

Conditional generative models for sampling and phase transition indication in spin systems

Japneet Singh ¹, Mathias S. Scheurer ^{2,3*}, Vipul Arora ¹

¹ Department Of Electrical Engineering, Indian Institute Of Technology Kanpur, Kanpur, Uttar Pradesh, India

² Institute for Theoretical Physics, University of Innsbruck, Innsbruck A-6020, Austria

³ Department of Physics, Harvard University, Cambridge MA 02138, USA

* Mathias.Scheurer@uibk.ac.at

July 28, 2021

1 Abstract

In this work, we study generative adversarial networks (GANs) as a tool to learn the distribution of spin configurations and to generate samples, conditioned on external tuning parameters or other quantities associated with individual configurations. For concreteness, we focus on two examples of conditional variables—the temperature of the system and the energy of the samples. We show that temperature-conditioned models can not only be used to generate samples across thermal phase transitions, but also be employed as unsupervised indicators of transitions. To this end, we introduce a GAN-fidelity measure that captures the model’s susceptibility to external changes of parameters. The proposed energy-conditioned models are integrated with Monte Carlo simulations to perform over-relaxation steps, which break the Markov chain and reduce auto-correlations. We propose ways of efficiently representing the physical states in our network architectures, e.g., by exploiting symmetries, and to minimize the correlations between generated samples. A detailed evaluation, using the two-dimensional XY model as an example, shows that these incorporations bring in considerable improvements over standard machine-learning approaches. We further study the performance of our architectures when no training data is provided near the critical region.

20

21 Contents

22	1 Introduction	2
23	2 Generative modelling and XY model	5
24	2.1 Variational autoencoders	5
25	2.2 Generative adversarial networks	6
26	2.3 2D XY model	7
27	3 Proposed method	8
28	3.1 Representation of physical states	8
29	3.1.1 Exploiting symmetries	8
30	3.1.2 Topology of degrees of freedom	9
31	3.1.3 Periodic boundary conditions	9

32	3.2	Proposed conditional models	9
33	3.2.1	Minimizing output biases	9
34	3.2.2	Maximizing the output entropy	10
35	3.3	Unsupervised detection of phase transitions	11
36	3.4	Over-relaxation and models conditioned on energy	12
37	3.4.1	General procedure	13
38	3.4.2	Solving the optimization problem	13
39	4	Numerical experiments	14
40	4.1	Generation of training data	14
41	4.2	Evaluation metrics	15
42	4.2.1	Percentage overlap (%OL)	15
43	4.2.2	Earth mover distance (EMD)	15
44	4.3	Baseline models for comparison	15
45	4.3.1	C-HG-VAE	15
46	4.3.2	C-GAN	16
47	4.4	Proposed method: ImplicitGAN	16
48	4.5	Results	17
49	4.5.1	Comparison with baselines: matching observables	17
50	4.5.2	Detecting phase transitions	18
51	4.5.3	Models conditioned on energy	19
52	4.5.4	Interpolating across unseen temperatures around T_c	20
53	5	Conclusions	21
54	A	Ablation analysis	22
55	B	Characteristics of ImplicitGAN-E	24
56		References	26

58

59 1 Introduction

60 Generative models [1–4] aim at modelling complicated probability distributions of data in
61 a way that they can readily be used to generate new samples. These techniques model the
62 joint distribution of data, such as images of handwritten digits, and some useful quantities
63 associated with the data, e.g., which of the ten digits is shown. The model is then used
64 to generate unseen data by sampling from the learnt joint probability distribution, e.g.,
65 produce unseen images of digits.

66 In physics, we often start from a Hamiltonian, an action, or just a classical configura-
67 tion energy, describing the system of interest, and, as such, formally, know the distribution
68 of the elementary degrees of freedom, such as the fields in a field theory or the spin con-
69 figurations in a classical spin model. Typically, one is interested in studying the behavior
70 of these distributions as a function of tuning parameters, e.g., temperature or coupling
71 constants, and one can think of them as the distribution of data conditioned on these
72 tuning parameters. Since, however, this data is usually very high-dimensional, the es-
73 sential physical properties can only be captured by evaluating physical quantities, such

74 as symmetry-breaking order parameters and their susceptibilities, or non-local probes of
75 topological properties. In most interesting cases, their evaluation cannot be performed
76 analytically and, hence, numerical techniques have to be used. Among those, in particu-
77 lar, Monte Carlo methods, where observables are estimated by sampling from the data,
78 are powerful, as they, at least in principle, guarantee asymptotic convergence to the true
79 distribution.

80 Markov chain Monte Carlo (MCMC) techniques work by constructing a first order
81 Markov sequence where the next sample is dependent on the current sample. Unfortu-
82 nately, these methods can suffer from the problem of large thermalization times and large
83 auto-correlation times (especially near phase transitions), both of which increase drasti-
84 cally with the increase in lattice size. For quickly generating uncorrelated samples, we
85 need the auto-correlation time to be small. Starting from a random configuration, for
86 efficiently reaching the state of generating valid samples that conform to the underlying
87 true distribution, the thermalization time has to be short as well.

88 To curtail the effect of dramatic increase of auto-correlation time near criticality, many
89 global update methods have been developed, which simultaneously change the variables
90 at many sites in a single MC update, such as Swendsen-Wang [5], Wolff [6], worm [7],
91 loop [8,9] and directed loop [10,11] algorithms. But these methods work only for specific
92 types of models and not for any generic system.

93 Besides several other promising applications of machine-learning methods in physics
94 [12–16], generative modelling techniques have been explored for enhanced generalizabil-
95 ity and performance. For instance, Efthymiou and Melko [17] use deep-learning-based
96 super-resolution techniques to produce spin configurations of larger system sizes from
97 MCMC-generated configurations of smaller sizes by the use of convolutional neural net-
98 works (CNNs). The resolved configurations have thermodynamic observables that agree
99 with Monte-Carlo calculations for one and two-dimensional (2D) Ising models. Another
100 approach is ‘self-learning Monte Carlo’ [18–21] that, in principle, works for any generic
101 system and applies machine-learning-based approaches on top of MCMC to speed up the
102 simulations and to reduce the increase in auto-correlation time near the critical temper-
103 ature. Other approaches which apply machine-learning techniques as a supplement or
104 alternative to MCMC are based on normalizing flow [22], Boltzmann machines [23–26], on
105 reinforcement learning [27], on generative adversarial networks (GANs) [28–33], autoen-
106 coders [34–36], and on variational autoregressive networks [37–40].

107 So far, in most of these approaches, the underlying generative model is trained sepa-
108 rately for different values of the tuning parameters of the system, such as different temper-
109 atures. But when configurations for multiple temperatures, including close to criticality,
110 need to be generated, either they require configurations for that corresponding tempera-
111 ture and training a model again and/or the Markov chain has to be re-started altogether.
112 For this reason, we here explore a different and less used [31–33] strategy, which consists
113 of learning the *conditional* probability distribution of physical samples, conditioned on a
114 (in general set of) parameter(s) c .

115 One can distinguish two different types of conditional parameters relevant for physical
116 models: c can either be an external tuning parameter, such as temperature for a thermal
117 phase transition or coupling constants in a model, or a quantity that is associated with and
118 a unique function of each sample, such as its energy or the number of topological defects
119 in it. In this work, we study an example of each of the two types of c : temperature-
120 conditioned and energy-conditioned models. In the former case, as the name suggests,
121 we provide temperature as conditional information in the training data set (obtained via
122 MCMC) for our deep-learning-based conditional generative models. Most notably, these
123 include conditional GANs [41], among other models employed as baselines. After training,

124 our models are used to generate samples at different temperatures, which are not necessar-
125 ily equal to the values of temperature in the training data set. For our energy-conditioned
126 models, we show how they can be integrated with MCMC and can be used for additional
127 over-relaxation steps which break the Markov chain and dampen auto-correlations. They
128 are well-suited for this purpose, as they can quickly sample configurations with energy
129 close to the energy of the current sample in the Markov chain while being locally dissimi-
130 lar. We also study the performance of these two different applications when the training
131 data is limited to temperatures away from the transition. Due to the generality of our
132 approach, we believe that the optimization strategies for generative modeling of physical
133 systems we discuss in this work will also be useful for the application to experimentally
134 generated data [33, 42].

135 Generative models can be broadly subsumed into two categories—prescribed and im-
136 plicit [43]. *Prescribed models* are those that provide an explicit parametric specification of
137 the distribution of the output (data). These models typically deploy Bernoulli or Gaus-
138 sian outputs, depending on the type of data. On the other hand, *implicit models* directly
139 generate data by passing a noise vector through a deterministic function which is generally
140 a neural network. Implicit models can be more expressive than their prescribed counter-
141 parts but calculating likelihood becomes intractable in most cases. Most of the generative
142 models in machine learning are prescribed models as they have a notion of likelihood, are
143 easy to optimize and produce excellent results. But, generally, they make an assumption
144 of independence between the parametric distribution across various pixels or lattice sites.
145 Such assumptions in physics can be quite restrictive as the models need to capture the cor-
146 relations between lattice sites. Prescribed models would otherwise need to estimate large
147 co-variance matrices and ensure their positive-definiteness. For this reason, we expect and
148 also confirm by our numerical experiments that implicit generative models, in particular
149 in the GAN framework, are more suitable for modelling the site-to-site correlations in
150 physical systems.

151 Additionally, we propose other modifications that exploit the underlying structure
152 of the physical systems and enhance the model’s utility. The proposed modifications
153 can bring significant improvement in performance as compared to the prescribed models
154 treated as baselines. We also show that, for implicit models, maximizing the mutual
155 information between a set of structured latent variables and reconstructed configurations
156 leads to maximizing a lower bound on the entropy of the learnt distribution; this reduces
157 the correlations among configurations generated by the model and can act as an indicator
158 of phase transitions. We evaluate in detail the improvements in performance of the various
159 modifications we propose. While our approaches can be readily applied to other systems as
160 well, we focus for concreteness in our numerical studies on the 2D XY model, as it provides
161 a transparent example to benchmark these modifications and has been established as a
162 challenging model for neural networks [44].

163 If the type of phase transition and the associated observable, e.g., a local order param-
164 eter, are known, these quantities can be evaluated with the generated samples to capture
165 the phase transition. For instance, in case of the XY model, the finite-temperature BKT
166 transition is associated with the proliferation/suppression of vortices [45–48]. While we
167 show that our generative models can indeed reproduce the expected behavior of vortices,
168 we also demonstrate that our trained network can be used to reveal the transition without
169 requiring knowledge about the underlying nature of the phase transition. This unsuper-
170 vised detection of phase transitions is another central topic of machine learning in physics.
171 In particular, topological transitions, such as the BKT transition, are challenging due to
172 their non-local nature; however, the method proposed in [49] has been demonstrated to
173 work in a variety of different models [49–51] and extensions [52] for symmetry-protected

174 topological phases have been developed. We here demonstrate that trained generative
 175 models can also be used to indicate the phase transition in an unsupervised way: as ex-
 176 pected [53–56], we find that the model is particularly susceptible to parameter changes
 177 in the vicinity of the transition. We quantify this by introducing a fidelity measure con-
 178 structed on the trained GAN that can be efficiently evaluated and shows peaks in the
 179 vicinity of the phase transition.

180 The remainder of this paper is organized as follows. In Sec. 2, we provide an introduc-
 181 tion to the different generative modelling techniques we explore in this work and to the XY
 182 model. The modifications we propose for an effective modelling of physical systems are
 183 described in detail in Sec. 3. The numerical experiments, using the XY model as concrete
 184 example, are presented in Sec. 4. Finally, Sec. 5 contains a brief summary.

185 2 Generative modelling and XY model

186 To establish notation and nomenclature, we first provide an introduction to the generative
 187 machine-learning methods we use—variational autoencoders (VAEs) and GANs, as well as
 188 their conditional extensions; we also define the 2D XY model, which is the model we use
 189 to benchmark our machine learning approach with, and the physical quantities we study.
 190 Readers familiar with the XY model and these generative machine-learning techniques,
 191 can skip this section and proceed directly with Sec. 3.

192 2.1 Variational autoencoders

193 VAEs are powerful continuous latent variable models used for generative modelling of a
 194 high-dimensional distribution over a given data set, allowing one to sample directly from
 195 the data distribution [57]. They have shown promising results in producing unseen fake
 196 images and audio files which are almost indistinguishable from real data, see Ref. [58] for
 197 instance. In its standard form, a VAE consists of an encoder and a decoder. The encoder
 198 maps from data space \mathbf{X} to a latent space $\mathbf{z} \subseteq \mathbb{R}^D$ and consists of a family of distributions
 199 \mathbb{Q}_ϕ on \mathbf{z} parameterized by ϕ ; it is typically modeled by deep neural networks. The decoder
 200 consists of a family of distributions \mathbb{P}_θ on \mathbf{X} parameterized by θ . As the name implies,
 201 the encoder encodes the semantic information present in the data into the latent space.
 202 The decoder uses the encoded information in latent space to reconstruct the data. The
 203 overall objective is to maximize the likelihood of the data, independently and identically
 204 distributed as $P(x) = \int P_\theta(x|z)P(z)dz$, where, $x \in \mathbf{X}$, $z \in \mathbf{z}$, $P_\theta(x|z) \in \mathbb{P}_\theta$, and $P(z)$ is
 205 the prior distribution, often taken as Gaussian. The likelihood is generally intractable to
 206 compute but can be maximized by maximizing the evidence lower bound (ELBO). The
 207 ELBO for marginal log-likelihood $P_\theta(x)$ for a data-point x is expressed as

$$\log P_\theta(x) \geq \mathbb{E}_{z \sim Q_\phi(z|x)}[\log P_\theta(x|z)] - D_{\text{KL}}[Q_\phi(z|x)||P(z)],$$

208 where $Q_\phi(z|x) \in \mathbb{Q}_\phi$. The ELBO consists of 2 terms: (i) a loss term accounting for the
 209 error in the reconstructed data and (ii) a regularizing term which makes the encoder to
 210 encode information such that its distribution is close in Kullback-Leibler (KL) divergence,
 211 D_{KL} , to the prior distribution $P(z)$.

212 **Conditional VAE (C-VAE)** is a simple extension of standard VAE, with the only
 213 difference that the data distribution as well as the latent distribution are both condi-
 214 tioned by some external information. We illustrate the typical structure of a C-VAE in
 215 Fig. 1a. The objective is now to maximize the likelihood conditioned on a given condi-
 216 tional information c . For our purposes here of generating samples of a physical model,

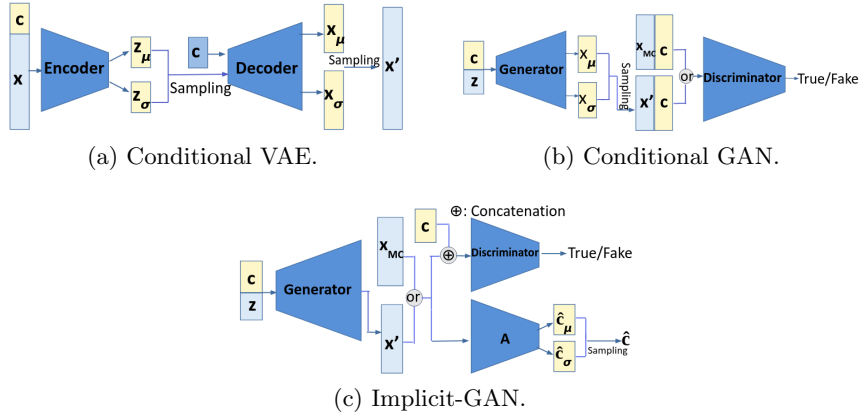


Figure 1: Block-diagram representation of (a) C-VAE, (b) C-GAN, and (c), our proposed method, an Implicit-GAN. We refer to the respective parts of the main text, Sec. 2.1, Sec. 2.2, and Sec. 3.2, for a detailed description.

217 the “conditional information” refers to the tuning parameters of interest in that model,
 218 such as temperature, T , ratios of exchange interactions in spin models, and the energy of
 219 samples, which can be used for sampling of the corresponding microcanonical ensemble or,
 220 as we will demonstrate below, decorrelate regular MCMC schemes by providing efficient
 221 overrelaxation steps. In general, c can be a multi-component vector comprising several
 222 physical tuning parameters or quantities associated with the individual samples.

223 To train the C-VAE, we again maximize the ELBO, now assuming the form

$$\log P_\theta(x|c) \geq \mathbb{E}_{z \sim Q_\phi(z|x,c)}[\log P_\theta(x|z,c)] - D_{\text{KL}}[Q_\phi(z|x,c) \| P(z|c)].$$

224 Here, we will assume the prior distribution to be independent of c and to follow a normal
 225 distribution with zero mean and variance 1, i.e., $P(z|c) = P(z) = \mathcal{N}(0, I)$.

226 2.2 Generative adversarial networks

227 GANs [59] are another powerful framework for modelling a probability distribution. In
 228 physics, GANs have been successfully applied to many different models ranging from
 229 binary spin systems like the Ising model [29], to the Fermi-Hubbard model [33], high-energy
 230 physics [28], cosmology [60], and material science [30]. A GAN consists of two models, a
 231 generator $G(z)$ and a discriminator $D(x)$. The generator is a function $G: z \rightarrow \mathbf{X}$ which
 232 tries to capture the data distribution and produces samples x that closely resemble samples
 233 from the training data. On the other hand, the discriminator is a function $D: \mathbf{X} \rightarrow (0, 1)$
 234 which tries to estimate the probability that a sample came from the true data distribution
 235 (true sample) rather than from the generative model G (fake/negative sample). G tries to
 236 maximize the probability of D making a mistake while D tries to minimize the probability
 237 of being fooled by G . The result is a minimax game between two players, described by
 238 the value function

$$V(G, D) = \mathbb{E}_{x \sim p_{\text{Data}}}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]. \quad (1)$$

239 The objective of this game can be expressed as $\min_G \max_D V(G, D)$.

240 **Conditional GANs (C-GANs)** are a simple extension [41] of standard GANs in
 241 which the generator produces samples based on the external information c while the dis-
 242 criminator tries to estimate the probability that the sample came from the true conditional

243 data distribution rather than from G . The associated minimax objective now becomes

$$\min_G \max_D V(G, D; c) = \min_G \max_D \left(\mathbb{E}_{x \sim p_{\text{Data}}} [\log D(x; c)] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z; c); c))] \right) \quad (2)$$

244 and we show the basic structure of a C-GAN in Fig. 1b.

245 2.3 2D XY model

246 While the methods we propose and compare in this work are more generally applicable,
 247 we will employ one specific physical model, the classical 2D XY-spin model, to illustrate
 248 and test the generative machine-learning methods. The XY model was chosen as it fea-
 249 tures key challenges—compact local degrees of freedom (two-component unit vectors) and
 250 non-local, topological excitations (vortices) together with conventional excitations (spin
 251 waves)—in a minimal setting. At the same time, it is accessible via conventional MCMC
 252 sampling schemes, which is important for us since it allows to test the accuracy of our
 253 generative models.

254 More specifically, the XY model consists of two-component spins on every site i of
 255 the lattice with fixed magnitude, which we set to 1 and, hence, are described by the unit
 256 vectors $\mathbf{s}_i = (\cos \theta_i, \sin \theta_i)^T$, $\theta_i \in [0, 2\pi)$. We here consider a 2D square-lattice of size
 257 $N \times N$ and restrict ourselves to ferromagnetic nearest-neighbor interactions, $J > 0$; using
 258 the latter as unit of energy, $J \equiv 1$, the energy of a configuration $\boldsymbol{\theta} = \{\theta_i\}$ is given by

$$E(\boldsymbol{\theta}) = - \sum_{\langle i, j \rangle} \mathbf{s}_i \cdot \mathbf{s}_j = - \sum_{\langle i, j \rangle} \cos(\theta_i - \theta_j), \quad (3)$$

259 where the sum over $\langle i, j \rangle$ includes all the adjacent sites on the lattice.

260 The probability density of a configuration $\boldsymbol{\theta}$ at a given temperature $T \in \mathbb{R}^+$ is given by

$$P_T(\boldsymbol{\theta}) = \frac{1}{Z(T)} e^{-\frac{E(\boldsymbol{\theta})}{T}}, \quad (4)$$

262 where the Boltzmann constant is set to unity and $Z(T) = \sum_{\boldsymbol{\theta}} e^{-\frac{E(\boldsymbol{\theta})}{T}}$ is the partition
 263 function. Thermal expectation values, $\langle \mathcal{O} \rangle_T$, of physical quantities $\mathcal{O} = \mathcal{O}(\boldsymbol{\theta})$, such as
 264 mean magnetization, $\mathbf{m}(\boldsymbol{\theta}) = N^{-2} \sum_i \mathbf{s}_i(\theta_i)$, or mean energy, $e(\boldsymbol{\theta}) = N^{-2} E(\boldsymbol{\theta})$, follow from
 265 Eq. (4) as

$$\langle \mathcal{O} \rangle_T = \sum_{\boldsymbol{\theta}} \mathcal{O}(\boldsymbol{\theta}) P_T(\boldsymbol{\theta}). \quad (5)$$

266 In general, Eq. (5) cannot be evaluated exactly and, hence, has to be analyzed with ap-
 267 proximate analytical techniques or numerical approaches. One of the most common ways
 268 of evaluating the sum in Eq. (5) numerically, proceed via MCMC sampling of configu-
 269 rations $\boldsymbol{\theta}$ according to the distribution $P_T(\boldsymbol{\theta})$, e.g., via the Metropolis-Hastings (MH)
 270 algorithm [61]. In each step of the MH algorithm, a configuration $\boldsymbol{\theta}'$ is generated from
 271 a current configuration $\boldsymbol{\theta}$ with some *a priori* selection probability $W(\boldsymbol{\theta}'|\boldsymbol{\theta})$. This new
 272 configuration is then accepted with probability

$$W_A(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \min \left(1, \frac{W(\boldsymbol{\theta}|\boldsymbol{\theta}') e^{-E(\boldsymbol{\theta}')}}{W(\boldsymbol{\theta}'|\boldsymbol{\theta}) e^{-E(\boldsymbol{\theta})}} \right). \quad (6)$$

273 When $W(\cdot)$ is symmetric, i.e., $W(\boldsymbol{\theta}|\boldsymbol{\theta}') = W(\boldsymbol{\theta}'|\boldsymbol{\theta})$, then Eq. (6) becomes

$$W_A(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \min(1, e^{-(E(\boldsymbol{\theta}') - E(\boldsymbol{\theta}))}). \quad (7)$$

274 The goal of this work is to investigate how generative models can be used to generate
 275 samples θ for efficient evaluation of the expectation values of observables in Eq. (5).

276 Besides the mean energy and magnetization mentioned above, we also investigate the
 277 number of vortices in the system at a given temperature. Vortices are non-local excitations
 278 defined by a non-zero winding, $\nu \neq 0$, of the unit vector \mathbf{s}_i on any closed path encircling
 279 the core of the vortex. Proliferation or suppression of vortices are the defining feature for
 280 the finite-temperature phase transition, the BKT transition [45–48], of the 2D XY model.
 281 Studying vortices is not only motivated by the fact that they are integral to the physics of
 282 the XY model, but also due to their non-local, topological nature; as a consequence, one
 283 might expect that vortices are more difficult to capture by machine-learning techniques
 284 than local excitations.

285 In practice, we detect vortices in samples by counting, for every site i , the angle
 286 differences in anti-clockwise sense around the (3×3) square centered at i . Each difference
 287 was constrained to lie in $(-\pi, \pi]$ using a saw function. The “vorticity” V of a configuration
 288 θ is the number of vortices with winding number $\nu = +1$.

289 3 Proposed method

290 Having introduced the basic generative models, we will next discuss our proposed imple-
 291 mentation and some additional modifications which improve the models’ performance in
 292 generating samples. To be concrete, we will discuss them mostly in the context of the
 293 2D XY model, although they apply equally well to many other systems as well. These
 294 modifications are motivated from the structure of the physical system. First, we discuss
 295 how the states are represented in both of our implementations. Then we detail the changes
 296 in the models’ structures and training objectives. To analyze systematically how relevant
 297 the different modifications are, we present an ablation analysis in Appendix A.

298 3.1 Representation of physical states

299 The first set of modifications concerns the representation of states. As we will see, choosing
 300 a proper way of parameterizing the physical states is integral to an efficient and feasible
 301 generative modelling.

302 3.1.1 Exploiting symmetries

303 First of all, many physical systems exhibit symmetries. Formally, this means that the
 304 energy $E(x)$ of any state x is the same as that of the transformed state, x' , $E(x) = E(x')$.
 305 This can be exploited to find a more compact representation of the state: one can represent
 306 states such that two states that are related by a symmetry have the exact same representa-
 307 tion. Unbiased sampling is guaranteed by randomly performing symmetry transformations
 308 on the generated state, since $E(x) = E(x')$ implies that any two symmetry-related states
 309 are equally likely.

310 In the case of the XY model, an important symmetry is the invariance under global
 311 rotation of all spins,

$$\theta \longrightarrow \theta' = \theta + \theta_0, \quad \theta_0 \in \mathbb{R}. \quad (8a)$$

312 This symmetry allows us to reduce the dimensionality of the representation of the states
 313 from N^2 to $N^2 - 1$. In practice, for any given state θ we choose θ_0 such that

$$(\mathbf{m}(\theta'_i))_y = N^{-2} \sum_i \sin(\theta'_i) = 0, \quad (8b)$$

314 i.e., describe the state by deviations of the spin orientations about a certain ‘mean-
 315 direction’ (here chosen along the x -axis). As $E(\boldsymbol{\theta})$ for the XY model is invariant un-
 316 der Eq. (8a), we know $P_T(\boldsymbol{\theta}) = P_T(\boldsymbol{\theta}', \theta_0) = P(\boldsymbol{\theta}')P(\theta_0)$, with uniform $P(\theta_0)$. We will
 317 model $P(\boldsymbol{\theta}')$ using a deep generative model, and sample θ_0 uniformly in $[0, 2\pi)$. Thus,
 318 we have reduced the dimensionality of space (the degrees of freedom of data) in which the
 319 manifold of lattice configurations is embedded and, more importantly, made sure that the
 320 symmetries are respected *exactly* by our sampling procedure.

321 3.1.2 Topology of degrees of freedom

322 For many physical systems, the degrees of freedom on every site are compact. For instance,
 323 for XY-spin or Heisenberg-spin models, the local configuration space is a one-dimensional
 324 or two-dimensional sphere, respectively. In these cases, one has to be careful about choos-
 325 ing a smooth representation of these spaces that respects their topology.

326 For the XY, the angles $\theta_i \in [0, 2\pi)$ have discontinuous jumps at 2π . As such, directly
 327 using angles as input to the model does not explicitly take into account the topological
 328 and geometrical properties of the space of XY spins. For example, an angle of 2° is quite
 329 similar to 358° , and also 180° is not a good estimate of the mean spin orientation. The
 330 topology at each lattice site can be taken into account by using a two-channel lattice
 331 consisting of cosines and sines of lattice angles at both input and output of our model;
 332 this means that instead of θ_i , we use the two-component unit vectors $\mathbf{s}_i = (\cos \theta_i, \sin \theta_i)^T$,
 333 as has previously been implemented for different machine learning studies of the XY model
 334 (see, e.g., Ref. [62]).

335 Such a choice of input and output makes the model an implicit model. This also
 336 allows to overcome the limitations on the model’s ability to capture correlations between
 337 lattice sites due to independent sampling from $N(\mu_i, \sigma_i)$ at each lattice site i . We use this
 338 representation for the GAN framework. A similar extension to VAE framework makes the
 339 ELBO intractable. While there exist approaches like that of Ref. [63] to overcome this
 340 issue, most of them are based on adversarial training (or likelihood free inference).

341 3.1.3 Periodic boundary conditions

342 As we are interested in the bulk properties of the XY model and not in the behavior around
 343 edges, we will assume periodic boundary conditions throughout this work. Mathematically,
 344 this means that we replace $\theta_{(i_1, i_2)}$, by $\tilde{\theta}_{(i_1, i_2)} := \theta_{((i_1)_N, (i_2)_N)}$, where $(i)_N$ denotes i modulo
 345 N . For the implementation with deep neural networks, we increase the size of the lattice
 346 from $N \times N$ to $(N + 2) \times (N + 2)$, keeping the middle $N \times N$ lattice sites the same and
 347 filling the sites at the new edges in accordance with the periodic boundary conditions.
 348 We expect that this improves the performance of feature extracting kernels of the CNNs
 349 especially at the “edges” of a lattice. We use this form of periodic padding on the input
 350 layer of the encoder (for VAE) or discriminator (for GAN).

351 3.2 Proposed conditional models

352 Now, we describe the proposed implicit GAN models for lattice simulations. The Implicit-
 353 GAN can be conditioned on temperature or on energy, which we denote as “ImplicitGAN-
 354 T ” and “ImplicitGAN- E ”, respectively.

355 3.2.1 Minimizing output biases

356 As mentioned above, we propose to normalize the spin configurations such that their net
 357 magnetization vector $\mathbf{m}(\boldsymbol{\theta}')$ always points along the x -axis, see Eq. (8). But, there is

358 nothing in the training objective in Eq. (2) which explicitly incentivizes the network to
 359 produce configurations with their magnetization to point along the x-axis. If this condition
 360 is not satisfied, it implies that our model has developed some bias, which may be due to the
 361 model parameters being stuck in a local minimum during training. We indeed observed
 362 that the training objective in Eq. (2) can lead to bad local optima, as discussed later in
 363 Sec. A. Thus, if we add a term forcing the generative model to minimize the square of the
 364 y -component of the magnetization in a configuration we can minimize such biases. The
 365 GAN value function now becomes

$$V_b(G, D; c) = V(G, D; c) + \lambda \mathbb{E}_{z \sim p(z); \theta' = G(z; c)} \left[\sum_i \frac{\sin(\theta'_i)}{N^2} \right]^2 \quad (9)$$

366 where, $\lambda \in \mathbb{R}^+$ is a constant hyper-parameter.

367 3.2.2 Maximizing the output entropy

368 The generated samples will hardly have any practical significance if we cannot guarantee
 369 convergence to the exact distribution—especially considering the fact that GANs are sus-
 370 ceptible to the mode-collapse problem, i.e., they might miss a subset of the modes of a
 371 multimodal distribution of the samples. In practice, we could use the generated x as the
 372 initial configuration for MCMC. But if the different samples generated by our model have
 373 high correlations among themselves, the number of MCMC steps needed to obtain uncor-
 374 related samples would be large, thereby, defeating the purpose of the extra computational
 375 efforts for training the generator. We can decrease the number of MCMC steps needed if
 376 we can reduce the initial correlation among the different samples generated by our model.

377 To achieve this, we propose to additionally maximize the overall entropy (more specifi-
 378 cally, the ‘differential entropy’) of the learnt distribution $h(G(z, c))$, i.e., to make the
 379 learnt distribution more ‘diffused’, while also keeping the distribution of generated sam-
 380 ples in close agreement to the true distribution for all temperatures. It has been shown
 381 that, in the case of prescribed models, the entropy-regularized loss function reduces the
 382 problem of mode-collapse [64]. In practice, the problem is that $h(x)$ is difficult to compute
 383 or maximize. However, we can instead maximize a lower bound on $h(x)$ in the following
 384 way: due to the symmetry, $I(x; c) = I(c; x)$, of the mutual information I , it holds

$$\begin{aligned} h(x) &= h(c) - h(c|x) + h(x|c) \\ &\geq h(c) - h(c|x) + h(x|c, z). \end{aligned}$$

385 Now, $h(x|c, z) = 0$ for an implicit model (as opposed to prescribed models, where $h(x|c, z)$
 386 may not be non-negative), because the value of x is completely determined by the value
 387 of $\{c, z\}$. Thus,

$$h(x) \geq h(c) - h(c|x) = I(c; x). \quad (10)$$

388 Here $h(c)$ is constant because we have already specified and fixed the latent distribution of
 389 conditional information: in the case of temperature-conditioned models, $c \equiv T$ and $P(T)$
 390 is uniform over all temperatures in the training data. For energy-conditioned models, we
 391 have $c \equiv E$ with $P(E)$ being determined by the physical system and the choice of training
 392 data. Consequently, minimizing $h(c|x)$ maximizes the lower bound on $h(x)$.

393 Minimizing $h(c|x)$ requires access to the posterior $P(c|x)$. But, we can minimize an

394 upper bound on $h(c|x)$ by defining an auxiliary distribution $A(\hat{c}|x)$ as:

$$\begin{aligned}
h(c|x) &= -\mathbb{E}_x[\mathbb{E}_{c \sim P(c|x)}[\log P(c|x)]] \\
&= -\mathbb{E}_x[D_{\text{KL}}(P(\hat{c}|x)||A(\hat{c}|x)) + \mathbb{E}_{\hat{c} \sim P(c|x)}[\log A(\hat{c}|x)]] \\
&\leq -\mathbb{E}_x[\mathbb{E}_{\hat{c} \sim P(c|x)}[\log A(\hat{c}|x)]] \\
&= -\mathbb{E}_{\hat{c} \sim P(c)}[\mathbb{E}_{x \sim P(x|c)}[\log A(\hat{c}|x)]] \\
&\equiv L_H(G, A)
\end{aligned}
\tag{11}$$

395 We use an auxiliary network A to estimate c from x , i.e., maximize the probability $P(\hat{c} = c)$.
396 Such a technique of maximizing a lower bound on mutual information in terms of an
397 auxiliary distribution was previously proposed in [65]. According to Eq. (11), $h(\hat{c}|x)$
398 can be minimized by minimizing its upper bound given by $L_H(G, A)$. Note the bound
399 becomes tight when $\mathbb{E}_x[D_{\text{KL}}(P(\hat{c}|x)||A(\hat{c}|x))] \rightarrow 0$. The modified objective, which involves
400 the auxiliary distribution, is given by

$$\min_{G, A} \max_D \{V_b(G, D; c) + \gamma L_H(G, A)\},
\tag{12}$$

401 where $\gamma \in \mathbb{R}^+$ is a constant hyper-parameter and V_b as in Eq. (9). Note $L_H(G, A)$ maximizes
402 only a lower bound on the entropy and, hence, $h(x)$ is not guaranteed to increase. The gap
403 $h(x|c) - h(x|c, z) = I(x; z|c)$ is expected to be small since, by the structure of the model, one
404 does not expect large mutual information between noise variables and generated samples.
405 Since $I(x; z|c) \geq 0$, the overall entropy is likely to increase in practice.

406 Typically, A and D are implemented as neural networks sharing most of the layers.
407 But, in our case, the information of c should only be given to D and not to A . Therefore,
408 they were employed as separate neural networks, as shown in Fig. 1c. The discriminator
409 D tries to predict the probability that the sample belongs to the true distribution, while
410 the auxiliary network A outputs a distribution over c for a given configuration. The
411 distribution is assumed to be Gaussian with mean and variance \hat{c}_μ and \hat{c}_σ predicted by
412 the network A .

413 3.3 Unsupervised detection of phase transitions

414 So far, our focus has been on generating samples following Eq. (4) for the evaluation of
415 physical observables according to Eq. (5). If we are interested in studying phase tran-
416 sitions and know which observables capture the transition, e.g., a local order parameter
417 in case of a conventional, symmetry-breaking phase transition, we can simply evaluate
418 these observables with our generated samples. However, one of the central questions of
419 machine learning in the context of condensed matter and statistical physics is to find ways
420 of detecting the transition without “telling” the algorithm which observables are relevant.
421 The in this sense “unsupervised” detection of phase transitions could potentially be useful
422 in cases where the order parameter or topological invariant characterizing the transition
423 are not known.

424 Having constructed models that can generate samples at a given value of the conditional
425 parameter(s) c , we here analyze whether the behavior of these models upon tuning c can
426 be used to infer where phase transitions take place, without requiring any knowledge
427 about the underlying order parameter. In line with previous works [53–56], dealing with
428 different machine-learning setups, we expect that our generative models are particularly
429 susceptible to changes in c in the vicinity of phase transitions. For ease of reading and
430 since we explicit study this choice in our numerical experiments, we will use $c = T$ in the
431 remainder of this subsection. We reiterate, however, that our machine-learning framework

432 is able to provide samples subject to, in principle, arbitrary conditional constraints c . For
 433 instance, $c = E$ will allow studying transitions as a function of energy in a microcanonical
 434 ensemble or studying the behavior of the system as a function of other “post-selection”
 435 conditions on the samples is achievable as well.

436 The first measure we use is directly related to the one defined in previous works [53, 56]
 437 and makes use of the auxiliary network $A(x) = \hat{T}$ that we implemented to estimate the
 438 temperature from the samples x , needed to maximize the output entropy. One expects
 439 that the expectation value $\mathbb{E}_{x \sim P_T}[A(x)]$ over samples x at temperature T is approximately
 440 constant deep inside the two phases and that it varies maximally at the transition. As
 441 such

$$\mathcal{D}(T_0) = \left. \frac{\partial \mathbb{E}_{x \sim P_T}[A(x)]}{\partial T} \right|_{T=T_0} \approx \frac{\mathbb{E}_{x \sim P_{T_0+\Delta T}}[A(x)] - \mathbb{E}_{x \sim P_{T_0-\Delta T}}[A(x)]}{2\Delta T} \quad (13)$$

442 should be peaked around the critical temperature.

443 The second measure we introduce is unique to GANs and can be defined for any GAN
 444 architecture, not only for the modified version with the additional auxiliary network. This
 445 measure is analogous to the widely studied quantum fidelity, which has also been extended
 446 to finite temperature and thermal phase transitions [66]. It is based on the idea that the
 447 form of a state (density matrix for thermal ensembles) will change most dramatically upon
 448 modifying a tuning parameter by a small amount (such as temperature $T \rightarrow T + \Delta T$) in the
 449 vicinity of a phase transition. This will first require a measure of similarity of two states
 450 or ensembles. For this we will use the expectation value of $D(x, T)$ with x taken from
 451 some given ensemble p' . Since $D(x, T)$ estimates the probability of x coming from the
 452 true thermal ensemble, this expectation value quantifies how similar the thermal ensemble
 453 and p' are. Since we are interested in tuning temperature, we replace p' by the ensemble
 454 generated by the generator at a different temperature and, thus, define the *GAN fidelity*
 455 as

$$\mathcal{F}_{\text{GAN}}(T) = \frac{1}{\Delta T} \mathbb{E}_{z \sim p(z)} [D(G(z; T), T) - D(G(z; T), T + \Delta T)]. \quad (14)$$

456 Imagine starting in the high-temperature phase and gradually decreasing T . Once T
 457 reaches the phase transition, the generator in the second term in Eq. (14) starts producing
 458 samples that are not “expected” by the discriminator. Thus, the latter decreases its value,
 459 $\mathcal{F}_{\text{GAN}}(T)$ increases, and is expected to peak in the vicinity of the phase transition. We
 460 emphasize that the GAN fidelity in Eq. (14) is defined entirely in terms of the networks
 461 and can be evaluated very efficiently, once the networks have been trained.

462 3.4 Over-relaxation and models conditioned on energy

463 Similar to their temperature-conditioned counterparts, models conditioned on energy can
 464 also be used to provide samples directly and to study phase transitions. However, we
 465 here focus on a different application and discuss how energy-conditioned models can be
 466 integrated with MCMC to accelerate lattice simulations. Inspired by Ref. [28], where
 467 the potential of non-conditional GANs was explored as over-relaxation steps in MCMC
 468 simulations, we here propose to use *conditional* GANs for this purpose. By construction,
 469 our energy-conditioned models can provide samples with energy close to that of the current
 470 sample in the Monte-Carlo chain. As opposed to using unconditional GANs, no in general
 471 numerically expensive pre-sampling of the model is required to obtain samples within the
 472 desired energy range.

473 More specifically, the model we use here has the ImplicitGAN architecture introduced
 474 above. As opposed to the discussion in Sec. 3.3, where we focused on temperature-
 475 conditioned models, we here use the energy per site $e(\theta)$ of each sample θ rather than

476 temperature as conditional input and focus on $G(z, e)$ instead of the generalized form
 477 $G(z, c)$.

478 3.4.1 General procedure

479 Once the models are trained we generate samples in the following way:

- 480 1. Starting from an initial configuration θ_0 ,
- 481 2. perform n_{MC} MCMC updates to obtain a configuration θ_t .
- 482 3. To implement an over-relaxation step, we use the trained model and construct a new
 483 configuration, θ'_t , according to $\theta'_t = G(z_t, e_t^*)$, where e_t^* is obtained by fine-tuning the
 484 energy of the sample to the desired value,

$$e_t^* = \arg \min_e [E(G(z_t, e)) - E(\theta_t)]^2, \quad (15)$$

485 with z_t being sampled from the prior distribution $P(z)$.

- 486 4. Move to step 2 until enough samples are retained.

487 Note that, ideally, $e = E(\theta_t)/N^2$ would minimize Eq. (15), but this is not the case
 488 since GANs only approximately learn the distribution (see Appendix B for a discussion).
 489 Nonetheless, the energy of the samples produced by $G(z, E/N^2)$ are close to E and the
 490 true optimum of Eq. (15) is expected to be in the vicinity of $E(\theta_t)/N^2$. This makes finding
 491 e_t^* more efficient in our energy-conditioned model.

492 While it was argued in Ref. [28] that the selection probability W [entering Eq. (6)] of the
 493 GAN-based over-relaxation step is expected to be (approximately) symmetric, $W(\theta|\theta') =$
 494 $W(\theta'|\theta)$, we emphasize that this will strictly speaking not hold in general nor exactly. For
 495 instance, GANs suffering from the mode-collapse problem will fail to lead to a symmetric
 496 W . Nonetheless, we here *assume* that it holds for our trained models, which allows
 497 simplifying Eq. (6) to Eq. (7) and test, in Sec. 4.5.3, whether the samples generated from
 498 it have statistical properties close to the ground truth. The validity of this assumption is
 499 supported empirically by the good performance of the models.

500 3.4.2 Solving the optimization problem

501 One way to solve Eq. (15) is to back-propagate the gradients through the entire generator,
 502 keeping its weights fixed, which will be very expensive as it requires multiple forward and
 503 backward passes over a deep neural network and the number of iterations may be very
 504 large. Another practical problem with this approach is that in our architecture multiple
 505 copies of conditional information are set as input to the generator. If gradient descent
 506 is used, it is possible that it may decrease some of the values and may cause others to
 507 increase. If only a single copy of conditional information is used during training, the
 508 GAN may completely ignore this conditional information among relatively larger number
 509 of noise variables.

510 A simpler way is to solve it as a bandit optimization problem, where the only feedback
 511 one gets is the function value $f(e) = E(G(z, e))$ and not the gradient. When the model
 512 is only conditioned on energy, the bandit version of the problem is only one dimensional.
 513 Most well-known methods existing in the literature solve this problem by constructing an
 514 unbiased estimate of the gradient of ‘close approximation’ of f and then performing the
 515 updates from $e \rightarrow e + \Delta e$ according to gradient descent, i.e.

$$\Delta e = -\alpha(f(e_t) - E(\theta_t))f'(e_t), \quad (16)$$

516 where α is the step size. There are several methods to obtain an estimate of the gradient
 517 for a function $f(x)$. Here we use a two-point feedback estimate [67],

$$f'(x) \approx \frac{\mathbb{E}_u[(f(x + \delta u) - f(x))u]}{\delta}. \quad (17)$$

518 In Eq. (17), $u \sim \mathcal{N}(0, I)$ and δ should be kept sufficiently small to obtain high accuracy,
 519 while not too small to avoid increasing the variance of the gradient estimate. Instead of
 520 computing the exact expectation value, we use a stochastic estimate with only a single
 521 realization of u . In this way, $E(G(z, e))$ can be made arbitrarily close to $E(\boldsymbol{\theta})$. In practice,
 522 we set a threshold value ΔE_{thr} and the optimization will be done until a configuration
 523 with $\Delta E = |E(G(z_t, e)) - E(\boldsymbol{\theta})| \leq |\Delta E_{\text{thr}}|$ is found.

524 When considered over multiple over-relaxation steps, the problem in Eq. (15) can also
 525 be interpreted as an online optimization problem where at time step t an agent receives a
 526 loss function $f_t(e) = (E(G(e, z)) - E(\boldsymbol{\theta}_t))^2$ and the goal is to minimize the loss accumulated
 527 over various time steps. In our implementation, we exploit this nature and use the optimum
 528 of $f_t(\cdot)$ as starting value for our iterative minimization of $f_{t+1}(\cdot)$. Note that this does not
 529 induce additional correlations in our samples since z_t is sampled independently at each
 530 time step.

531 4 Numerical experiments

532 In this section, we present a detailed study of the performance of the generative modelling
 533 approaches outlined above, using the 2D XY model as a concrete example. We first
 534 compare the model conditioned on temperature with certain baseline approaches that are
 535 defined below. In the second set of experiments, we test the ability of our model to detect
 536 phase transitions in an unsupervised way by evaluation of $\mathcal{D}(T)$ and $\mathcal{F}_{\text{GAN}}(T)$ in Eqs. (13)
 537 and (14). Then we present results for models conditioned on energy and their integration
 538 with MCMC. In the next set of experiments, we train our models only over configurations
 539 with temperatures that are below and above the critical temperature. We then test both
 540 classes of models over the complete range of temperatures, i.e., investigate how well it can
 541 interpolate over unseen temperatures near criticality.

542 4.1 Generation of training data

543 In this work, we use lattices of size $N \times N$, where $N = \{8, 16\}$. The training data is
 544 obtained using the MH algorithm for 32 uniformly spaced values of temperature T in
 545 the range $[0.05, 2.05]$. For each value of T , 10000 configurations are generated. Starting
 546 from a randomly initialized state for each T , a sufficiently large number of configurations
 547 are rejected initially, to account for thermalization. A configuration is included in the
 548 training data set after every 120 MCMC steps for 8×8 and after 400 steps for 16×16
 549 lattice, to reduce correlations in the training data. The angle at each lattice site is scaled
 550 down linearly from $[0, 2\pi)$ to $[0, 1)$. Thus each configuration is a 2D matrix with each
 551 entry between $[0, 1)$. The data is then characterized by investigating the distribution of
 552 observables like magnetization \mathbf{m} , energy E , and vorticity V , all as a function of T . The
 553 samples generated via MCMC as well as the estimated observables serve as the ground
 554 truth for evaluations.

555 4.2 Evaluation metrics

556 How do we know whether and to which extent the ensemble of generated configurations
 557 follow the true distribution? To evaluate, we compute the aforementioned observables
 558 using generated samples, and compare the distribution of these observables with the dis-
 559 tribution of those obtained from MCMC simulations. To compare these distributions, we
 560 deploy the following measures on the histograms of observables generated for 500 different
 561 configurations.

562 4.2.1 Percentage overlap (%OL)

563 Our first measure is %OL, which corresponds to the overlap between two histograms, each
 564 of which is normalized to unit sum. Mathematically, the %OL of two distributions P_r and
 565 P_θ is calculated as:

$$\%OL(P_r, P_\theta) = \sum_i \min(P_r(i), P_\theta(i)), \quad (18)$$

566 where i is the bin index. We use 40 bins in the range $[0,1]$ for the histogram of magneti-
 567 zation and 80 bins in the range $[-2,0]$ for energy. It is not a self-sufficient measure in the
 568 sense that the %OL between the histograms can be quite small even though the computed
 569 values of observables are sufficiently close to each other.

570 4.2.2 Earth mover distance (EMD)

571 The second measure of the distance between two probability distributions we use is EMD
 572 with the following interpretation: if the distributions are thought of as two different ways
 573 of piling up a certain amount of dirt, the EMD is the minimum cost of turning one pile into
 574 the other. Here, the cost is assumed to be the amount of dirt moved times the distance by
 575 which it is moved. The EMD $W(P_r, P_\theta)$ between two distributions P_r and P_θ of a scalar
 576 observable y is defined as

$$W(P_r, P_\theta) = \sum_{x=-\infty}^{\infty} \left| \sum_{y=-\infty}^x (P_r(y) - P_\theta(y)) \right|.$$

577 4.3 Baseline models for comparison

578 We perform a series of numerical experiments to test the effectiveness of the proposed
 579 methods. For comparison, we use modifications and extensions of the method of [34] as
 580 our two baselines, which provide a reference for the performance of our proposed Implicit-
 581 GAN approach.

582 4.3.1 C-HG-VAE

583 The first baseline model we use is C-HG-VAE. It is a prescribed generative model and was
 584 proposed in [34], referred to by them as HG-VAE. Being the (to the best of our knowledge)
 585 only available generative model which has been designed specifically for sampling the 2D
 586 XY model, it is the most natural starting point for us to construct a baseline model.

587 The C-HG-VAE employs CNNs instead of fully connected networks to account for
 588 translational symmetry of the physical system. To improve the agreement of thermo-
 589 dynamic observables with the ground truth, Ref. [34] modified the standard VAE loss
 590 function by additionally including the following term:

$$\mathcal{L}_H = [e(\boldsymbol{\theta}) - e(\hat{\boldsymbol{\theta}})]^2, \quad (19)$$

591 which involves the energies $e(\boldsymbol{\theta})$ and $e(\hat{\boldsymbol{\theta}})$ per lattice site of the ground truth ($\boldsymbol{\theta}$) and the
 592 generated configurations ($\hat{\boldsymbol{\theta}}$), respectively. A multivariate standard normal distribution
 593 was chosen as the prior $P(z)$ and, during training, the input spin configuration to the
 594 encoder is $\mathbf{s} = \{\theta_i\} \in \mathbb{R}^{N \times N}$. For the ease of implementation with standard CNN libraries,
 595 the input is formatted as two channels, one consisting of the spin configuration and the
 596 other consisting of T . This format has also been used by AlphaGo [68]. The output of the
 597 decoder (i.e., reconstruction layer) is split into two terms μ and σ corresponding to the
 598 parameters of a Gaussian distribution. Configurations were generated by sampling from
 599 the Gaussian $\mathcal{N}(\mu_i, \sigma_i)$, $\mu \in \mathbb{R}^{N \times N}$, $\sigma \in \mathbb{R}^{N \times N}$, with each lattice site i distributed indepen-
 600 dently. In the abbreviation HG-VAE, H refers to the \mathcal{L}_H term and G to the Gaussian
 601 parametric specification of the reconstruction layer. HG-VAE generates new configura-
 602 tions using z sampled from the approximately learned variational distribution $Q_\phi(z|x)$ and
 603 then feeds these z to the decoder. Generating z from $Q_\phi(z|x)$ requires use of MC samples
 604 for that corresponding temperature. Hence, their method cannot generate configurations
 605 for temperatures not in the training data. But since our goal is to generate configurations
 606 even for temperatures for which no training data is available, we modify their method to a
 607 conditional model named C-HG-VAE by providing additional information of temperature
 608 to both encoder and decoder. For generating new configurations, we provide $z \sim \mathcal{N}(0, I)$
 609 and T to the decoder. T is concatenated multiple times with z so as the decoder does
 610 not ignore this information along with multiple z . The block diagram representation of
 611 C-HG-VAE is the same as that of the C-VAE in Fig. 1a.

612 4.3.2 C-GAN

613 As second baseline model, we use a prescribed form of a standard C-GAN, introduced in
 614 Sec. 2.2. The C-GAN employing CNNs was trained on the space of angles to reconstruct
 615 configurations, given T . The input to the generator consists of T concatenated with $\mathbf{z} \in \mathbb{R}^N$
 616 sampled from a Gaussian prior, where N is the linear lattice size. Similar to C-HG-VAE,
 617 the generator outputs $\boldsymbol{\mu}_i \in \mathbb{R}^{N \times N}$ and $\boldsymbol{\sigma}_i \in \mathbb{R}^{N \times N}$ corresponding to the parameters of a
 618 Gaussian distribution from which the configurations are sampled. The reparametrization
 619 trick [57] is used to ensure differentiability of the network. The input of the discriminator
 620 has two channels—one consisting of the spin configurations x and the other of T . The
 621 output of the discriminator is a scalar distinguishing the real from the fake sample.

622 4.4 Proposed method: ImplicitGAN

623 This is the proposed implicit C-GAN approach. While all of the key components of this
 624 method have been motivated and explained in detail in Sec. 3.2 above, we here provide a
 625 concise summary of it:

- 626 1. The angles θ_i of the spins in each sample are shifted, $\theta_i \rightarrow \theta_i + \theta_0$, such that the net
 627 magnetization vector (\mathbf{m}) always points in the direction corresponding to $\theta_i = 0$.
- 628 2. The reconstruction layer of the generator consists of two channels $[x_i, y_i]$, which we
 629 normalize at each site as $[x_i, y_i] \rightarrow [x_i, y_i] / \sqrt{x_i^2 + y_i^2}$. The input of the discriminator
 630 has 3 channels, with the first two channels consisting of cosines and sines of lattice
 631 angles and the 3rd channel containing conditional variable, T or E .
- 632 3. To take into account the periodic boundary conditions of the lattice, we use periodic
 633 padding of size 1 for the input layer of the discriminator.
- 634 4. To minimize the biases, Eq. (9) was used as objective function. The value of λ was
 635 chosen to be 10 for 8×8 and 1 for 16×16 lattices.

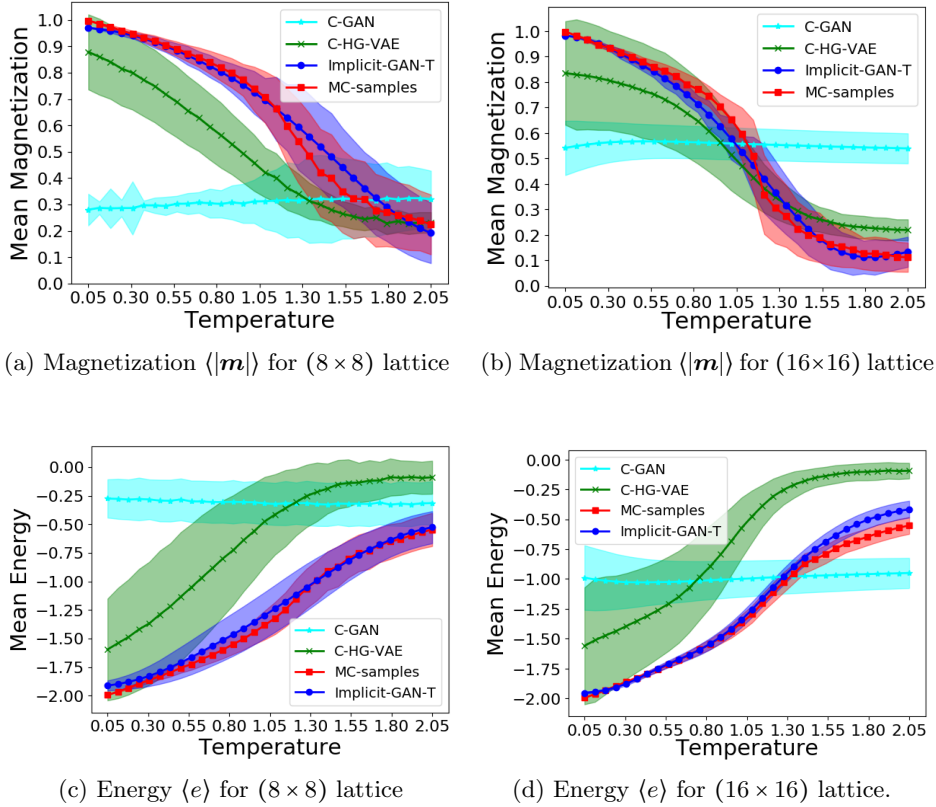


Figure 2: Expectation values (dots and lines) of observables (normalized per site) computed from samples generated by the indicated methods as a function of temperature. Shaded portions represent the standard deviation of the corresponding observable. MC samples are taken as the ground truth; the method giving more overlap with the ground truth is better.

636 5. To maximize the entropy of the generated samples, the output layer of the discrim-
 637 inator now has two outputs, $A(\hat{T}|G(z, c))$ and $D(x)$, with learning objective given
 638 in Eq. (12). The value of γ was chosen to be 100 and 10 for 16×16 and 8×8 lattices,
 639 respectively.

640 Below, we use “ImplicitGAN- T ” to refer to the situation that samples are generated by
 641 the GAN trained conditioned on $c = T$. While “ImplicitGAN- E ” indicates that sampling
 642 is performed by local-update MCMC for a given T combined with over-relaxation steps
 643 with the ImplicitGAN- E model as explained in Sec. 3.4.

644 4.5 Results

645 4.5.1 Comparison with baselines: matching observables

646 For comparison with baselines, the trained temperature-conditioned models described in
 647 Sec. 4.3 were tested by computing observables, namely magnetization and energy, over the
 648 generated configurations. Fig. 2 illustrates mean magnetization $\langle |m| \rangle$ and mean energy
 649 $\langle e \rangle$ values as a function of T . We can notice that $\langle |m| \rangle$ decreases and $\langle e \rangle$ increases with
 650 T for all methods except C-GAN. This shows that C-GAN fails completely to capture
 651 the statistics of the data it is supposed to generate. We can also see that the distribu-
 652 tion of ImplicitGAN- T -generated observables is much closer to the ground truth (MC) as

Table 1: Evaluation metrics, as defined in Sec. 4.2, along with standard deviation, computed over 500 configurations and averaged across all temperatures. Smaller EMD and higher %OL are better. Best values are indicated in bold.

Metric	Lattice size	C-GAN	C-HG-VAE	ImplicitGAN- T
EMD	8×8	0.358 ± 0.246	0.157 ± 0.086	0.038 ± 0.024
Magnetization	16×16	0.152 ± 0.056	0.118 ± 0.028	0.041 ± 0.043
EMD	8×8	0.484 ± 0.250	0.256 ± 0.063	0.022 ± 0.012
Energy	16×16	0.233 ± 0.140	0.296 ± 0.060	0.010 ± 0.005
%OL	8×8	29.31 ± 33.35	52.18 ± 19.15	76.69 ± 6.46
Magnetization	16×16	7.97 ± 16.39	42.78 ± 17.33	67.34 ± 20.41
%OL	8×8	9.43 ± 13.94	10.29 ± 5.43	68.28 ± 20.72
Energy	16×16	13.64 ± 19.33	0.62 ± 0.03	65.83 ± 18.35

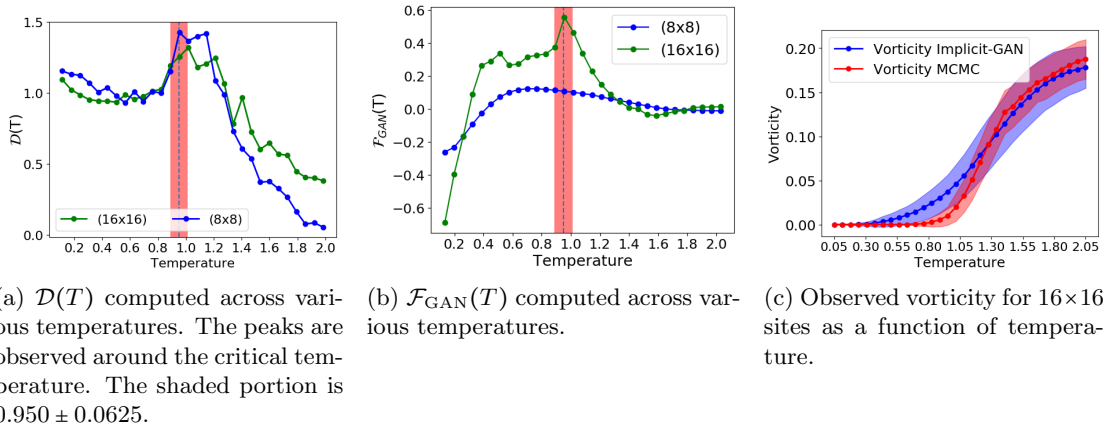


Figure 3: Detection of BKT phase transition (a,b) directly from measures defined in terms of the networks (unsupervised) and (c) by evaluation of the vorticity of generated samples.

653 compared to that of C-HG-VAE generated observables. These results, with the metrics
 654 averaged across temperatures, are quantified in Table 1. The implicit-GAN- T produces
 655 the best results over all the metrics as well as lattice sizes. Our ablation analysis presented
 656 in Appendix A shows which of the different improvements of the method were particularly
 657 crucial in enhancing the performance.

658 4.5.2 Detecting phase transitions

659 We now analyze the ability of the model to detect phase transitions by analyzing its
 660 susceptibility to changes in temperature using the two measures introduced in Sec. 3.3.

661 We begin with \mathcal{D} in Eq. (13) which is plotted in Fig. 3a with $\Delta T = 0.0625$, computed
 662 over 500 configurations produced by the generator. We observe that it exhibits peaks in
 663 the vicinity of the expected phase transition. However, there is no clear maximum, but
 664 rather a double-peak feature. Also the finite-size scaling is opposite to what one would
 665 expect, since the double-peak features move to larger rather than smaller temperatures
 666 with increasing N . More dramatically, the trend does not indicate that these features
 667 approach the true location of the transition at large N as they are further away from the
 668 BKT transition temperature for larger N .

669 Due to these shortcomings of \mathcal{D} for detecting the BKT transition in our GAN archi-
 670 tecture, we here focus on the second measure—the GAN-fidelity—introduced in Eq. (14)

671 with corresponding plot in Fig. 3b, using $\Delta T = 0.0625$. For the larger system size, we
 672 here observe a clear, isolated peak very close (at around $T \approx 0.95$) to the expected transi-
 673 tion temperature for that system size. For the smaller system size, the peak gets broader
 674 and is also shifted to the left. While the broadening is a natural feature of smaller N ,
 675 the shift of its maximum is not the expected finite-size scaling trend—this is similar to
 676 \mathcal{D} , but now seems to approach the correct value with increasing N . One reason for the
 677 unexpected trend in the peak position could be that \mathcal{F}_{GAN} is more reliable for the GAN
 678 with the larger system size: we found that, at lower N , the discriminator is not as suc-
 679 cessful in determining fake samples (we find $\mathbb{E}[D(G(z; T), T)]$ around 0.45 for $N = 8$ as
 680 opposed to around 0.15 for $N = 16$). Note that the negative values of \mathcal{F}_{GAN} at very low T
 681 are clearly unphysical and just related to the fact that the generator underestimates the
 682 magnetization slightly at low temperatures, see Fig. 2(a,b).

683 Notwithstanding these issues, it is encouraging to see that we can capture the phase
 684 transition without prior knowledge of the underlying relevant observable, using the simple
 685 measure \mathcal{F}_{GAN} that is readily evaluated once the generative model has been trained.
 686 Further work, however, is required to see what the advantages and limitations of this
 687 approach are and to understand the finite size scaling behavior in the XY and other
 688 models. Likely, a combination with unsupervised clustering algorithms, e.g., that of [49],
 689 can provide additional assistance in detecting phase transitions in an unsupervised way.
 690 We leave a detailed, system-size-dependent study of these aspects for future work.

691 On top of being able to capture the phase transition in an unsupervised way, we are
 692 dealing with a generative model. Consequently, in cases where we do know the physical
 693 quantity capturing the phase transition, we can also directly compute it with the samples
 694 generate by the networks. In the case of the 2D XY model, the transition is characterized
 695 by the suppression (proliferation) of vortices when entering the low-temperature (high-
 696 temperature) phase. For this reason, we have computed the number of vortices as a
 697 function of temperature, both in the generated and in the MCMC samples; as can be seen
 698 in Fig. 3c, we find good agreement. This shows that the Implicit-GAN approach can,
 699 indeed, capture topological excitations, which have cause problems in other applications
 700 of neural networks [44].

701 4.5.3 Models conditioned on energy

702 We next test the procedure introduced in Sec. 3.4 of using energy-conditioned models
 703 for over-relaxation steps in the context of the 2D XY model. In terms of training and
 704 architecture, the only difference to ImplicitGAN- T is that the prior distribution was chosen
 705 to be uniform in $[-1, 1]$ (instead of Gaussian) and that $e(\boldsymbol{\theta})$ was provided as conditional
 706 information (instead of temperature), which we have shifted by 1.0 so that its mean value
 707 is around zero over the temperature range. The same training data, see Sec. 4.1, was used.

708 To solve Eq. (15), only 3 iterations according to Eq. (16) with $\delta = 0.075$ in Eq. (17)
 709 were performed. If the best of these 3 iterations did not yield a configuration with ΔE
 710 less than the chosen ΔE_{thr} , the over-relation step was dropped. A temperature-dependent
 711 threshold ΔE_{thr} linearly increasing from $[1/N^2, 8/N^2]$ across 32 temperatures was used
 712 in our numerics. For stability purposes, gradients clipping between $[-0.02, 0.02]$ was also
 713 done.

714 Naturally, if only very few over-relaxation steps are performed, it will be very difficult to
 715 see in the data whether ImplicitGAN- E biases the Monte-Carlo chain and leads to incorrect
 716 results. For that reason, we focused our experiments on the regime where significant biases
 717 would be apparent if they were present and performed only $n_M = 2N$ (recall N is the *linear*
 718 system size) local updates in between over-relaxation steps. Nonetheless, as can be seen in
 719 Fig. 4, there is very good agreement between the ground truth (pure MCMC simulations)

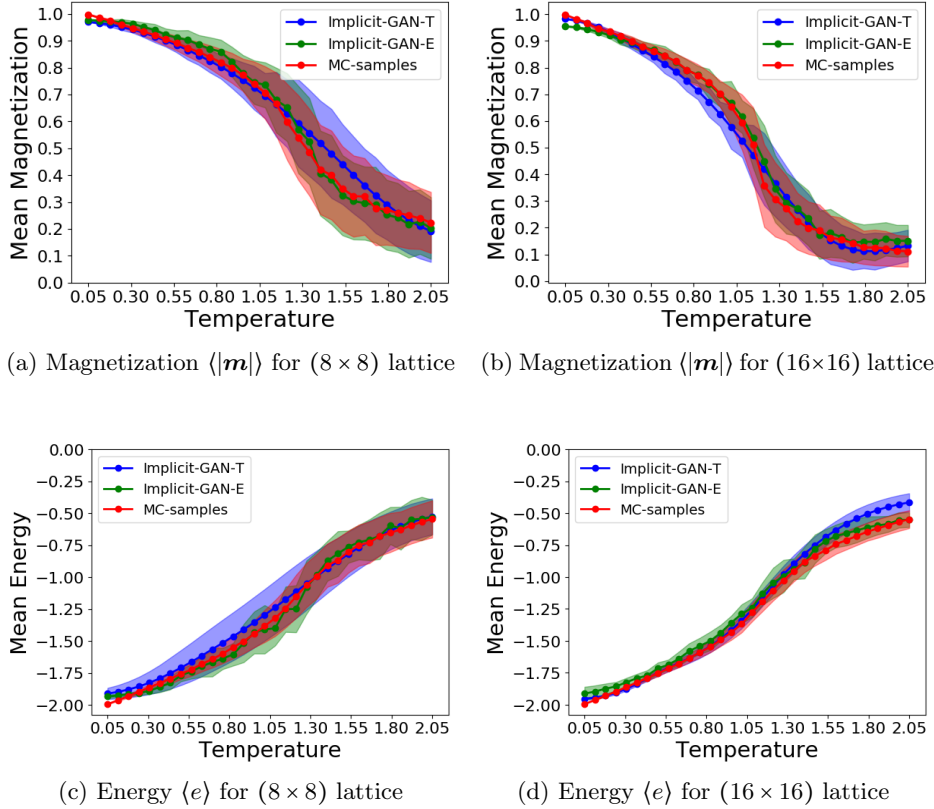


Figure 4: Comparison of energy-conditioned models integrated with MCMC and temperature-conditioned (direct sampling) models with ground truth (MC samples). Symbols and lines indicate average values and shaded portion the standard deviation of the corresponding observable as a function of temperature.

720 and our heavily GAN-over-relaxed simulations. As we show in Appendix B, the additional
 721 over-relaxation steps reduce the correlations significantly between subsequent samples in
 722 the Monte-Carlo chain.

723 To reduce the thermalization time, the Markov chain is initialized by generating samples
 724 from Implicit-GAN-E itself. The input e needed for the initial configuration is obtained
 725 for a given T by a linear approximation of the energy vs. temperature curve of MC samples
 726 (Figs. 4c, 4d). Other initializations, including random initialization, give similar results,
 727 but need higher burnout.

728 4.5.4 Interpolating across unseen temperatures around T_c

729 After having obtained architectures capable of modelling the joint distribution of spin
 730 configurations across temperatures, we next test whether these models can also generate
 731 samples in the vicinity of the phase transition without having been trained on samples
 732 in that regime—a much more challenging problem. We define the critical region as $T \in$
 733 $[0.75, 1.25]$. Note that the critical temperature is $T_c \approx 0.89$ [69] for large system sizes,
 734 $N \rightarrow \infty$; due to logarithmic finite-size corrections, we expect it to be larger, about 0.95,
 735 for our system sizes [44].

736 To test this, we trained a new ImplicitGAN model for both classes of conditional
 737 models discussed above, on the configurations for temperatures in the interval $[0.05, 0.75] \cup$
 738 $[1.25, 2.05]$, i.e., outside the critical region. This corresponds to a 25% reduction in training

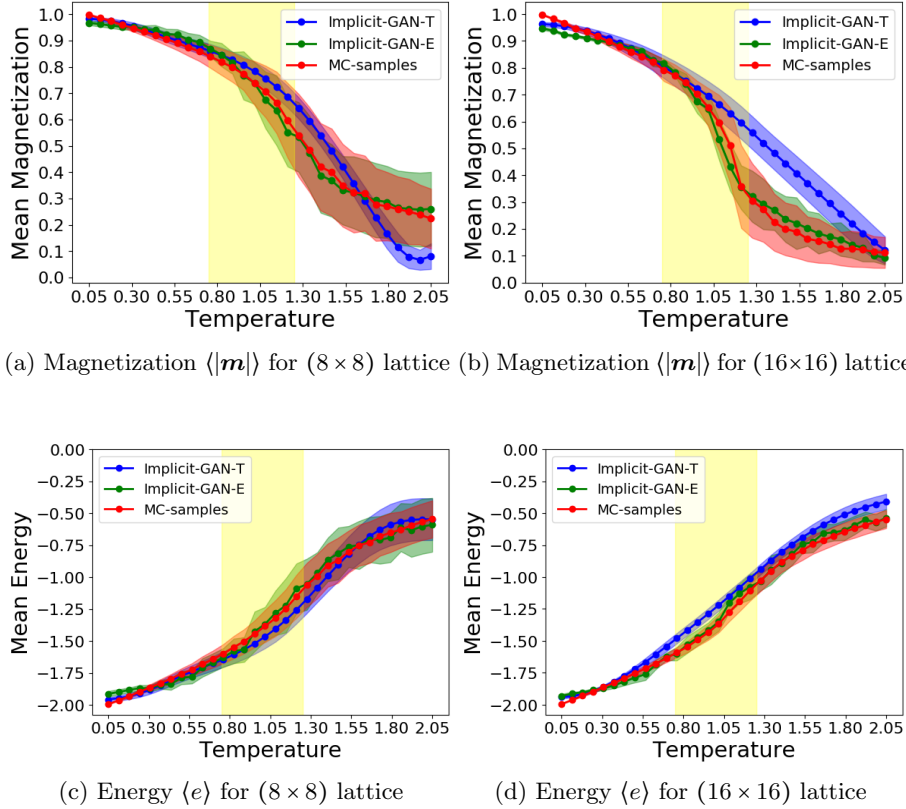


Figure 5: Same as Fig. 4, but no training data was provided in the region $T \in [0.75, 1.25]$ (highlighted in yellow) in the vicinity of the transition.

739 data. Then we test our model by also interpolating for the temperatures which are not
 740 even present in the training data.

741 The results are presented in Fig. 5, where all hyper-parameters were kept the same as
 742 before. One can see that both Implicit-GAN- T and Implicit-GAN- E still capture the main
 743 tendencies of the data, although the former has significantly reduced accuracy in magneti-
 744 zation. The performance of the latter, however, is almost unaffected. Consequently, using
 745 GANs to enhance MCMC simulations is even possible when no training data is provided
 746 in the critical region.

747 5 Conclusions

748 In this work, we have studied different deep-learning-based approaches for generating spin
 749 configurations. We have discussed in detail several modifications of the basic models in
 750 order to warrant a more efficient representation of the states, that, e.g., takes into account
 751 symmetries of the system and the geometry of the local degrees of freedom. Furthermore,
 752 the correlations between the samples generated by the model are shown to be reduced
 753 by incentivizing our model to increase the entropy of the learnt distribution. Although
 754 the approaches used are more generally applicable, we employed the 2D XY model to
 755 benchmark the models' performances. To this end, samples were generated using MCMC
 756 to train the models. MCMC was also used to provide the ground truth to compare
 757 the generated samples with. For the latter, we investigated the histograms of relevant
 758 observables—magnetization, energy, and vorticity. Overall, we found that implicit mod-

els perform better and, in particular, our proposed ImplicitGAN outperforms all other machine-learning models considered.

We have focused on conditional models, which, after training, can be used to generate configurations for in principle arbitrary values of tuning parameters—in our case temperature and energy. We demonstrate that this can be employed for generating configurations near criticality, even without providing training data in the vicinity of the transition. This could be useful for circumventing or at least mitigating critical slowing down in MCMC simulations. It also provides the perspective that, instead of storing a huge amount of samples for an interesting model, one could just store (and make publicly available) a precisely trained neural network to generate samples for future use. We further hope that, when applied to experimental data, it can be used to gain insights about parameter regimes inaccessible in the lab. For these applications, the flexibility of conditional models could prove crucial, since they allow for a multitude of possible conditional variables associated with samples, including complex post-selection criteria.

Finally, we have also shown that conditional models themselves can be employed to detect phase transitions, without any prior knowledge, by investigating the networks' susceptibility to parameter changes. Most importantly, we propose a GAN fidelity measure \mathcal{F}_{GAN} that can be readily evaluated for any trained GAN and is demonstrated to peak in the vicinity of transitions, in analogy to the well-known quantum fidelity measure and its thermal extensions [66]. We hope that this can supplement unsupervised clustering algorithms, such as that of Ref. [49], for future machine-learning-based studies of phase transitions. One could also explore interpretable ML models [70] to extract the crucial physical aspects, such as order parameters or defect proliferation, underlying the phase transition. On a more general level, this illustrates the advantages of the additional “tuning parameter” c of conditional models, which further opens up the possibility to study phase transitions in one and the same neural network as a function of c . One might wonder whether (and what kind of) different universality classes of transitions can be established in conditional networks.

In the future, we are also planning to further test and refine the ImplicitGAN approach, by applying it to other classical models and systematically studying the behavior of observables and \mathcal{F}_{GAN} with increasing system size. Additional directions include developing models conditioned on system size and exploring quantum mechanical systems.

Acknowledgements

Funding information MS acknowledges support from the National Science Foundation under Grant No. DMR-1664842.

Note added—During the final stages of the completion of this project, another work appeared on arXiv [71], where a different generative ML technique is applied to the 2D XY model. The emphasis of this work is different from ours and, in particular, does not contain the analysis of implicit and prescribed models, the application as an over-relaxation step, nor that of network-based unsupervised indicators (\mathcal{D} and \mathcal{F}_{GAN}) of the phase transition, but instead relies on the helicity modulus.

A Ablation analysis

In this appendix, we perform a detailed ablation analysis for the temperature-conditioned models, to examine the effect of each of the components of our proposed Implicit-GAN

803 approach, see Sec. 3 and Sec. 4.4, separately. For the sake of comparison, we average the
 804 values of the metrics defined in Sec. 4.2 across all the temperatures used in the training
 805 data and we name our models as

- 806 1. C-GAN: The standard prescribed C-GAN, which is also used as a baseline (Sec. 4.3).
- 807 2. C-GAN₁: A standard implicit C-GAN modeling θ_i using the angles θ_i rather than the
 808 two-component unit vectors \mathbf{s}_i as input. The generator is a deterministic function
 809 of z and outputs the angles θ_i .
- 810 3. C-GAN₂: It is same as the C-GAN₁ model but trained using $\mathbf{s}_i = (\cos \theta_i, \sin \theta_i)$ as
 811 input. It also includes periodic padding of size 1 but the total magnetization of each
 812 sample of the training data was not rotated to point along the x-axis.
- 813 4. C-GAN₃: It is same as C-GAN₂ with magnetization direction normalization as in
 814 Eq. (8).
- 815 5. C-GAN₄: Same as C-GAN₃ but the training objective is now modified according to
 816 Eq. (9), in order to minimize the output bias.
- 817 6. Implicit-GAN: This is the proposed implicit C-GAN as was used in Sec. 4.4 in the
 818 main text. It is the same as C-GAN₄ but with the entropy-regularized objective of
 819 Sec. 3.2.2.

820 The performance of each of these models over the metrics is given in Table 2. A com-
 821 parison between C-GAN and C-GAN₁ illustrates that, keeping other factors the same,
 822 implicit models perform better than prescribed models. Accounting for the continuity of
 823 the space of angles and the periodic boundary conditions further improves the performance
 824 as can be seen by comparing C-GAN₁ with C-GAN₂. Exploiting the global spin-rotation
 825 symmetry of the XY model brings further improvement in the agreement of the observ-
 826 ables, as is visible from the performance of C-GAN₃. Thus, discontinuous jumps of θ_i at
 827 2π and not taking into account periodic boundary conditions and spin-rotation symmetry
 828 seem to be important factors causing the bad performance of C-GAN₁ and C-GAN₂. Con-
 829 sistent with [34], we observed that this was not a serious problem when (unconditioned)
 830 GANs were trained only for a single temperature.

831 We see that the performance of C-GAN₄ is comparable to C-GAN₃ for the metrics in
 832 Table 2. However, one has to note that these metrics are not directly sensitive to whether
 833 the generator satisfies the constraint of total magnetization pointing along the x axis,
 834 $\sum_i \sin(\theta_i)/N^2 = 0$; the additional term $\propto \lambda$ in Eq. (9) explicitly incentivizes the generator
 835 to obey the constraint. To test this, we compare the average values of the y-component of
 836 the magnetization, before (C-GAN₃) and after (C-GAN₄) adding the term $\propto \lambda$. Figure 6
 837 shows a significant reduction in the average ‘bias’, as with C-GAN₄ the curves are closer
 838 to x-axis. This can be considered as a first-order moment matching test to check whether
 839 the model learns the true distribution of the samples, which were reprocessed according
 840 to Eq. (8). The parameter $\lambda \approx 1 - 10$ was observed to work well. With a large value
 841 of λ (≈ 100), the average bias across temperatures becomes small but the performance of
 842 the model over the metrics starts degrading. Hence, there exists a trade-off between the
 843 performance and bias.

844 Finally, we can see in Table 2 that the performance of Implicit C-GAN, in terms of
 845 reproducing the distribution of observables, is comparable to that of C-GAN₃ and C-
 846 GAN₄ for magnetization and seems to become even better for the energy. On top of that,
 847 the key advantage of the Implicit-GAN is that it generates more uncorrelated samples

Table 2: **Ablation analysis:** Evaluation metrics, along with standard deviation, computed over 500 configurations of a 16×16 lattice, averaged across all temperatures. Smaller EMD and higher %OL are better.

Metric	EMD Mag.	EMD Energy	%OL Mag.	%OL Energy
C-GAN	0.304 ± 0.113	0.234 ± 0.14	7.969 ± 16.394	16.643 ± 13.863
C-GAN₁	0.290 ± 0.101	0.212 ± 0.122	20.6 ± 21.275	18.381 ± 8.303
C-GAN₂	0.136 ± 0.04	0.098 ± 0.064	41.181 ± 21.295	35.269 ± 23.922
C-GAN₃	0.071 ± 0.075	0.034 ± 0.028	67.068 ± 16.092	47.25 ± 21.815
C-GAN₄	0.043 ± 0.038	0.041 ± 0.035	69.275 ± 22.586	37.181 ± 22.397
ImplicitGAN-T	0.041 ± 0.043	0.010 ± 0.005	67.343 ± 20.415	65.832 ± 18.351

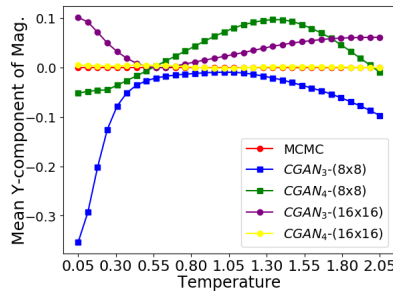


Figure 6: Average value of Y-component of magnetization computed over 500 configurations. Due to pre-processing of the MCMC data, the curves should be close zero.

848 as compared to the latter. To quantify this, we measure correlations between a pair of
 849 samples, $\theta = \{\theta_j\}$ and $\theta' = \{\theta'_j\}$, generated by our models. To this end, we introduce

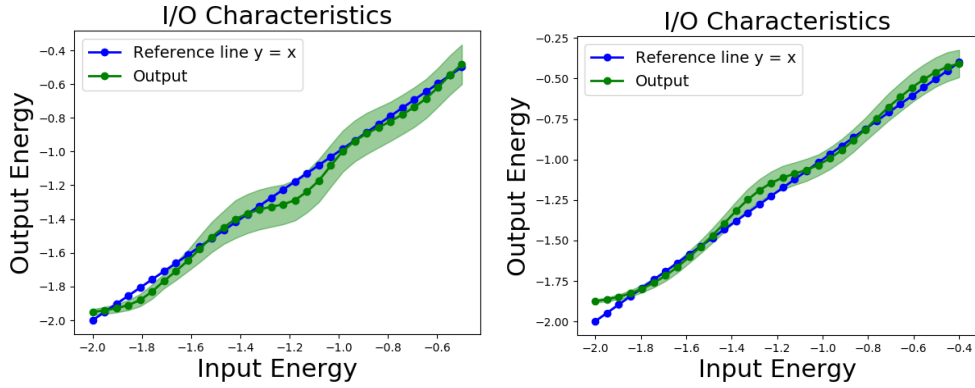
$$\kappa(T) = \frac{1}{N^2} \sum_j \left| \mathbb{E} \left[e^{i(\theta_j - \theta_0)} e^{-i(\theta'_j - \theta'_0)} \right] \right| \quad (20)$$

850 as our measure for the *average cross-correlation*. Here, $\theta_0 = \sum_j (\theta_j / N^2)_{2\pi}$ and $\theta'_0 =$
 851 $\sum_j (\theta'_j / N^2)_{2\pi}$ to make sure that we do not get $\kappa \approx 0$ simply because we have exploited
 852 the global spin-rotation symmetry, see Sec. 3.1.1. The expectation value in Eq. (20) is
 853 taken with respect to the configurations generated by the models.

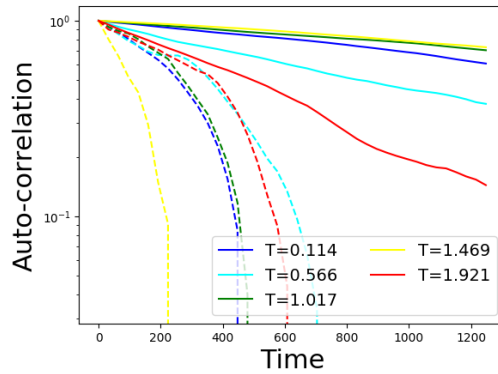
854 For instance, from C-GAN₃ to Implicit-GAN, we obtain an improvement from $\kappa =$
 855 0.65 ± 0.38 to $\kappa = 0.27 \pm 0.2$ at $T = 1.5$ and for $N = 16$. We observed a significant reduction
 856 in cross-correlation as compared to C-GAN₃ for both 8×8 and 16×16 lattices and across
 857 temperatures. Nonetheless, a comparison with the ground truth (MC) still reveals an
 858 enhanced κ in the disordered high-temperature phase, which means that the Implicit-
 859 GAN generated samples are not perfect and do not completely explore the state space.

860 B Characteristics of ImplicitGAN-E

861 We here present more details on the properties of our ImplicitGAN-E models. As already
 862 mentioned in Sec. 3.4, GANs learn distributions only approximately. As such, the energy
 863 of the states generated by $G(z, e)$, $z \sim P(z)$, will have energy densities only approximately
 864 equal to e . To quantify this, we plot in Fig. 7a the I/O characteristic of our ImplicitGAN-
 865 E models, i.e., the distribution of $E(G(z, e))/N^2$, $z \sim P(z)$, as a function of e . As can
 866 be seen, $E(G(z, e))/N^2$ and e clearly follow each other, but systematic deviations exist.



(a) Input-Output Characteristics for (8×8) and (16×16) lattices respectively.



(b) Auto-correlation, as defined in Eq. (21), as a function of the number of local updates for 5 different temperatures and $N = 16$. Solid lines are MCMC with local updates and dashed lines is MCMC with over-relaxation.

Figure 7: Characteristics of our ImplicitGAN- E model.

867 These observations show that the use of conditional models can accelerate the search for
 868 states with the desired energy significantly, as compared to regular GANs. At the same
 869 time, we also learn that fine-tuning e to obtain the required energy with high precision
 870 via Eq. (15) is still important; otherwise, we would obtain systematic deviations in our
 871 Markov chain.

872 To explicitly demonstrate that the use of ImplicitGAN- E as over-relaxation steps de-
 873 creases the correlations between samples in the Markov chain, we here compute the fol-
 874 lowing auto-correlation function:

$$R_m(\tau) = \left[\frac{\sum_{i=1}^{M-\tau} m_i m_{i+\tau} - (M-\tau) \langle m \rangle_{[1, M-\tau]} \langle m \rangle_{[\tau+1, M]}}{\langle m^2 \rangle_{[1, M]} - \langle m \rangle_{[1, M]}^2} \right], \quad (21)$$

875 where m_i denotes the value of $|\mathbf{m}|$ in the i th sample in the Markov chain and $\langle m \rangle_{[j_1, j_2]} =$
 876 $\frac{1}{[j_2 - j_1 + 1]} \sum_{j=j_1}^{j_2} m_j$. A plot of the auto-correlation function $R_m(\tau)$ with and without the
 877 over-relaxation step is shown in Fig. 7b. Clearly, at all temperatures, the addition of
 878 GAN-based over-relaxations steps significantly reduces the correlations of samples in the
 879 Markov chain.

References

- 880
- 881 [1] R. Salakhutdinov, *Learning deep generative models*, Annual Review of Statistics and
882 Its Application **2**(1), 361 (2015), doi:10.1146/annurev-statistics-010814-020120.
- 883 [2] L. Wang, *Generative Models for Physicists* (2018), [http://wangleiphy.github.io/
884 lectures/PILTutorial.pdf](http://wangleiphy.github.io/lectures/PILTutorial.pdf).
- 885 [3] Z. Ou, *A Review of Learning with Deep Generative Models from Perspective of Graphical
886 Modeling*, arXiv e-prints (2018), 1808.01630.
- 887 [4] J. Gui, Z. Sun, Y. Wen, D. Tao and J. Ye, *A Review on Generative Adversarial
888 Networks: Algorithms, Theory, and Applications*, arXiv e-prints (2020), 2001.06937.
- 889 [5] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **58**, 86 p. 5 (1987).
- 890 [6] U. Wolff, Phys. Rev. Lett. **361**, 62 (1989).
- 891 [7] B. S. N. Prokofev and I. Tupitsyn, Phys. Lett. A **238** p. 253 (1998).
- 892 [8] G. L. H. G. Evertz and M. Marcu, Phys. Rev. Lett. **70** p. 875 (1998).
- 893 [9] H. G. Evertz, Advances in Physics p. 52 (2003).
- 894 [10] O. F. S. asen and A. W. Sandvik, Phys. Rev. E **66**, (2002).
- 895 [11] S. W. F. Alet and M. Troyer, Phys. Rev. E **71**, 036706 (2005).
- 896 [12] V. Dunjko and H. J. Briegel, *Machine learning & artificial intelligence in the quantum
897 domain: a review of recent progress*, Reports on Progress in Physics **81**(7), 074001
898 (2018), doi:10.1088/1361-6633/aab406.
- 899 [13] S. Das Sarma, D.-L. Deng and L.-M. Duan, *Machine learning meets quantum physics*,
900 Physics Today **72**(3), 48 (2019), doi:10.1063/PT.3.4164, 1903.03516.
- 901 [14] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher and D. J.
902 Schwab, *A high-bias, low-variance introduction to machine learning for physicists*,
903 Physics Reports **810**, 1 (2019), doi:https://doi.org/10.1016/j.physrep.2019.03.001.
- 904 [15] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto
905 and L. Zdeborová, *Machine learning and the physical sciences*, Rev. Mod. Phys. **91**,
906 045002 (2019), doi:10.1103/RevModPhys.91.045002.
- 907 [16] R. G. Melko, G. Carleo, J. Carrasquilla and J. I. Cirac, *Restricted boltzmann machines
908 in quantum physics*, Nature Physics **15**(9), 887 (2019), doi:10.1038/s41567-019-0545-
909 1.
- 910 [17] M. J. B. Stavros Efthymiou and R. G. Melko, *Super-resolving the ising model with
911 convolutional neural networks*, PHYSICAL REVIEW B **99**, 075113 (2019).
- 912 [18] J. Liu, Y. Qi, Z. Y. Meng and L. Fu, *Self-learning monte carlo method*, Phys. Rev.
913 B **95**, 041101 (2017), doi:10.1103/PhysRevB.95.041101.
- 914 [19] Z. H. Liu, X. Y. Xu, Y. Qi, K. Sun and Z. Y. Meng, *Itinerant quantum critical
915 point with frustration and a non-fermi liquid*, Phys. Rev. B **98**, 045116 (2018),
916 doi:10.1103/PhysRevB.98.045116.

- 917 [20] X. Y. Xu, Y. Qi, J. Liu, L. Fu and Z. Y. Meng, *Self-learning quantum monte*
918 *carlo method in interacting fermion systems*, Phys. Rev. B **96**, 041119 (2017),
919 doi:10.1103/PhysRevB.96.041119.
- 920 [21] H. Kohshiro and Y. Nagai, *Effective Ruderman-Kittel-Kasuya-Yosida-like interaction*
921 *in diluted double-exchange model: self-learning Monte Carlo approach*, arXiv e-prints
922 (2020), 2005.06992.
- 923 [22] M. S. Albergo, G. Kanwar and P. E. Shanahan, *Flow-based generative mod-*
924 *els for Markov chain Monte Carlo in lattice field theory*, **100**(3), 034515 (2019),
925 doi:10.1103/PhysRevD.100.034515, 1904.12072.
- 926 [23] G. Torlai and R. G. Melko, *Learning thermodynamics with Boltzmann machines*,
927 **94**(16), 165134 (2016), doi:10.1103/PhysRevB.94.165134, 1606.02718.
- 928 [24] A. Morningstar and R. G. Melko, *Deep Learning the Ising Model Near Criticality*,
929 arXiv e-prints (2017), 1708.04622.
- 930 [25] G. Carleo and M. Troyer, *Solving the quantum many-body problem with artificial*
931 *neural networks*, Science **355**(6325), 602 (2017), doi:10.1126/science.aag2302.
- 932 [26] L. Huang and L. Wang, *Accelerated monte carlo simulations with restricted boltzmann*
933 *machines*, Phys. Rev. B **95**, 035105 (2017), doi:10.1103/PhysRevB.95.035105.
- 934 [27] K.-W. Zhao, W.-H. Kao, K.-H. Wu and Y.-J. Kao, *Generation of ice states through*
935 *deep reinforcement learning*, **99**(6), 062106 (2019), doi:10.1103/PhysRevE.99.062106,
936 1903.04698.
- 937 [28] J. M. Pawłowski and J. M. Urban, *Reducing autocorrelation times in lattice simula-*
938 *tions with generative adversarial networks*, Machine Learning: Science and Technol-
939 ogy **1**(4), 045011 (2020), doi:10.1088/2632-2153/abae73.
- 940 [29] K. Mills and I. Tamblin, *Phase space sampling and operator confidence with genera-*
941 *tive adversarial networks*, arXiv e-prints (2017), 1710.08053.
- 942 [30] K. Mills, C. Casert and I. Tamblin, *Adversarial generation of mesoscale surfaces*
943 *from small scale chemical motifs* .
- 944 [31] K. Zhou, G. Endródi, L.-G. Pang and H. Stöcker, *Regressive and genera-*
945 *tive neural networks for scalar field theory*, Phys. Rev. D **100**, 011501 (2019),
946 doi:10.1103/PhysRevD.100.011501.
- 947 [32] Z. Liu, S. P. Rodrigues and W. Cai, *Simulating the Ising Model with a Deep Convo-*
948 *lutional Generative Adversarial Network*, arXiv e-prints (2017), 1710.04987.
- 949 [33] C. Casert, K. Mills, T. Vieijra, J. Ryckebusch and I. Tamblin, *Optical lattice ex-*
950 *periments at unobserved conditions and scales through generative adversarial deep*
951 *learning*, arXiv e-prints (2020), 2002.07055.
- 952 [34] M. Cristoforetti, G. Jurman, A. I. Nardelli and C. Furlanello, *Towards meaningful*
953 *physics from generative models*, arXiv:1705.09524 (2017).
- 954 [35] B. Nosarzewski, *Variational Autoencoders for Classical Spin Models* (2017).
- 955 [36] P. S. S. F. I. Luchnikov, A. Ryzhov and H. Ouerdane, *Variational autoencoder recon-*
956 *struction of complex many-body physics.*, arxiv 1910.03957 (2019).

- 957 [37] D. Wu, L. Wang and P. Zhang, *Solving Statistical Mechanics Using Variational Au-*
958 *toressive Networks*, **122**(8), 080602 (2019), doi:10.1103/PhysRevLett.122.080602,
959 1809.10606.
- 960 [38] O. Sharir, Y. Levine, N. Wies, G. Carleo and A. Shashua, *Deep autoregressive models*
961 *for the efficient variational simulation of many-body quantum systems*, Phys. Rev.
962 Lett. **124**, 020503 (2020), doi:10.1103/PhysRevLett.124.020503.
- 963 [39] X. Ding and B. Zhang, *Computing Absolute Free Energy with Deep Generative Models*,
964 arXiv e-prints (2020), 2005.00638.
- 965 [40] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K.-R. Müller and P. Kessel,
966 *Asymptotically unbiased estimation of physical observables with neural samplers*,
967 **101**(2), 023304 (2020), doi:10.1103/PhysRevE.101.023304, 1910.13496.
- 968 [41] M. Mirza and S. Osindero, *Conditional Generative Adversarial Nets*, arXiv e-prints
969 arXiv:1411.1784 (2014), 1411.1784.
- 970 [42] G. Torlai, B. Timar, E. P. L. van Nieuwenburg, H. Levine, A. Omran, A. Keesling,
971 H. Bernien, M. Greiner, V. Vuletić, M. D. Lukin, R. G. Melko and M. Endres, *In-*
972 *tegrating neural networks with a quantum simulator for state reconstruction*, Phys.
973 Rev. Lett. **123**, 230504 (2019), doi:10.1103/PhysRevLett.123.230504.
- 974 [43] B. L. Shakir Mohamed, *Learning in implicit generative models.*, arxiv:1610.03483.
975 (2017).
- 976 [44] M. J. S. Beach, A. Golubeva and R. G. Melko, *Machine learning vor-*
977 *tices at the kosterlitz-thouless transition*, Phys. Rev. B **97**, 045207 (2018),
978 doi:10.1103/PhysRevB.97.045207.
- 979 [45] J. M. Kosterlitz and D. J. Thouless, *Ordering, metastability and phase transitions in*
980 *two-dimensional systems*, Journal of Physics C: Solid State Physics **6**(7), 1181 (1973).
- 981 [46] V. Berezinskii, *Destruction of long-range order in one-dimensional and two-*
982 *dimensional systems having a continuous symmetry group i. classical systems*, Sov.
983 Phys. JETP **32**(3), 493 (1971).
- 984 [47] V. Berezinskii, *Destruction of long-range order in one-dimensional and two-*
985 *dimensional systems possessing a continuous symmetry group. ii. quantum systems*,
986 Soviet Journal of Experimental and Theoretical Physics **34**, 610 (1972).
- 987 [48] J. M. Kosterlitz, *The critical properties of the two-dimensional xy model*, Journal of
988 Physics C: Solid State Physics **7**(6), 1046 (1974).
- 989 [49] J. F. Rodriguez-Nieva and M. S. Scheurer, *Identifying topological order through un-*
990 *supervised machine learning*, Nature Physics **15**(8), 790 (2019), doi:10.1038/s41567-
991 019-0512-x.
- 992 [50] Y. Long, J. Ren and H. Chen, *Unsupervised manifold clustering of topological phonon-*
993 *ics*, Phys. Rev. Lett. **124**, 185501 (2020), doi:10.1103/PhysRevLett.124.185501.
- 994 [51] Y. Che, C. Gneiting, T. Liu and F. Nori, *Topological Quantum Phase Transitions*
995 *Retrieved from Manifold Learning*, arXiv e-prints (2020), 2002.02363.
- 996 [52] M. S. Scheurer and R.-J. Slager, *Unsupervised Machine Learning and Band Topology*,
997 **124**(22), 226401 (2020), doi:10.1103/PhysRevLett.124.226401, 2001.01711.

- 998 [53] F. Schäfer and N. Lörch, *Vector field divergence of predictive model output as indi-*
999 *cation of phase transitions*, PHYSICAL REVIEW E **99**, 062107 (2019).
- 1000 [54] K. Kashiwa, Y. Kikuchi and A. Tomiya, *Phase transition encoded in neural network*,
1001 *Progress of Theoretical and Experimental Physics* **2019**(8) (2019), 083A04.
- 1002 [55] A. Tanaka and A. Tomiya, *Detection of Phase Transition via Convolutional Neu-*
1003 *ral Networks*, *Journal of the Physical Society of Japan* **86**(6), 063001 (2017),
1004 doi:10.7566/JPSJ.86.063001, 1609.09087.
- 1005 [56] G. B. F. S. N. L. Eliska Greplova, Agnes Valenti and S. Huber, *Unsupervised identi-*
1006 *fication of topological order using predictive models*, arxiv.org 1910.10124 (2019).
- 1007 [57] C. Doersch, *A tutorial on variational autoencoders*, arxiv:1606.05908 (2016).
- 1008 [58] A. Razavi, A. van den Oord and O. Vinyals, *Generating Diverse High-Fidelity Images*
1009 *with VQ-VAE-2*, arXiv e-prints arXiv:1906.00446 (2019), 1906.00446.
- 1010 [59] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair,
1011 A. Courville and Y. Bengio, *Generative adversarial networks* (2014), 1406.2661.
- 1012 [60] M. Mustafa, D. Bard, W. Bhimji, Z. Lukić, R. Al-Rfou and J. M. Kratochvil, *Cos-*
1013 *mogon: creating high-fidelity weak lensing convergence maps using generative ad-*
1014 *versarial networks*, *Computational Astrophysics and Cosmology* **6**(1), 1 (2019),
1015 doi:10.1186/s40668-019-0029-9.
- 1016 [61] L. Tierney, *Markov Chains for Exploring Posterior Distributions*, *Annals of Statistics*
1017 (1992).
- 1018 [62] C. Wang and H. Zhai, *Machine learning of frustrated classical spin models (ii):*
1019 *Kernel principal component analysis*, *Frontiers of Physics* **13**(5), 130507 (2018),
1020 doi:10.1007/s11467-018-0798-7.
- 1021 [63] A. Makhzani, *Implicit autoencoders.*, arxiv 1805.09804 (2019).
- 1022 [64] D. M. B. Adji B. Dieng, Francisco J. R. Ruiz and M. K. Titsias, *Prescribed generative*
1023 *adversarial networks*, arxiv 1910.04302 (2019).
- 1024 [65] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever and P. Abbeel, *Info-*
1025 *GAN: Interpretable Representation Learning by Information Maximizing Generative*
1026 *Adversarial Nets*, arXiv e-prints (2016), 1606.03657.
- 1027 [66] H. T. Quan and F. M. Cucchietti, *Quantum fidelity and thermal phase transitions*,
1028 *Phys. Rev. E* **79**, 031101 (2009), doi:10.1103/PhysRevE.79.031101.
- 1029 [67] K. Balasubramanian and S. Ghadimi, *Zeroth-order nonconvex stochastic optimiza-*
1030 *tion: Handling constraints, high-dimensionality and saddle-points* (2019), 1809.
1031 06474.
- 1032 [68] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrit-
1033 *twieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe*
1034 *et al.*, *Mastering the game of go with deep neural networks and tree search*, *Nature*
1035 **529**(7587), 484 (2016), doi:10.1038/nature16961.

- 1036 [69] Y. Komura and Y. Okabe, *Large-scale monte carlo simulation of two-dimensional*
1037 *classical xy model using multiple gpus*, Journal of the Physical Society of Japan
1038 **81**(11), 113001 (2012), doi:10.1143/JPSJ.81.113001, [https://doi.org/10.1143/](https://doi.org/10.1143/JPSJ.81.113001)
1039 [JPSJ.81.113001](https://doi.org/10.1143/JPSJ.81.113001).
- 1040 [70] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl and B. Yu, *Definitions, meth-*
1041 *ods, and applications in interpretable machine learning*, Proceedings of the National
1042 Academy of Sciences **116**(44), 22071–22080 (2019), doi:10.1073/pnas.1900654116.
- 1043 [71] L. Wang, Y. Jiang, L. He and K. Zhou, *Recognizing the topological phase transition*
1044 *by Variational Autoregressive Networks*, arXiv e-prints (2020), 2005.04857.