducing the spatial scale, the rescaling operation is an average-pooling layer with kernel size 2 and stride 2. When increasing the spatial scale, the rescaling consists of nearest-neighbor interpolation with a scale factor of 2. The convolutional operations with their corresponding kernel sizes are shown in blue. Note that we do not use batch normalization in the critic [48]. (d) Ising configurations created by a RUGAN, trained on L = 64 configurations, at different spatial scales. The training size is shown in red on the larger configurations.

RUGAN model on a data set of configurations where the smallest doping values are excluded. Note that the vast majority of the data is obtained at low doping. Hence, these results are obtained with much fewer training samples, demonstrating that our model works well with a few thousand training examples. The RUGAN results in the training region match very well with experimental observations. The average string length in the extrapolation region  $\delta < 0.09$  is well reproduced by the RUGAN. Although the RUGAN models the string count per site excellently in the training regime, it appears that the geometric string model describes the data better than RUGAN in the extrapolation regime. We stress that this model works better at small doping by construction, as it alters the experimental configurations obtained at zero doping and hence has to make relatively few changes at small  $\delta$ . The reduced quality in performance of this particular RUGAN model in extrapolating towards low doping is possibly due to the much smaller training data set size.

## APPENDIX C: TRAINING ON SYNTHETIC DATA

#### 1. Classical Ising model

We now give a detailed illustration of our framework on the prototypical classical Ising model on a twodimensional square lattice of length L with Hamiltonian  $\mathcal{H} =$  $-J \sum_{\langle \mathbf{i}, \mathbf{j} \rangle} s_{\mathbf{i}} s_{\mathbf{j}}$ , where the sum runs over nearest-neighbor spins and set J = 1. For N binary spins  $s \in \{-1, +1\}$ , the configuration space of the Ising model has a dimension of  $2^N$ ; sampling configurations with desired properties directly from this space is intractable for all but small system sizes. Here we show that this task can be accomplished with a RUGAN [Fig. 8(a)] by training it on data of small-scale Ising configurations, and conditioning it on the energy and magnetization density  $m = \sum_{i} s_i / N$  of each training example. The conditioning allows for the efficient creation of microstates with desired properties from the high-dimensional configuration space as well as the creation of microstates with conditioning labels for which no training examples are available. We implemented a modified form of umbrella sampling called "targeted sampling" [10] to obtain a training set with a uniform energy distribution. This sampling method resembles the Metropolis-Hastings algorithm in structure, but instead of seeking low-energy states, we target specific energies, accepting configurations that move toward the target energy, and rejecting ones (with a Gaussian probability) that move away. In doing this, we efficiently collect examples across the energy spectrum. For the results of Figs. 9(a)-9(d), we have trained a RUGAN on a data set of Ising microstates restricted to high and low energy values and magnetizations near zero, and use it to sample the entire space of possible energy and magnetization combinations. The generator only makes large errors on the conditioning label combinations with a relatively small density of states. Note that spin-flip symmetry is not enforced, which could potentially decrease the errors shown here. The accuracy in this conditioning is retained when using the upscaling property [Fig. 8(b)] to create configurations at much larger scales [Figs. 8(d), 9(e)-9(h)]. This implies that we can greatly accelerate sampling of uncorrelated large scale synthetic configurations, as costly simulations are only needed for a small subset of the configuration space and only of small-scale microstates.

#### 2. Transverse-field Ising model

Additionally, we trained a RUGAN on a data set of a one-dimensional transverse-field Ising (TFI) model. The corresponding Hamiltonian reads

$$\widehat{\mathcal{H}} = -J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^x, \qquad (C1)$$

where the first sum runs over nearest-neighbor sites. We sample configurations of this model using path-integral quantum Monte Carlo, where we decompose the thermal density matrix using a Trotter decomposition. The resulting configurations are (1 + 1)-dimensional, where the second dimension is along imaginary time. We sampled the (1 + 1)-dimensional



FIG. 9. (a) During training of a RUGAN on classical Ising model data, we condition the RUGAN on the energy and magnetization of each configuration, and only show it examples with labels in the regions enclosed by dashed lines. After training, it is requested to create configurations with all possible labels; we show the median absolute error in the magnetization. The results are obtained by sampling 100 configurations at possible combinations of the energy and magnetization per site with energy spacing  $\Delta E/L^2 = 1/64$  and magnetization spacing  $\Delta m = 1/1024$ . Results shown here are for the training system size L = 64. (b) Same as panel (a), but now showing the median absolute error in the energy. (c) Average error and its standard deviation at fixed energy [indicated by the blue line in panel (a)] made by the RUGAN for generating configurations with a requested value of the magnetization. (d) Average error and its standard deviation at fixed magnetization [indicated by the blue line in panel (b)] made by the RUGAN for generating configurations with a requested value of the energy. (e-h) Same as panels (a-d), but now with configurations created at a larger scale L = 256, containing 16 times more spins than the training examples.

configurations using a local Markov-chain Monte Carlo model. The TFI was simulated for 10 spins at J = 1and a range of 11 values for the transverse field  $h \in$  $\{0.5, 0.6, \dots, 1.4, 1.5\}$ , and an inverse temperature set to  $\beta =$ 4. The time step of the Trotter decomposition is set to 0.1, resulting in configurations of  $10 \times 40$  spins. We generated 10 000 samples for each value of h and trained a RUGAN conditioned on h on a data set which only includes the three highest and three lowest values of h. Once trained, we use the RUGAN to generate data both at the values of the transverse field included in training, as well as those intermediate values on which the RUGAN was not trained. As shown in Fig. 10, the RUGAN and QMC data are indistinguishably distributed according to a dimensionality reduction—even for the interpolation regime in h.

## APPENDIX D: HIDDEN ORDER IN THE TWO-DIMENSIONAL FERMI-HUBBARD MODEL

To benchmark the string patterns in the experimental data and our RUGAN, we compare them to the predictions of the frameworks of sprinkled holes and geometric string theory [33]. Thereby we apply an analysis procedure identical to the one used to describe experimental observations [26]. For the sake of completeness, we here briefly describe those models but refer to Refs. [26,33] for more detailed information.

## 1. Sprinkled holes and geometric string theory

Sprinkled holes is a model for the doped Fermi-Hubbard model in the limit of noninteracting holes. To obtain snapshots at different dopings, we start from experimentally obtained snapshots at half filling and add holes on random positions until the doping matches the requested one.

In geometric string theory holes do not interact with each other but do interact with the surrounding spins. First, a single hole is placed at a random position on the lattice. The dynamics of the hole can be described by introducing an effective Hamiltonian and an effective Hilbert space for a single string [26,33]. The Hilbert space consists of string patterns, which can be viewed as paths without loops on a Cayley tree with coordination number z = 4. Using the frozen spin approximation and  $U \gg t$ , the strings can be modeled by the effective Schrödinger equation

$$t\sum_{s}^{z-1}\psi_{l+1,s} + t\psi_{l-1} + V_{l}\psi_{l} = E\psi_{l},$$
 (D1)

where  $\psi_l$  is a shorthand notation for a path on the Cayley tree of length l, and  $\psi_{l+1,s}$  denotes the string obtained by continuing the string  $\psi_l$  along one of the z-1 directions on the Cayley tree. The parameter t is the coupling constant for tunneling between string lengths (equal to t in the Fermi-Hubbard Hamiltonian) and  $V_l$  is an effective potential. The effective potential  $V_l = (dE/dl)l + g\delta_{l,0}$  consists of a linear tension with magnitude dE/dl and an attractive term with magnitude g. Solving this string model for finite temperature yields a string length distribution p(l). The snapshots are then created by starting from experimental snapshots at half filling and adding strings on random positions in the lattice—with a length according to the string length distribution p(l)—until



FIG. 10. The distribution of configurations of an L = 10 TFI created by RUGAN (blue) cannot be distinguished from the training examples with other unsupervised machine learning methods. Here, we show a dimensionality reduction with the UMAP algorithm [25] of an equal number of RUGAN and QMC configurations. Note that the RUGAN was not trained on h = 1.1.

the desired doping is reached. More details can be found in Refs. [26,33].

## 2. String detection algorithm

In Fig. 2 the number of strings and their average length as a function of doping are compared between the RUGAN, experiment and the theories explained above. The detection algorithm of these strings is applied to snapshots where one of the two spin species and doublons are removed, and is performed in multiple steps [26]. The geometric strings describe the deviation between the doped Fermi-Hubbard snapshots and a checkerboard state. Hence, the first step involves selecting a window (here with a diameter of seven sites) for each configuration with the highest staggered magnetization. Using a window with a diameter smaller than the configuration itself negates some of the finite temperature effects. For a given doping, 60% of the resulting windows with the highest staggered magnetization are kept for further analysis. In the next step, each of these windows is compared to a checkerboard state. The strings are then identified as deviations from this checkerboard, and the string-pattern length distribution  $p^{\delta}(l)$ is measured.

As for the string count shown in Fig. 2, only those patterns of length greater than two are included as to negate the contribution from quantum fluctuations such as doublon-hole pairs [26]. The average string lengths  $\bar{l}(\delta)$  are calculated from the string length histograms as  $\bar{l}(\delta) = \sum_{l} l p^{\delta}(l) / \sum_{l} p^{\delta}(l)$ .

# 3. Spin-spin correlators

One way to assert the validity of the snapshots created by the RUGAN described is to verify whether the sign-corrected two-point spin correlator [Eq. (2)] matches well with experimental results. Here,  $\hat{S}_i^z = \frac{1}{2}(\hat{n}_i^{\uparrow} - \hat{n}_i^{\downarrow})$  with  $\hat{n}_i^{\sigma}$  the number operator for spin  $\sigma$  on site **i**. The spin correlator can be calculated from the experimental snapshots as [46]

$$C_{s}(r) = (-1)^{\|\mathbf{r}\|_{1}} \left[ 2 \sum_{\sigma \in \{\uparrow,\downarrow\}} \left( \langle pp \rangle_{R\sigma} - \langle p \rangle_{R\sigma}^{2} \right) - \left( \langle pp \rangle_{NR} - \langle p \rangle_{NR}^{2} \right) \right],$$
(D2)

where the spatial indices are dropped for simplicity. Here, p denotes a singly occupied site, the expectation value  $\langle \cdot \rangle_{NR}$  is taken over images where neither spin species was removed, and  $\langle \cdot \rangle_{R\sigma}$  over images where the spin state  $\sigma$  was removed. To calculate the expectation values  $\langle \cdot \rangle_{NR}$ , we train a second RUGAN on a data set of snapshots containing both spin species, and also condition it on the doping. Here, the doping conditioning can be explicitly checked, as the doping  $\delta \approx 1.22(0.905 - n_s)$ , where  $n_s$  is the density of singly occupied sites [26].

- M. Lewenstein, A. Sanpera, V. Ahufinger, B. Damski, A. Sen(De), and U. Sen, Ultracold atomic gases in optical lattices: Mimicking condensed matter physics and beyond, Adv. Phys. 56, 243 (2007).
- [2] I. Bloch, J. Dalibard, and W. Zwerger, Many-body physics with ultracold gases, Rev. Mod. Phys. 80, 885 (2008).
- [3] C. Gross and I. Bloch, Quantum simulations with ultracold atoms in optical lattices, Science 357, 995 (2017).
- [4] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, Rev. Mod. Phys. 86, 153 (2014).
- [5] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. 91, 045002 (2019).

- [6] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. 13, 435 (2017).
- [7] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. 13, 431 (2017).
- [8] Y.-H. Liu and E. P. L. van Nieuwenburg, Discriminative Cooperative Networks for Detecting Phase Transitions, Phys. Rev. Lett. 120, 176401 (2018).
- [9] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine Learning Phases of Strongly Correlated Fermions, Phys. Rev. X 7, 031038 (2017).
- [10] K. Mills and I. Tamblyn, Deep neural networks for direct, featureless learning through observation: The case of 2D spin models, Phys. Rev. E 97, 032119 (2018).

- [11] S. Pilati and P. Pieri, Supervised machine learning of ultracold atoms with speckle disorder, Sci. Rep. 9, 5613 (2019).
- [12] N. Saraceni, S. Cantori, and S. Pilati, Scalable neural networks for the efficient learning of disordered quantum systems, Phys. Rev. E 102, 033301 (2020).
- [13] Y. Zhang, A. Mesaros, K. Fujita, S. D. Edkins, M. H. Hamidian, K. Ch'ng, H. Eisaki, S. Uchida, J. C. S. Davis, E. Khatami, and E.-A. Kim, Machine learning in electronic-quantum-matter imaging experiments, Nature (London) 570, 484 (2019).
- [14] A. Bohrdt, C. S. Chiu, G. Ji, M. Xu, D. Greif, M. Greiner, E. Demler, F. Grusdt, and M. Knap, Classifying snapshots of the doped Hubbard model with machine learning, Nat. Phys. 15, 921 (2019).
- [15] E. Khatami, E. Guardado-Sanchez, B. M. Spar, J. F. Carrasquilla, W. S. Bakr, and R. T. Scalettar, Visualizing strange metallic correlations in the two-dimensional Fermi-Hubbard model with artificial intelligence, Phys. Rev. A 102, 033326 (2020).
- [16] B. S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, and C. Weitenberg, Identifying quantum phase transitions using artificial neural networks on experimental data, Nat. Phys. 15, 917 (2019).
- [17] F. Noé, S. Olsson, J. Köhler, and H. Wu, Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning, Science 365, eaaw1147 (2019).
- [18] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science 355, 602 (2017).
- [19] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nat. Phys. 14, 447 (2018).
- [20] G. Torlai and R. G. Melko, Machine-learning quantum states in the NISQ era, Annu. Rev. Condens. Matter Phys. 11, 325 (2020).
- [21] K. Choo, G. Carleo, N. Regnault, and T. Neupert, Symmetries and Many-Body Excitations with Neural-Network Quantum States, Phys. Rev. Lett. 121, 167204 (2018).
- [22] T. Vieijra, C. Casert, J. Nys, W. De Neve, J. Haegeman, J. Ryckebusch, and F. Verstraete, Restricted Boltzmann Machines for Quantum States with Non-Abelian or Anyonic Symmetries, Phys. Rev. Lett. **124**, 097201 (2020).
- [23] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, Restricted Boltzmann machines in quantum physics, Nat. Phys. 15, 887 (2019).
- [24] R. Zen, L. My, R. Tan, F. Hébert, M. Gattobigio, C. Miniatura, D. Poletti, and S. Bressan, Transfer learning for scalability of neural-network quantum states, Phys. Rev. E 101, 053301 (2020).
- [25] L. McInnes, J. Healy, N. Saul, and L. Großberger, UMAP: Uniform manifold approximation and projection, J. Open Source Softw. 3, 861 (2018).
- [26] C. S. Chiu, G. Ji, A. Bohrdt, M. Xu, M. Knap, E. Demler, F. Grusdt, M. Greiner, and D. Greif, String patterns in the doped Hubbard model, Science 365, 251 (2019).
- [27] P. A. Lee, N. Nagaosa, and X.-G. Wen, Doping a Mott insulator: Physics of high-temperature superconductivity, Rev. Mod. Phys. 78, 17 (2006).
- [28] B. Keimer, S. A. Kivelson, M. R. Norman, S. Uchida, and J. Zaanen, From quantum matter to high-temperature superconductivity in copper oxides, Nature (London) 518, 179 (2015).

- [29] T. Esslinger, Fermi-Hubbard physics with atoms in an optical lattice, Annu. Rev. Condens. Matter Phys. 1, 129 (2010).
- [30] G. Ness, A. Vainbaum, C. Shkedrov, Y. Florshaim, and Y. Sagi, Single-Exposure Absorption Imaging of Ultracold Atoms Using Deep Learning, Phys. Rev. Appl. 14, 014011 (2020).
- [31] C. S. Chiu, G. Ji, A. Bohrdt, M. Xu, M. Knap, E. Demler, F. Grusdt, M. Greiner, and D. Greif, Data for "String patterns in the doped Hubbard model," Harvard Dataverse, doi:10.7910/DVN/1CSVBV (2019).
- [32] A. Mazurenko, C. S. Chiu, G. Ji, M. F. Parsons, M. Kanász-Nagy, R. Schmidt, F. Grusdt, E. Demler, D. Greif, and M. Greiner, A cold-atom Fermi-Hubbard antiferromagnet, Nature (London) 545, 462 (2017).
- [33] F. Grusdt, M. Kénasz-Nagy, A. Bohrdt, C. S. Chiu, G. Ji, M. Greiner, D. Greif, and E. Demler, Parton Theory of Magnetic Polarons: Mesonic Resonances and Signatures in Dynamics, Phys. Rev. X 8, 011046 (2018).
- [34] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks, Advances in neural information processing systems 27 (2014).
- [35] M. Mirza and S. Osindero, Conditional generative adversarial nets, arXiv:1411.1784 (2014).
- [36] M. Arjovsky, S. Chintala, and L. Bottou, Wasserstein generative adversarial networks, *International Conference on Machine Learning* (PMLR, 2017), pp. 214–223.
- [37] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2017), pp. 4681–4690.
- [38] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, Learning to discover cross-domain relations with generative adversarial networks, *International Conference on Machine Learning* (PMLR, 2017), pp. 1857–1865.
- [39] Z. Liu, S. P. Rodrigues, and W. Cai, Simulating the Ising model with a deep convolutional generative adversarial network, arXiv:1710.04987 (2017).
- [40] K. Mills and I. Tamblyn, Phase space sampling and operator confidence with generative adversarial networks, arXiv:1710.08053 (2017).
- [41] J. M. Pawlowski and J. M. Urban, Reducing autocorrelation times in lattice simulations with generative adversarial networks, Mach. Learn.: Sci. Technol. 1, 045011 (2020).
- [42] K. Zhou, G. Endrödi, L.-G. Pang, and H. Stöcker, Regressive and generative neural networks for scalar field theory, Phys. Rev. D 100, 011501 (2019).
- [43] B. Sanchez-Lengeling and A. Aspuru-Guzik, Inverse molecular design using machine learning: Generative models for matter engineering, Science 361, 360 (2018).
- [44] L. de Oliveira, M. Paganini, and B. Nachman, Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis, Computing and Software for Big Science 1, 4 (2017).
- [45] A. B. Farimani, J. Gomes, and V. S. Pande, Deep learning the physics of transport phenomena, arXiv:1709.02432 (2017).
- [46] M. F. Parsons, A. Mazurenko, C. S. Chiu, G. Ji, D. Greif, and M. Greiner, Site-resolved measurement of the spin-correlation

function in the Fermi-Hubbard model, Science **353**, 1253 (2016).

- [47] T. A. Hilker, G. Salomon, F. Grusdt, A. Omran, M. Boll, E. Demler, I. Bloch, and C. Gross, Revealing hidden antiferromagnetic correlations in doped Hubbard chains via string correlators, Science 357, 484 (2017).
- [48] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, Improved training of Wasserstein GANs, arXiv:1704.00028 (2017).
- [49] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, Improving the improved training of Wasserstein GANs: A consistency term and its dual effect, arXiv:1803.01541 (2018).
- [50] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv:1207.0580 (2012).
- [51] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, Self-attention generative adversarial networks, *International*

Conference on Machine Learning (PMLR, 2019), pp. 7354–7363.

- [52] P. Ponte and R. G. Melko, Kernel methods for interpretable machine learning of order parameters, Phys. Rev. B 96, 205146 (2017).
- [53] S. J. Wetzel and M. Scherzer, Machine learning of explicit order parameters: From the Ising model to SU(2) lattice gauge theory, Phys. Rev. B 96, 184410 (2017).
- [54] C. Casert, T. Vieijra, J. Nys, and J. Ryckebusch, Interpretable machine learning for inferring the phase boundaries in a nonequilibrium system, Phys. Rev. E 99, 023304 (2019).
- [55] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, *Proceedings of the IEEE Conference* on Computer Vision and Pattern Recognition (IEEE, 2015), pp. 770–778.
- [56] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 (2014).
- [57] http://clean.energyscience.ca/codes.