



Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems



Jin-Long Wu^{a,1}, Karthik Kashinath^b, Adrian Albert^b, Dragos Chirila^c, Prabhat^b, Heng Xiao^{a,*}

^a Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24060, USA

^b Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

^c Alfred Wegener Institute, Helmholtz Centre for Polar and Marine Research (AWI), Bremerhaven, Germany

ARTICLE INFO

Article history:

Received 8 May 2019

Received in revised form 4 December 2019

Accepted 16 December 2019

Available online 27 December 2019

Keywords:

Machine learning

Generative adversarial networks

Statistical constraint

Partial differential equations

Rayleigh-Bénard convection

ABSTRACT

Simulating complex physical systems often involves solving partial differential equations (PDEs) with some closures due to the presence of multi-scale physics that cannot be fully resolved. Although the advancement of high performance computing has made resolving small-scale physics possible, such simulations are still very expensive. Therefore, reliable and accurate closure models for the unresolved physics remains an important requirement for many computational physics problems, e.g., turbulence simulation. Recently, several researchers have adopted generative adversarial networks (GANs), a novel paradigm of training machine learning models, to generate solutions of PDEs-governed complex systems without having to numerically solve these PDEs. However, GANs are known to be difficult in training and likely to converge to local minima, where the generated samples do not capture the true statistics of the training data. In this work, we present a statistical constrained generative adversarial network by enforcing constraints of covariance from the training data, which results in an improved machine-learning-based emulator to capture the statistics of the training data generated by solving fully resolved PDEs. We show that such a statistical regularization leads to better performance compared to standard GANs, measured by (1) the constrained model's ability to more faithfully emulate certain physical properties of the system and (2) the significantly reduced (by up to 80%) training time to reach the solution. We exemplify this approach on the Rayleigh-Bénard convection, a turbulent flow system that is an idealized model of the Earth's atmosphere. With the growth of high-fidelity simulation databases of physical systems, this work suggests great potential for being an alternative to the explicit modeling of closures or parameterizations for unresolved physics, which are known to be a major source of uncertainty in simulating multi-scale physical systems, e.g., turbulence or Earth's climate.

© 2019 Elsevier Inc. All rights reserved.

* Corresponding author.

E-mail address: hengxiao@vt.edu (H. Xiao).

¹ Current affiliation: California Institute of Technology, Pasadena, CA 91125, USA.

1. Introduction

Complex physical systems are usually characterized by PDE-governed processes with multi-scale physics. Resolving all the scales of multi-scale processes in simulations is still infeasible for many real applications. In practice, simulating those multi-scale physical systems often involves closures to model unresolved processes. However, these closure models also account for major sources of uncertainties in simulation results, partly due to neglecting high-order statistics from the true physical process. With the advancement of high performance computing in the last decades, high-fidelity simulations of complex multi-scale processes such as fully resolved turbulent flow is now becoming available for a growing number of scenarios. More recently, leveraging existing high-fidelity simulation datasets to build models that can emulate complex multi-scale processes (e.g., turbulence or Earth's climate) has been made possible by the rapid growth of the capabilities of machine learning.

Several machine-learning-based approaches have been proposed for improving the simulation of complex PDE-governed systems, e.g., turbulence modeling [1–3]. Machine-learning-based approaches in other complex systems (e.g., weather and climate modeling) is an area of increasing interest [4–8]. In addition to building closure models, machine learning techniques have been applied to various other science and engineering problems that require advanced data analyses of model output, e.g., classification, detection and segmentation of patterns [9–11]. Although machine learning techniques achieved remarkable successes in applications such as image recognition [12], the science and engineering communities gradually realize that it is important to inform machine learning some physics instead of merely relying on data-driven discovery for many scientific problems. Ling et al. [3] proposed a tensorial neural network to build an objective functional mapping of tensors by embedding a tensor expansion formula into their neural network. Thomas et al. [13] built a more general tensor network to ensure the important equivariance by introducing spherical harmonics as filters. Wu et al. [2] demonstrated that the important invariance (e.g., Galilean invariance) can be preserved by only using invariants as inputs and outputs for machine learning models. Other researchers incorporated known physics to machine learning techniques as additional constraints. For instance, Karpatne et al. [14] incorporated known physical knowledge by adding a penalty term into the machine learning loss function. Rassi et al. [15,16] proposed physics-informed neural networks by enforcing the structure of governing equations. Another promising direction is to explore the relations between the machine learning techniques and traditional frameworks for modeling physics, e.g., the analogy between LSTM network and Mori-Zwanzig formalism, for which preliminary success has been demonstrated by Ma et al. [17]. In addition, Lusch et al. [18] demonstrated that the Koopman theory can be made practical by using autoencoder to identify the linear embedding of nonlinear dynamics.

Recently, a novel architecture known as generative adversarial networks (GANs) has been proposed to generate data that mimics certain properties of images or behaviors of a given system. Specifically, GANs consist of a discriminator that learns properties from training samples and a generator that generates samples that mimic the learned properties. Recent research has shown how GANs can be used to generate new solutions of PDE-governed systems by training on existing datasets. For example, King et al. [19,20] used GANs to create turbulent flow realizations and showed that the generated realizations can capture several statistical constraints of turbulent flows, e.g., Kolmogorov's $-5/3$ law and small scale intermittency of turbulence. Moreover, GANs have been utilized in extracting information from high-fidelity simulations of other physical systems, e.g., cosmology that involves N-body simulations [21]. Although the standard GANs can capture the true distribution of training data when global minimum of the loss function is achieved [22,23], it is well known that GANs can be difficult to train and possibly converges to a local minimum when the complexity of true data distribution increases. Therefore, it is unlikely that standard GANs can capture all the statistics of the solutions for a complex PDE-governed system, indicating that the trained network would be unable to reproduce some important physical and statistical properties of the system. In addition, there are other challenges associated with GANs, e.g., the stability of training [24], the noise in generated samples [25] and the assessment of sample quality [26]. To improve GANs performances for physical problems, Xie et al. [27] incorporated temporal coherence to GANs to generate super-resolution realizations of turbulent flows. Yang et al. [28] encoded into the architecture of GANs the governing physical laws in the form of stochastic differential equations. Stinis et al. [29] augmented the inputs of the discriminator with residuals and introduced noises into training data as inspired by dynamical systems.

Considering the potential of GANs to physical systems and the limitation of standard GANs, it is worthwhile to investigate a general approach to improve the performance of GANs in emulating physical systems by introducing proper regularization. Inspired by the work of Karpatne et al. [14], we envision to embed both physical constraints (e.g., conservation laws) and statistical constraints (e.g., statistics from the data distribution) into the generator, in order to enhance the robustness and generalization capability of GANs-based physical system emulator. This work focuses on investigating the advantage of incorporating statistical constraints. A companion paper of Zeng et al. [30] studied the effects of enforcing physical constraints on GANs. They found that deterministic physical constraints (e.g., conservation laws) can be enforced by incorporating corresponding constraints into the loss function of the generator. They further demonstrated that the constrained GANs have better performance than standard ones even though the deterministic constraints are enforced *approximately*. Such approximate constraints are relevant when only partial knowledge is available on the physical systems to be emulated.

In this work, we present a novel approach to improve GANs performance by better preserving high-order statistics when using GANs to build an emulator of complex PDE-governed systems. We tested the statistical constraint with three relatively simple canonical systems, and discussed extensions that are applicable to more complex systems with the ultimate goal of leveraging existing high-fidelity simulation output to emulate unresolved processes, and provide reliable and accurate alternatives to closure models.

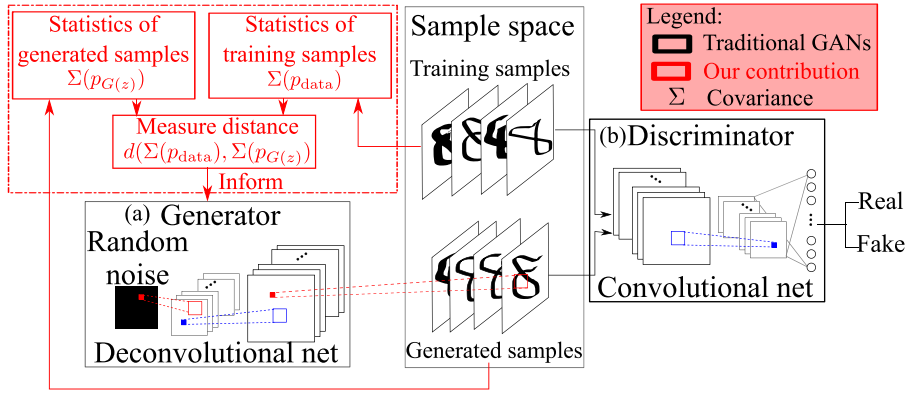


Fig. 1. The architecture of a statistics-informed generative adversarial network (GAN), including the architecture of a standard GAN (indicated by the black color) and the modification to help preserving statistics (indicated by the red color). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2. Methodology

2.1. Generative adversarial networks

Generative adversarial networks (GANs) have been introduced in [22] as a new technique for training machine learning models, originally in the context of computer image synthesis. The original formulation trains models in a purely unsupervised way; yet since their introduction many modifications have been proposed in the literature that make use of available labeled data in a semi-supervised way. The goal of GANs, as initially proposed, is to train a function G that samples from a complicated, *analytically-unknown* distribution of the data that the algorithm sees.

The main innovation of GANs is the formulation of the training procedure as a zero-sum game between two networks, a generator G and a discriminator D . In the standard setting, the generator receives input as an unstructured random noise vector z drawn from a simple distribution (such as an uniform or Gaussian), which it passes through a succession of deconvolutional layers and nonlinear transforms in a deterministic way and outputs a sample $x_{\text{fake}} = G(z)$. The role of the discriminator D is to act as a classifier, deciding if a sample x it receives is either real or generated by G . At optimality (Nash equilibrium in the game between G and D), the generator is provably able to produce “fake” samples x_{fake} that are implicitly drawn from the (unknown) data distribution that G seeks to emulate.

In this paper, G in Fig. 1a and D in Fig. 1c are deep deconvolutional neural network and convolutional neural network described by the weights vectors θ_G and θ_D , respectively. The architectural details follow the structure proposed in [31]. We train the model in the standard way of alternating between the two optimization problems:

$$\mathcal{L}_D = \mathbb{E}_{x \sim p_x} [\log D(x; \theta_D)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z; \theta_G); \theta_D))], \tag{1}$$

$$\mathcal{L}_G = \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z; \theta_G); \theta_D))] \tag{2}$$

where \mathbb{E} denotes the expectation and p_x represents the distribution of the training samples. p_z denotes the distribution from which the input of the generator G is sampled. Note that the training of standard GANs involves optimizations of \mathcal{L}_D and \mathcal{L}_G in Eqs. (1) and (2) iteratively. The weights coefficients θ_G are kept as constants when optimizing the discriminator loss \mathcal{L}_D , and vice versa. Conceptually, the training procedure corresponds to a minimax two-player game:

$$\min_{\theta_G} \max_{\theta_D} L(D, G) = \mathbb{E}_{x \sim p_x} [\log D(x; \theta_D)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z; \theta_G); \theta_D))] \tag{3}$$

where the ideal objective is to achieve Nash equilibrium, i.e., the discriminator cannot distinguish the generated samples from the training samples. However, the global optimal is usually difficult to achieve by using standard GANs. Therefore, it is unlikely that standard GANs can identify and preserve all the statistics of the training samples embedded in p_{data} . Those missing statistics are usually important to PDE-governed systems, e.g., the second order moments of the instantaneous velocity corresponds to the Reynolds stress in turbulence modeling. Therefore, we present an approach to better preserve these statistics when using GANs to emulate physical systems.

2.2. Constrained generative adversarial networks

The standard formulation above is purely data-driven, i.e., it utilizes no outside knowledge of the problem at hand but that which is encoded implicitly in the data that is used to train the model. As we argued above, constraining the model solution space to physically-feasible regions may yield certain benefits in terms of increased solution quality, decreased training time, and improved data efficiency. Constraints can be imposed in a variety of ways, among which:

- (i) *Hard constraints on network architecture.* Convolutional networks are by construction translational-invariant and ensure spatial locality of information propagation. It has long been recognized that convolutional filters can be interpreted as discretized differential operators in the finite difference form [32–34]. This fact has recently been exploited to train neural networks as surrogate models of differential equations [33] and to discover PDEs from data [34]. Such techniques may be used to constrain the space of feasible solutions of GANs by construction.
- (ii) *Explicit constraints on optimization.* Domain knowledge can be incorporated in a straightforward way through additional penalty terms added into the optimization loss function of GANs. This is in effect a form of regularization that can model many relevant constraints, including conservation laws and high-order statistics.

In this work, we focus on the second type of constraints discussed above and propose a constrained loss function $L_c(D, G)$ as follows:

$$L_c(D, G) = L(D, G) + \lambda d(\Sigma(p_{\text{data}}), \Sigma(p_{G(z)})) \quad (4)$$

where $\Sigma(p)$ denotes the covariance structure of a given distribution p , λ denotes the coefficient of the penalty term, and $d(\cdot, \cdot)$ represents a distance measure between two covariance structures. The distance d can be measured in Euclidean space by using Frobenius norm:

$$F(\Sigma_1, \Sigma_2) = \|\Sigma(p_{\text{data}}) - \Sigma(p_{G(z)})\|_F \quad (5)$$

or by using the symmetrized Kullback–Leibler divergence [35], i.e.,

$$KL_s(\Sigma_1, \Sigma_2) = \frac{1}{2} \text{tr} \left(\Sigma_1^{-1} \Sigma_2 + \Sigma_2^{-1} \Sigma_1 - 2\mathbf{I} \right), \quad (6)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, and \mathbf{I} represents the identity matrix. Note that the covariance structure must be positive semidefinite and thus is defined on a low-dimensional manifold within the Euclidean space. Therefore, an alternative is to use Riemannian distance

$$J(\Sigma_1, \Sigma_2) = \sqrt{\sum_{i=1}^n \ln^2 \delta_i(\Sigma_1^{-1} \Sigma_2)}$$

where δ_i denotes the i -th eigenvalue of a matrix, or by using the Jensen-Bregman LogDet (JBLD) divergence [36]:

$$J(\Sigma_1, \Sigma_2) = \log \left\| \frac{\Sigma_1 + \Sigma_2}{2} \right\| - \frac{1}{2} \|\Sigma_1 \Sigma_2\| \quad (7)$$

We have compared these different choices of the distance measure $d(\Sigma(p_{\text{data}}), \Sigma(p_{G(z)}))$ and concluded that the Frobenius norm serves as the best compromise in terms of the computational cost and the stability of training constrained GANs. The proposed statistics-informed GAN is implemented on the machine learning platform `TENSORFLOW` [37]. We used a deep convolutional GAN (DCGAN) [24] for most of the results presented below, but this is still referred to as GAN hereafter for simplicity.

In this work, both the covariance structures of generated and training data are estimated from samples. Specifically, $\Sigma(p_{G(z)})$ is estimated from generated samples at every iteration of training the network. Instead of implicitly estimating the probability distribution of training data $\Sigma(p_{\text{data}})$ as standard GANs, the penalty term in Eq. (4) explicitly estimates the second-order moment of the distributions and thus better constrains the difference between the distributions of the training data and the generated samples. However, the motivation of incorporating explicit statistical constraint is not to replace the duty of the discriminator. Instead, the purpose of such kind of constraint is to assist the training of GANs by constraining the searching of high-dimensional coefficients vector θ_G within a lower dimensional manifold that satisfies the covariance of training data. Once the covariance of generated samples gets similar to the training data, the penalty term has less effect and thus the training of GANs relies upon the discriminator more. As shown in a recent work [38] about Discriminator Rejection Sampling (DRS), the DRS methods employ a discriminator to characterize the distribution, and such a characterization can be better learnt implicitly instead of being imposed manually. Note that $\Sigma(p_{\text{data}})$ can also be estimated from the governing equation of the system, since the spatial correlation within the solutions of a PDE-governed system is related to the differential operators of the governing PDEs. More details about this alternative way of estimating the covariance structure of the training data can be found in a separate work [39].

Incorporating physical constraints (either explicitly or implicitly) into GANs may offer certain benefits, of which this paper highlights but several. We posit that appropriately imposing constraints may improve training stability and convergence properties via the regularization these constraints provide. Many hypotheses have been put forth about the source of the training instabilities observed in GANs (mode collapse in particular, as observed initially in [22,31]), including different ways in which to measure the distance between the generated and the real data distributions [40], penalties on the gradients around real data points [41], regularizers on the spectral radius of the Jacobian of the gradient vector [42], or dynamically choosing the metric to minimize in the objective function via an evolutionary strategy as in [43].

2.3. Lattice Boltzmann simulation of Rayleigh-Bénard convection

The physical system we investigated in this work is Rayleigh-Bénard convection (RBC), which are ultimately driven by buoyancy differences encountered at all spatial scales, ranging from small engineering devices to Earth sciences and even astrophysical phenomena. The RBC setup is a classical prototype for this class of flows. It consists of a slab of fluid, which is bounded by two horizontal surfaces.² It was observed from early on [44] that, when the lower wall is heated (while the upper wall is cooled), the dynamics of the system is especially interesting – a myriad of flow regimes appear, such as stationary or oscillating convection rolls, convection cells of different shapes, as well as turbulent flow. Although the detailed dynamics is strongly-dependent on the initial conditions, the selection of the different regimes is mostly dictated by the control-parameters, which for the RBC problem are the Rayleigh number ($Ra = \beta g H^3 (\Delta T) / \kappa \nu$ – the relative magnitude of the buoyancy velocity to the thermal velocity), the Prandtl number ($Pr = \nu / \kappa$ – the ratio of viscous diffusivity to thermal diffusivity), and the horizontal-to-vertical aspect ratio of the domain ($\Gamma = W/H$). In these definitions, β is the coefficient of thermal expansion, g is the gravitational acceleration, the spatial scales H and W are the height and width respectively, and $\Delta T \geq 0$ is the temperature difference between the lower and the upper walls.

The dataset we used for training a GAN to emulate the RBC problem consisted of simulation output, produced with a two-dimensional fluid dynamics model based on Lattice Boltzmann methods (LBM). The numerical scheme is described in [45] (see [46] for a more detailed description). To quantify how much more efficient the convective heat transfer is (compared to the conductive, diffusion-driven heat transfer), we can also evaluate the average Nusselt number $\langle Nu \rangle_0$ along the lower wall, which in our case (following the conventions in [45]) is defined as the dimensionless vertical heat flux:

$$\langle Nu \rangle_0 \approx \frac{1}{Nx \Delta T} \sum_{i=0}^{Nx-1} q_x(i, j)|_{y=0}, \quad (8)$$

where:

$$q_x = \nu T - \partial_y T. \quad (9)$$

To evaluate the vertical gradient of temperature, a second-order, three-points one-sided finite difference approximation may be used. The values of the control-parameters were $Pr = 0.71$ (for air) and $Ra = 2.5 \times 10^8$. For simplicity (and to facilitate comparison with some theoretical results), periodic boundary condition (BC) were used at the lateral (vertical) boundaries, with a relatively high aspect-ratio of $\Gamma = 7$, which minimizes unphysical artifacts due to periodicity itself.

The spatial resolution was $\{Nx, Ny\} = \{1792, 256\}$. It should be noted that the vertical resolution was the same as used in [47]. The choice of vertical resolution is based on the theoretical criteria from [48] to provide sufficient resolution within the near-wall boundary layers. The value for the pseudo Mach number (a model-specific parameter) was set to $Ma = 0.1$, to keep the artificial compressibility errors below 1%.

For non-dimensionalization of the equations, we used the free-fall velocity scale $U \equiv \sqrt{\beta g H (\Delta T)}$, from which a free-fall time-scale can be defined $\tau \equiv H/U$ (see e.g. [49]). The model was initially run for 400 free-fall times, of which the initial 320 free-fall times (necessary for spin-up from the rest state) were discarded. To verify that the data used for GAN training did in fact correspond to the final (turbulent) regime, this simulation was later extended until 5330 free-fall times. Two physical metrics of the flow (total kinetic energy $\varepsilon_{\text{total}}$ and Nusselt number Nu_0 close to the lower wall) are shown in Fig. 2. Both metrics are already stabilized at the start of the training window, showing that the data used for GAN training was already part of the final turbulent regime. To further support this hypothesis of statistical stationarity of the flow, the average value of the Nusselt number within the training window was found to be $\langle Nu_0 \rangle = 35.1137$, which is close to the values found by other authors – for example, [50] obtained $\langle Nu_0 \rangle = 34.1422$, for a similar setup, with a lower aspect-ratio of $\Gamma = 2$ (which artificially constrains the flow, considering that they also used periodic BC).

For an overview of the complexity of the physical patterns in the training dataset, we show in Fig. 3 the temperature field at three times. We note that the training data are limited to the two-dimensional case. Many physical and engineering systems are truly three-dimensional, displaying additional physical phenomena (such as vortex stretching), which are beyond the scope of the present study. Nonetheless, two-dimensional studies are still useful, as they still share much of the physics with the three-dimensional case (and therefore constitute a good test for the present GAN training with constraints). A future version of this study will treat the full, three-dimensional case.

3. Results

In this work, the performances of the standard GAN and the statistical constrained GAN are compared by investigating three different datasets. The first dataset is generated by sampling from a 2-D Gaussian process. The second and third datasets are from the Lattice Boltzmann simulations of 2-D Rayleigh-Bénard convection with different Rayleigh number and spatial resolution.

² For simplicity, in analytic or numerical studies the two bounding surfaces are sometimes assumed to extend infinitely in the horizontal directions.

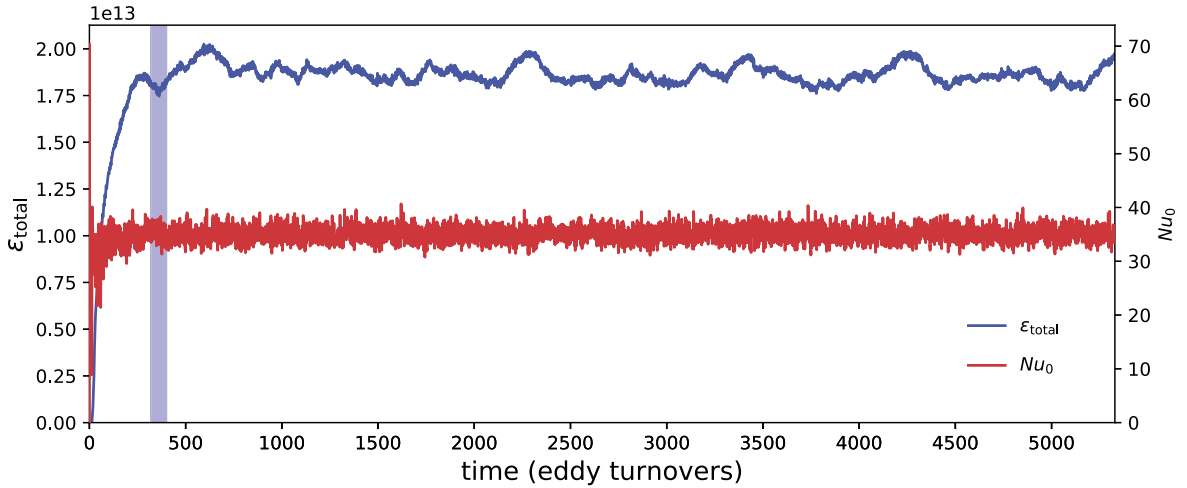


Fig. 2. Time-dependence of the total kinetic energy $\varepsilon_{\text{total}}$ (blue) and of the Nusselt number Nu_0 close to the lower wall (red) for the high-resolution RBC simulation.

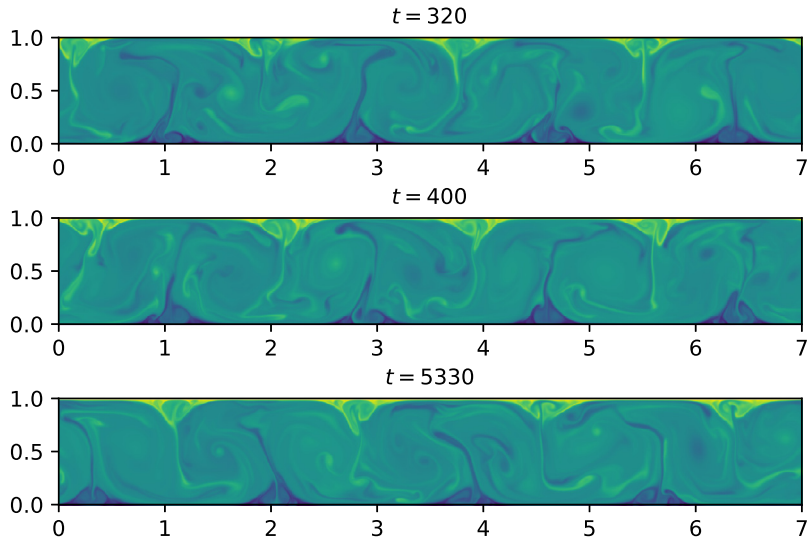


Fig. 3. Visualization of the temperature field for the high-resolution RBC simulation, at $t = 320$ free-fall times, $t = 400$ free-fall times and $t = 5330$ free-fall times (end of the extended simulation, produced for validation of the training dataset). The values of the control-parameters are $Pr = 0.71$ (for air) and $Ra = 2.5 \times 10^8$.

First, the results show that the constrained GAN better emulates the statistics of the training data by incorporating the constraint of covariance structure, indicating that the statistical constraint leads to the better converging towards the global minimum, where all statistics of training data can be captured by GANs. Similar performance may also be achieved by fine tuning the training process of the standard GANs. However, tuning a GAN largely relies on the experience of users and even may not be feasible for complex systems. Therefore, the constrained GAN is more stable in terms of training, making it a more practical tool for emulating complex systems.

Second, we show that the constrained GAN can achieve even higher quality of results at a significantly lower computational cost (up to 80% reduction of computational cost in model training) compared with the unconstrained model. In effect, the addition of the statistical constraint reduces the space of allowable solutions, forcing the training procedure (here, a standard stochastic gradient descent method) to explore only this reduced solution space. One could argue that, even if the model was trained by randomly selecting parameter configurations from the allowable set of solutions, the fact that the feasible set is now smaller will allow for a faster exploration on average, resulting in a reduced training time. We recognize that this is of course an experimental result on an idealized system, but these results are in line with the benefits expected from regularizing machine learning models.

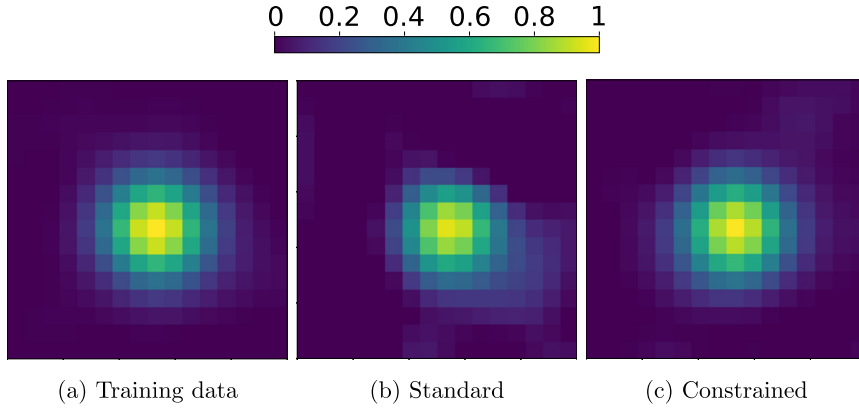


Fig. 4. Covariance with regard to the center point by using (a) training samples, (b) samples from standard GAN and (c) samples from constrained GAN. The statistical constrained GAN better captures the symmetry pattern of covariance from the training samples of the Gaussian process.

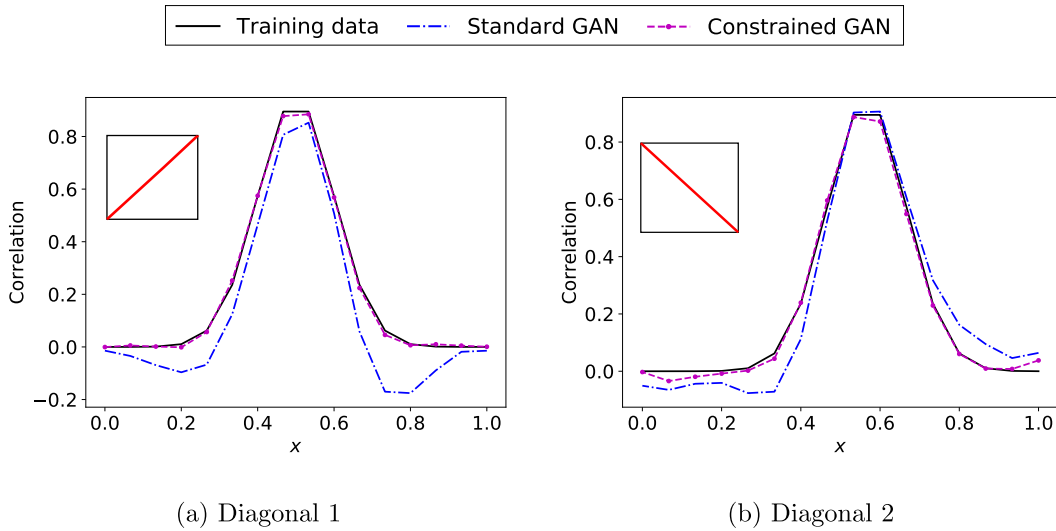


Fig. 5. Correlation profiles along (a) diagonal 1 and (b) diagonal 2. The red line within a square at the top-left part of each figure indicates the direction of the diagonal.

3.1. 2-D Gaussian process

We first compare the performances of standard GANs and the statistical constrained GANs by using the samples from 2-D Gaussian process as the training data. The 2-D Gaussian process is constructed in a square domain with the length of side $L = 1$. Square exponential kernel $K(\mathbf{x}, \mathbf{x}')$ in Eq. (10) is adopted to describe the covariance structure of the Gaussian process, and the standard deviation is set as $\sigma = 1$. The 2-D square domain is discretized with the resolution of 16×16 .

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\right) \tag{10}$$

The length scale is chosen as $\ell = 0.2L$, where L denotes the length of side of the square domain. The training dataset is obtained by sampling from the 2-D Gaussian process, and we acquired 10000 samples in total. These samples are provided to the discriminator of GANs with the label of 1 (true), and the objective is to use GANs to generate samples that capture the statistics of the training samples. The covariance of training data with regard to the center point is presented in Fig. 4a, in which a symmetrical pattern can be seen and the magnitude of covariance gradually decays from the center to the side.

The comparison of the covariance of the generated samples from GANs demonstrates the superior performance of the statistical constrained GAN as shown in Fig. 4. Specifically, it can be seen in Fig. 4b that the estimated covariance of the generated samples from the standard GAN shows a noticeable asymmetrical pattern. More quantitative comparison of correlation profiles along diagonals in Fig. 5 also confirms that the results from constrained GAN better capture the symmetrical pattern of the correlation field from training data. It indicates that the standard GAN converges to a local minimum and the statistics of the training data is not truthfully reproduced. On the contrary, the covariance of the generated samples from

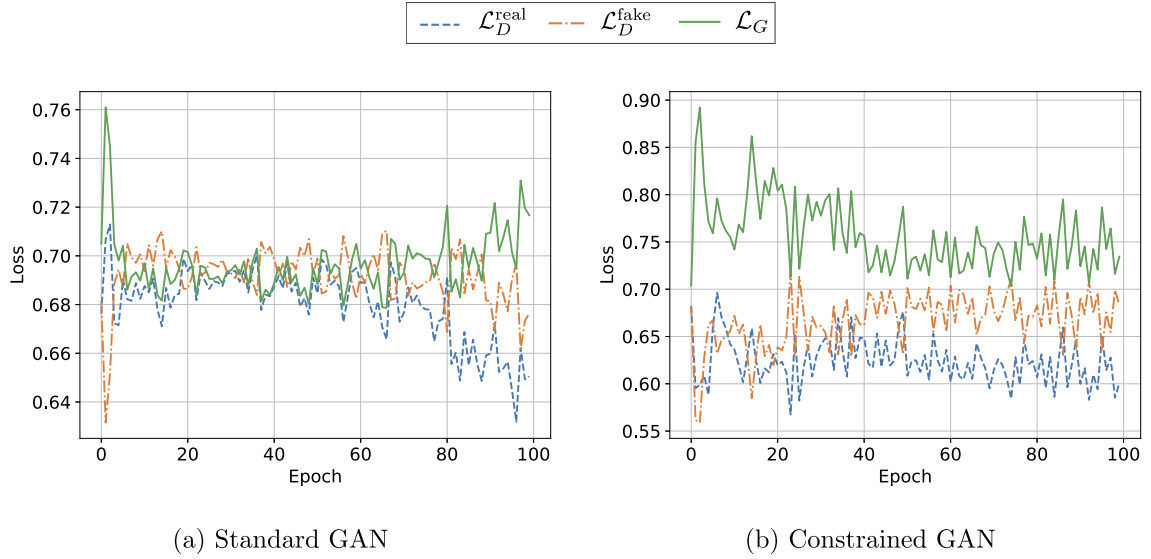


Fig. 6. Comparisons of loss functions between standard GAN and constrained GAN. $\mathcal{L}_D^{\text{real}}$ denotes the discriminator loss with training data as inputs, $\mathcal{L}_D^{\text{fake}}$ corresponds to the discriminator loss with generated samples as inputs, and \mathcal{L}_G represents the generator loss.

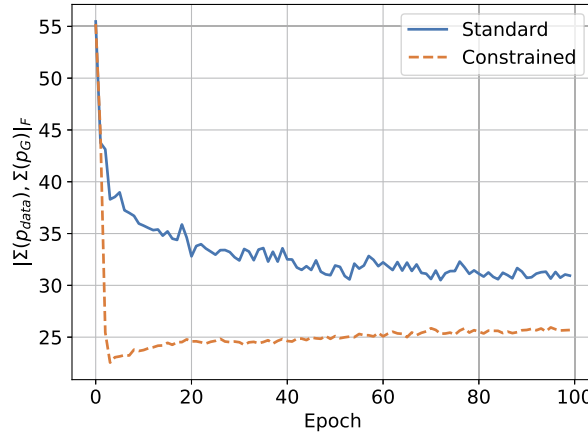


Fig. 7. Comparisons of Frobenius norm between the covariance structures of samples generated by standard GAN and constrained GAN.

the statistical constrained GAN shown in Fig. 4c demonstrates a better agreement with the training data. It indicates that the statistical constraint guides GANs to better converge toward global minimum.

As shown in Fig. 6, the comparison of different terms in the loss function of GAN demonstrates the effect of incorporating statistical constraint into generator. Compared to the results of standard GAN in Fig. 6a, the generator loss (\mathcal{L}_G) of the constrained GAN in Fig. 6b is slightly greater than the discriminator loss with generated samples as inputs ($\mathcal{L}_D^{\text{fake}}$) during the whole training process. It indicates that the optimization of the generator of constrained GAN is more effective, since the statistical constraint embeds the optimization of the generator within a lower-dimensional space. On the contrary, the optimization of the generator of standard GAN is less effective during the early stage of training, and the generator loss can be less than the discriminator loss with generated samples at some epochs. The effect of the statistical term on training is also investigated by observing the Frobenius norm between the covariance structures as shown in Fig. 7. It can be seen that the difference in the covariance structures drops much quicker by using constrained GAN, and the Frobenius norm of the result from constrained GAN is consistently smaller than the one from standard GAN during the whole training process.

The main purpose of introducing the statistical constraint can be interpreted in two ways. First, the constraint term serves as a regularization and reduces the loss function to a lower dimensional manifold, in which the convergence to global minimum would be easier to achieve. Second, the constraint term contributes a non-zero gradient when the training process get into local minima and thus the optimization is unlikely to stay at a local minimum. According to these two purposes, it is not necessary to define a precise metric of the difference between the distributions of the training data and the generated samples. Instead, an approximate distance metric should work well enough as long as it vanishes when two distributions are identical to each other. As shown in Eq. (4), the distance between two covariance matrices is defined in

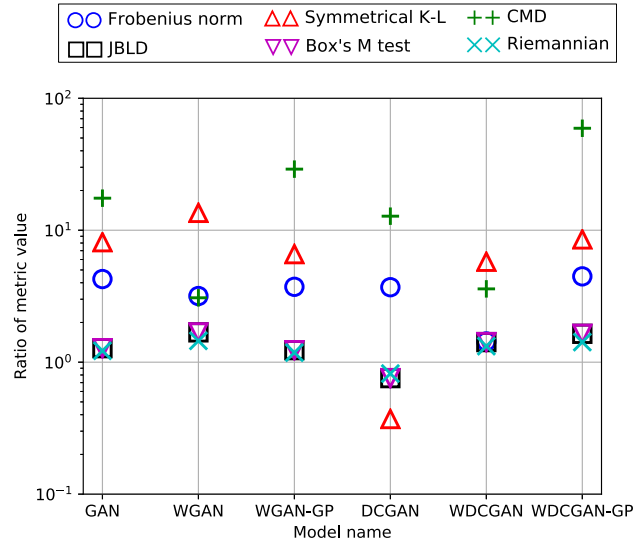


Fig. 8. Summary of the distance metric ratios of the covariance matrix from the generated samples of standard GAN and constrained GAN. Specifically, ratios are calculated between the distance metric of standard GAN and the one from constrained GAN. WGAN denotes *Wasserstein* GAN [25], and GP denotes gradient penalty as proposed in [40]. CMD denotes correlation matrix distance as defined in [51]. Note that WDCGAN represents deep convolutional GAN with loss function defined by *Wasserstein-1* metric.

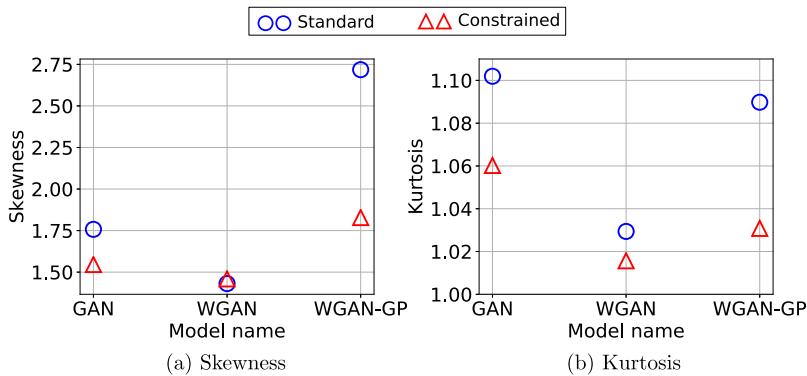


Fig. 9. High-order statistics of (a) skewness and (b) kurtosis provided by different types of GANs. The statistics-informed constraint leads to improvement of skewness and kurtosis for most types of GANs.

Euclidean space by using Frobenius norm. The main reason of using Frobenius norm is the relatively low computational cost, compared with other distance metrics, e.g., the K-L divergence and the Riemannian distance. After obtaining the generated samples, we evaluated different distance metrics to quantify the difference between the statistics of the training data and the generated samples. It can be seen in Fig. 8 that the statistical constraint term generally leads to smaller distance metrics when being implemented into different GANs architectures. Specifically, ratios between distance metrics of standard GAN and the ones from constrained GAN are presented in Fig. 8, and a value above 1 indicates the improvement by incorporating the statistical constraint. It confirms that the constrained GAN indeed better converges toward the global minimum, instead of merely reducing the specific distance metric adopted in the training.

We also investigated high-order statistics to illustrate the advantage of the statistical constrained GAN. The skewness and the kurtosis of the generated samples in Fig. 9 show that the high-order statistics of the samples have a better agreement with the training data when using the statistical constrained GAN, indicating that high-order statistics are also better captured by introducing the covariance constraint. Both the skewness and the kurtosis in Fig. 9 are normalized by the benchmark values of the multivariate Gaussian distribution, and thus the ideal values should be one. As shown in Fig. 9a, the larger skewness of generated samples from standard GAN indicates that the sample distribution is more asymmetrical, which has been confirmed by the visualization of covariance in Fig. 4.

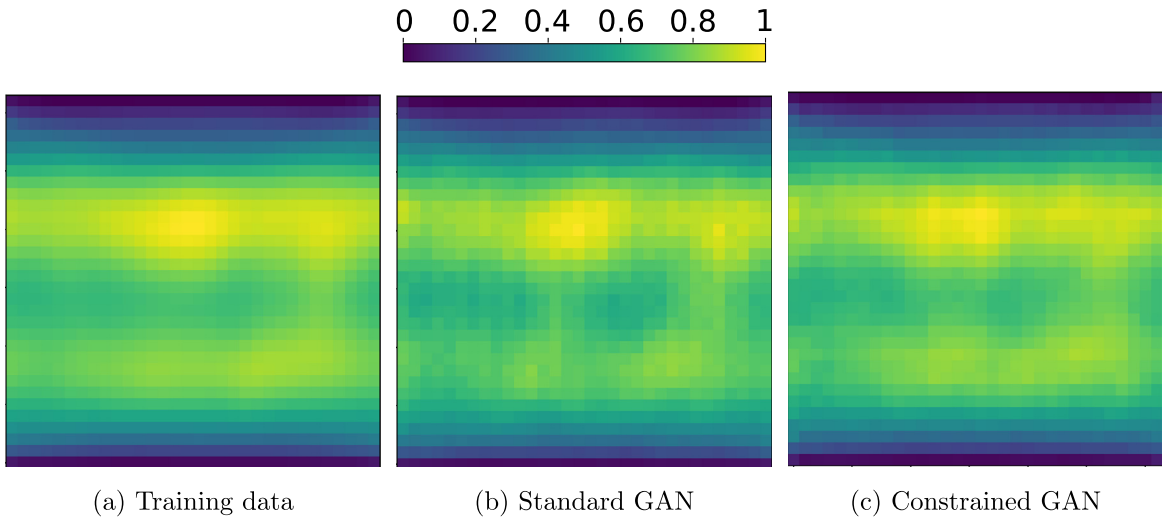


Fig. 10. Spatial distribution of turbulent kinetic energy of 2-D Rayleigh-Bénard convection from (a) training samples, (b) generated samples from GAN and (c) generated samples from statistics-informed GAN. The learning rate of the generator is set as 0.002.

3.2. Rayleigh-Bénard convection at a low Rayleigh number

We also studied the performance of the statistical constrained GAN by using training data from the simulations of Rayleigh-Bénard convection. The training data is obtained from a further simplified RBC system compared to the one described in Section 2.3, in order to first investigate a less chaotic RBC system (Rayleigh number $Ra = 10000$ and Prandtl number $Pr = 10$). Specifically, the simulation is performed in a two-dimensional domain of aspect ratio 1 with periodic boundary conditions on the left and right boundaries. The boundary conditions of the velocity field at the top and bottom boundaries are both no-slip walls. The top wall is at a low temperature and the bottom wall is at a high temperature. The training data corresponds to 10000 snapshots of the instantaneous flow field. The turbulent kinetic energy (TKE) of the training data is presented in Fig. 10. It can be seen that the TKE is relatively low near both the top and the bottom walls. In addition, higher TKE can be observed within two horizontal stripe-shape regions both above and below the center region. By carefully tuning the learning rate, it can be seen in Fig. 10 that the TKE of generated samples shows a good agreement with the training data by using either the standard GAN or the statistical constrained GAN. It is because the standard GAN is capable of preserving all the statistics of the training data if global minimum is achieved for the training process. Therefore, with a proper choice of training parameters, the performances of the standard GAN and the statistical constrained GAN should be comparable with each other, which has been confirmed in Fig. 10. It should be noted that the resolution (32×32) in Fig. 10 directly corresponds to the spatial mesh for the training data and the generated samples. Although Fig. 10c seems to indicate that the boundary layer thickness predicted by the constrained GAN is larger than the snapshot in the training data, the spatial resolution is not fine enough to warrant such a comparison, and the overall performance is comparable by using either standard GAN or constrained GAN in Fig. 10.

However, the optimal training is usually difficult to be achieved, especially for training a GAN to emulate more complex physical systems. In order to illustrate the robustness of training GANs, we adopt another learning rate $lr = 0.02$ and present the TKE of generated samples in Fig. 11. It can be seen in Fig. 11b that the result of standard GAN changes significantly, showing noticeable difference from the training data. Although the higher TKE regions generated by the standard GAN still locates above and below the horizontal center region, the regions with high TKE become less continuous and do not capture the pattern in Fig. 10. On the contrary, the results of statistical constrained GAN in Fig. 11b demonstrates less changes. We remark that the more noisy results in Fig. 11c compared with the one in Fig. 10c is mainly due to the larger learning rate, which introduces more noises during the training. According to the comparison shown in Fig. 11, the statistical constrained GAN is more robust with regard to the change of training parameters.

In order to illustrate the more robust training by using the statistical constrained GAN, we studied the training of GANs with different learning rate and penalty coefficient of the statistical constraint term. The comparison in Fig. 12 demonstrates that better performance can be achieved by introducing the statistical constraint term. Specifically, the mean squared error (MSE) of the TKE field against the number of epochs is presented. The MSE is defined between the TKE field calculated from generated samples of GANs and the one from training data. It can be seen in Fig. 12a that the performances are comparable at 100 training epochs by using different values of penalty coefficient λ . However, with the same number of epochs, the mean squared errors of the statistical constrained GAN ($\lambda > 0$) are generally smaller than the standard GAN ($\lambda = 0$). By using a larger learning rate as shown in Fig. 12b, it can be seen that the standard GAN has difficulty to converge. In terms of constrained GANs, the convergence looks similar to the standard GAN if a small penalty coefficient $\lambda = 0.03$ is adopted. On the other hand, a large penalty coefficient $\lambda = 3$ leads to smaller MSE after the first epoch, while the MSE

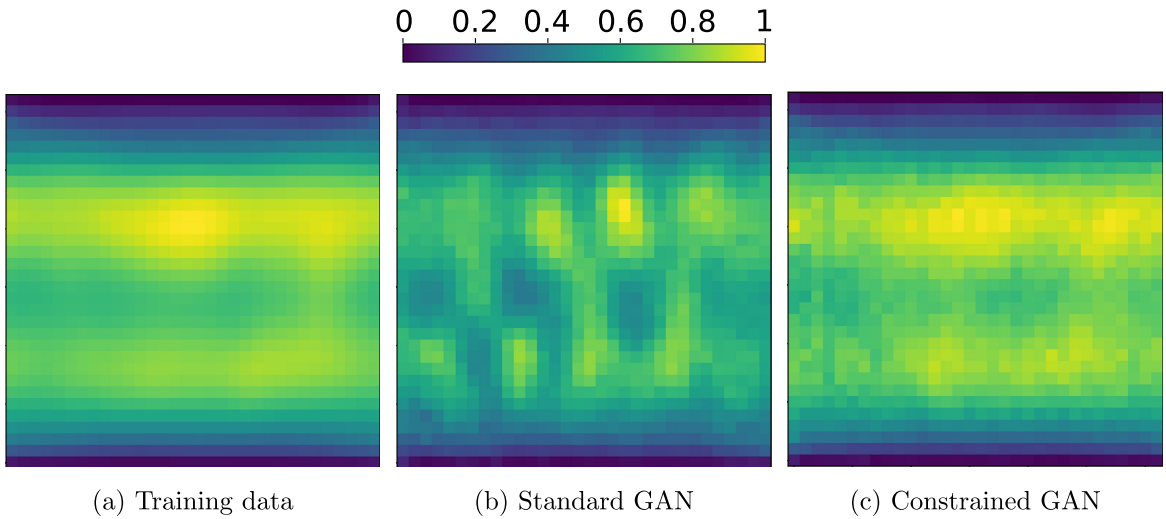


Fig. 11. Spatial distribution of turbulent kinetic energy of 2-D Rayleigh-Bénard convection from (a) training samples, (b) generated samples from GAN and (c) generated samples from physics-informed GAN. The learning rate of the generator is set as 0.02.

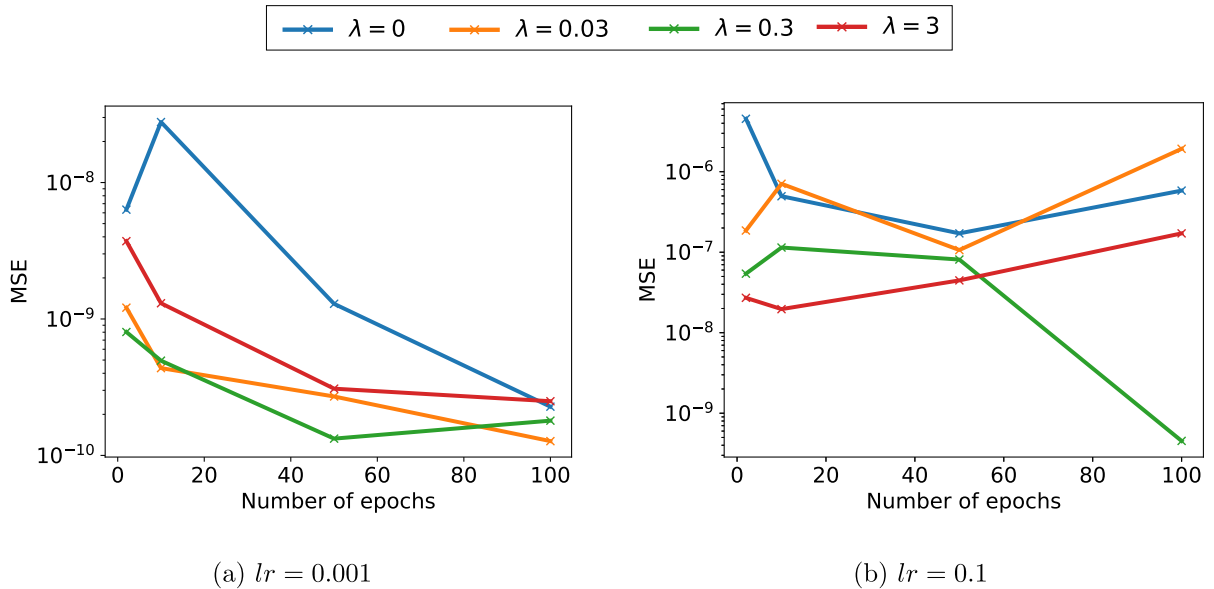


Fig. 12. Mean squared errors of the predicted turbulent kinetic energy of 2-D Rayleigh-Bénard convection with learning rates (a) $lr = 0.001$, (b) $lr = 0.1$. Four different coefficients λ of the constraint term are investigated. Note that $\lambda = 0$ corresponds to standard GANs.

could not be further reduced by continuing the training of GAN. These two extreme scenarios indicate possible issues of statistical-constrained GAN when the learning rate is large. A better convergence can be achieved even with a large learning rate ($\lambda = 0.1$) by properly choosing the penalty coefficient ($\lambda = 0.3$), demonstrating the merit of incorporating statistical constraint.

The comparison of spectrum of TKE in Fig. 13 also confirms that the statistical constrained GAN generates samples that better capture the pattern of the training data. Although such an improvement is marginal with learning rate $lr = 0.002$, it can still be seen in Fig. 13a that the spectrum from the statistical constrained GAN has a better agreement with the training data for relatively small wave numbers, e.g., between 2 to 5. At the region with large wave number, both the standard GAN and the statistical constrained GAN overpredict the level of energy spectrum, which is mainly because of the small scale noises in the generated data as shown in Fig. 10. If a less proper learning rate is chosen, it can be seen in Fig. 13b that the energy spectrum provided by the statistical constrained GAN shows a much better performance than the result of standard GAN, whose energy spectrum demonstrates disagreement with the training data across the whole range of wave numbers. The scaling of kinetic energy spectrum of the training data in Fig. 13 follows the trend of k^{-3} . It indicates that the flow is

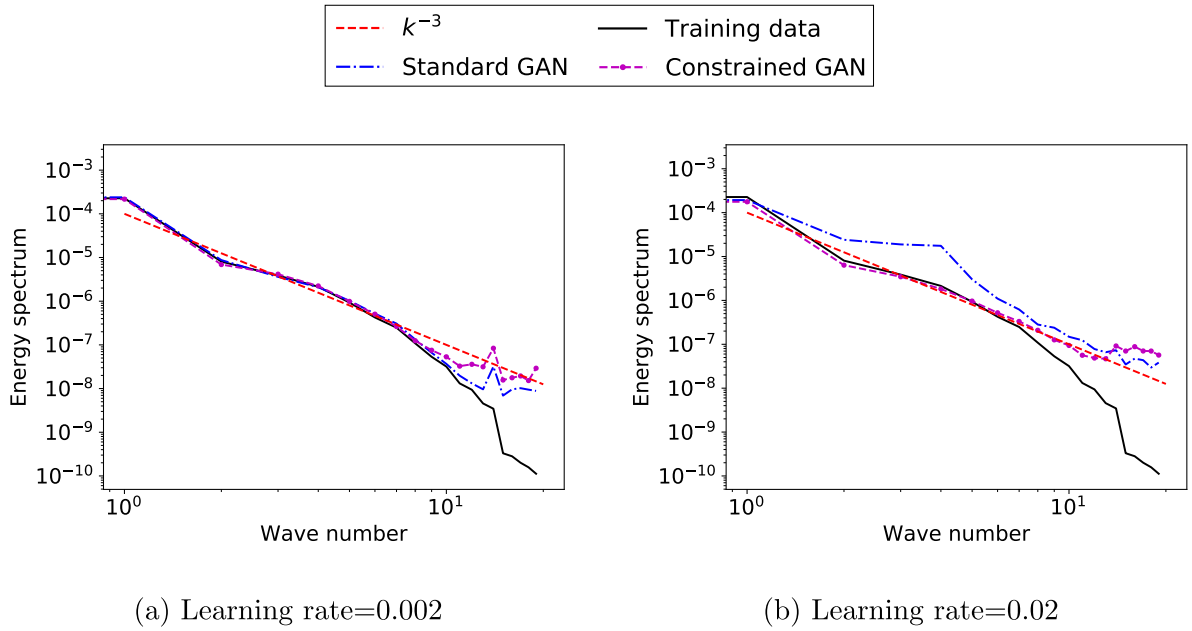


Fig. 13. Spectrum comparison of turbulent kinetic energy of 2-D Rayleigh-Bénard convection by using different learning rates.

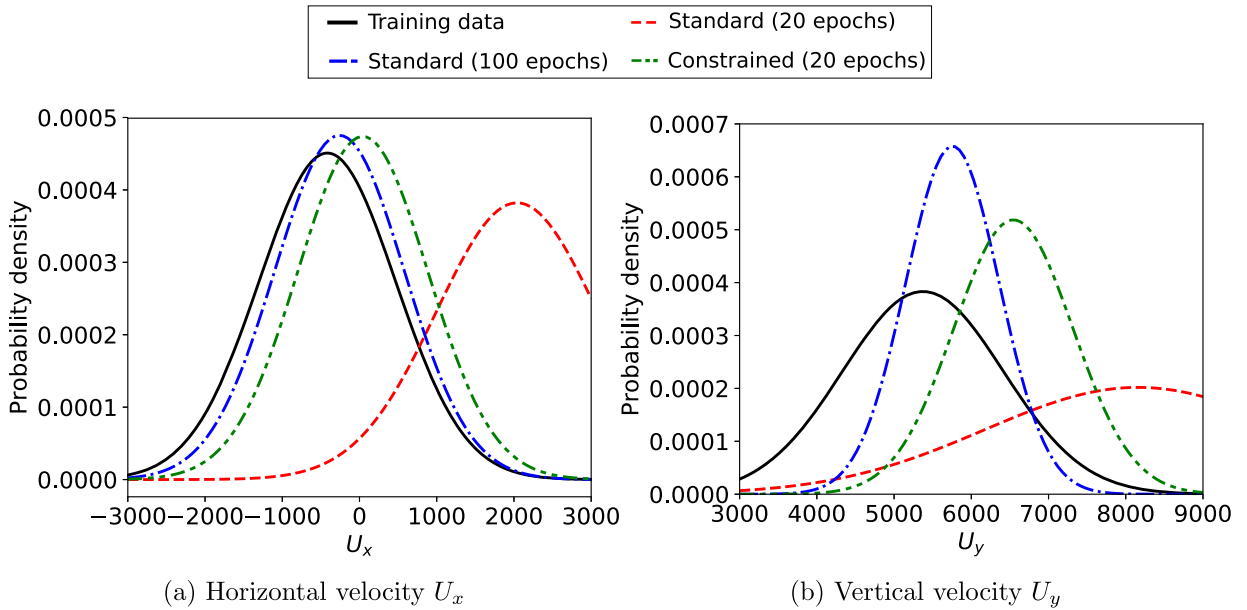


Fig. 14. The comparison of (a) horizontal velocity and (b) vertical velocity distribution of high Rayleigh number 2-D Rayleigh-Bénard convection at the center point of the training domain.

not in the fully turbulent regime with the simulation settings of this case, motivating us to further investigate with a more challenging and realistic case of Rayleigh-Bénard convection as presented in Sec. 3.3.

3.3. Rayleigh-Bénard convection at a high Rayleigh number

We further studied another dataset of Rayleigh-Bénard convection simulation at a higher Rayleigh number. More details about the Lattice Boltzmann simulation can be found in Section 2.3. As shown in Figs. 14 and 15, velocity distributions at two typical points confirm that the constrained GAN can better emulate the training data with the same training epochs. Specifically, the results at center point of the training domain is presented in Fig. 14, and the results at the upper right 1/4 point (referred to as *corner* point) along the diagonal of training domain are presented in Fig. 15. It can be seen in Fig. 14

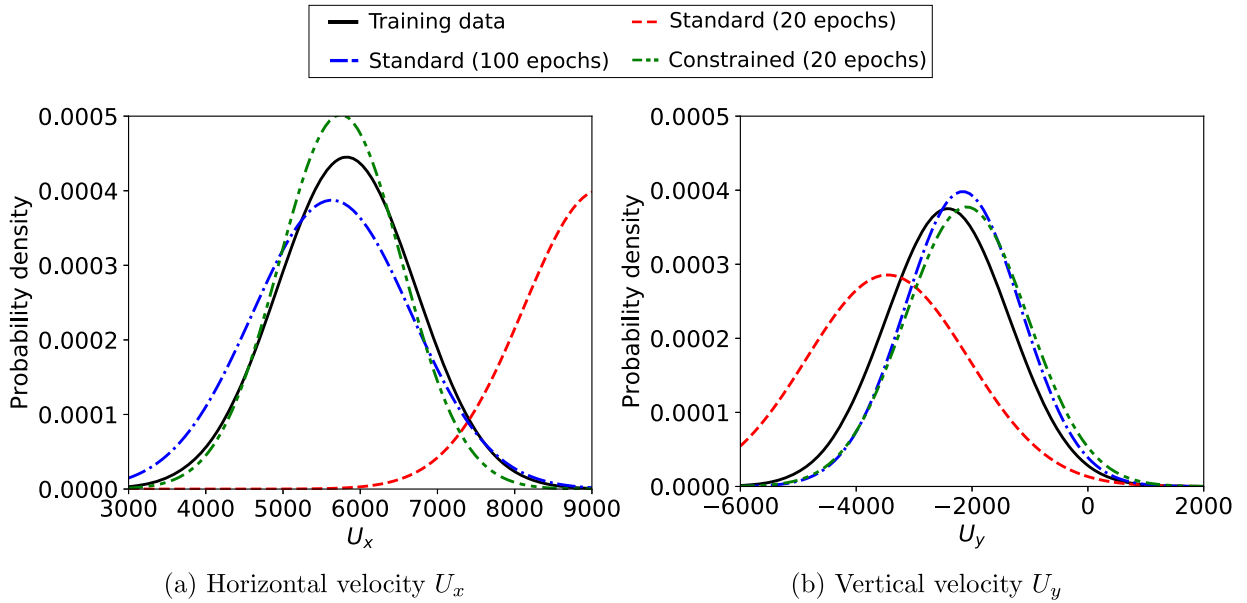


Fig. 15. The comparison of (a) horizontal velocity and (b) vertical velocity distribution of high Rayleigh number 2-D Rayleigh-Bénard convection at the corner point of the training domain.

Table 1
Wasserstein distance between training distribution and GAN emulated velocity distribution.

	Wasserstein distance			
	Center U_x	Center U_y	Corner U_x	Corner U_y
Standard GAN (20 epochs)	2455	2795	3287	1070
Standard GAN (100 epochs)	167	491	200	265
Constrained GAN (20 epochs)	463	1171	114	323

that the results of standard GAN at 20 epochs are noticeably biased. Similar biased results of the standard GAN at 20 epochs can also be observed in the results at the corner point as shown in Fig. 15. With 100 epochs, the results of standard GAN demonstrate much better agreement with the training data. On the other hand, the distributions of velocity at these two points from constrained GAN with only 20 epochs are much better than the results of standard GAN with the same training epochs. In addition, the results from the constrained GAN are even comparable to the results from standard GAN with 100 epochs, indicating that the time-averaged velocity field and the TKE of the training data are reasonably captured by using constrained GAN with less computational cost. More quantitative comparisons between distributions in Figs. 14 and 15 are presented in Table 1 by using Wasserstein distance. More detailed comparison of the velocity distributions is presented in Appendix A.

The comparison of time-averaged velocity field in Fig. 16 shows that the standard GAN at 20 epochs noticeably overestimates the velocity magnitude across the whole domain as shown in Fig. 16c, even though the circular flow pattern is qualitatively emulated. With 100 epochs, the time-averaged velocity provided by the standard GAN is getting more similar to the training data in large scale structure. However, there are unphysical small structures existing in Fig. 16d, indicating that the spatial correlation of time-averaged velocity is not well captured by using the standard GAN. Compared to the results of the standard GAN, the result of constrained GAN in Fig. 16b better emulates the training data with only 20 epochs. Although there are still some noises in small scale in Fig. 16b, the noise level is much less than the result of standard GAN with 100 epochs.

The comparison of the TKE fields as shown in Fig. 17 also demonstrates the better performance of the constrained GAN. The result in Fig. 17d from the standard GAN involves 100 epochs, while the result in Fig. 17b from the statistical constrained GAN only involves 20 epochs. With less training epochs, the result of the statistical constrained GAN still shows better agreement with the training data. Specifically, the spatial pattern of the training data around the left bottom corner region is much better captured by the constrained GAN. Also, improvements can be observed at other regions across the whole domain as shown in Fig. 17. Although current results of the constrained GAN clearly out-performs the results of standard GANs, it should be emphasized that the results of standard GANs can potentially be improved with a proper tuning of the training, similar to the results presented in Fig. 10. However, such a proper tuning of standard GANs is difficult to achieve for real applications with complex distributions. The results in Figs. 16 and 17 together demonstrate

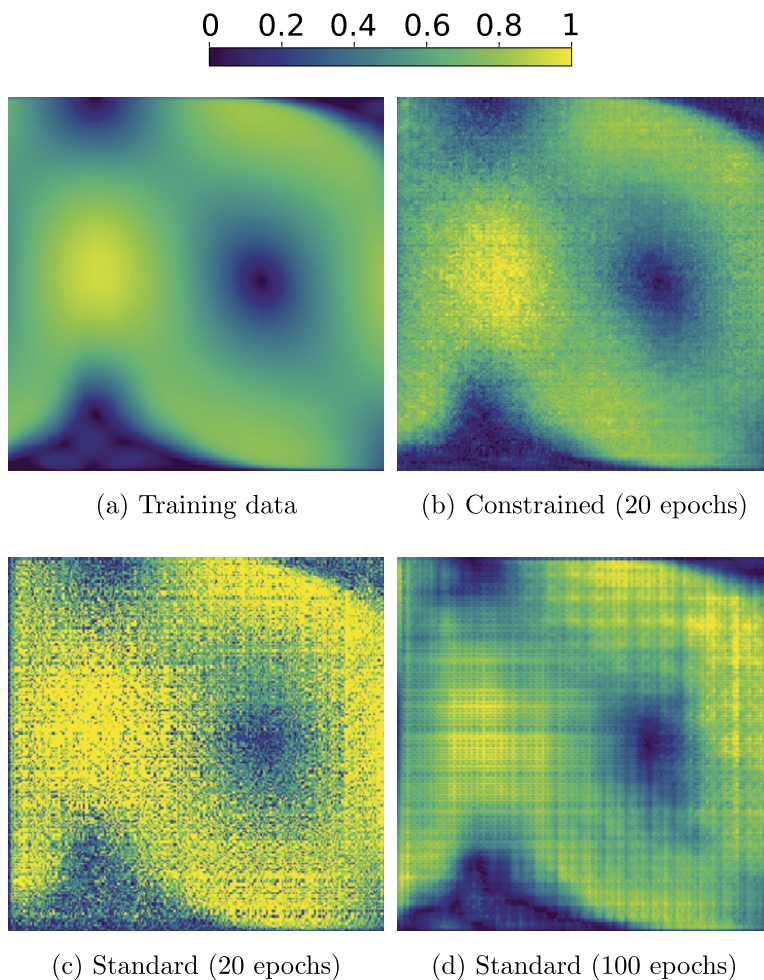


Fig. 16. The comparison of time-averaged velocity magnitude of 2-D Rayleigh-Bénard convection at high Rayleigh number from (a) training data, (b) constrained GAN at 20 epochs, (c) standard GAN at 20 epochs and (d) standard GAN at 100 epochs.

that the constrained GAN is a more practical tool for preserving important statistics (e.g., mean and second-order moment) when emulating complex systems.

We also studied the spectrum of TKE in Fig. 18. Before calculating the energy spectrum, the TKE fields are normalized by the maximum value of TKE in the training data. It can be seen that the result of standard GAN at 20 epochs overestimates energy spectrum across the whole range of wave numbers. With 100 training epochs, the result of the standard GAN better captures the energy spectrum for relatively small wave numbers (wave number is less than 5), but still shows noticeable difference from training data for large wave numbers (small scale structure in space). In addition, the noise level increases with more training epochs by using the standard GAN, indicating that asymptotically improvement of performance may not be achieved with more training epochs. On the other hand, the result of the constrained GAN at 20 epochs successfully captured the energy spectrum of the training data for most wave numbers, with only an exception of very large wave numbers ($\sim 10^2$). In terms of the training data at large wave numbers, an abrupt drop can be observed in Fig. 18. The main reason is that the training data has been mapped to the uniform mesh with a resolution of 256×256 (same as the generated samples from GANs). Therefore, sub-grid energy under this resolution has been truncated for the results presented in Fig. 18.

We would like to point out that the current work only trains on velocity fields of Rayleigh-Bénard convection and thus can only generate velocity fields. As a preliminary exploration, this case is sufficient to demonstrate the merits of incorporating statistical constraints into the training of GANs. It would be ideal to train on both velocity and temperature fields and to predict thermal-related quantities and the turbulent transport of temperature field as well. Although such a coupled training is not pursued in this work, it would be a natural and straightforward extension of the current work in order to make the generated samples of Rayleigh-Bénard convection more physical. Some candidate quantities (e.g., thermal boundary layer thickness) for the future study can be found in [52,53]. Considering that the current work only focuses on the training and generation of velocity field, Nusselt number of the generated samples is not investigated in this

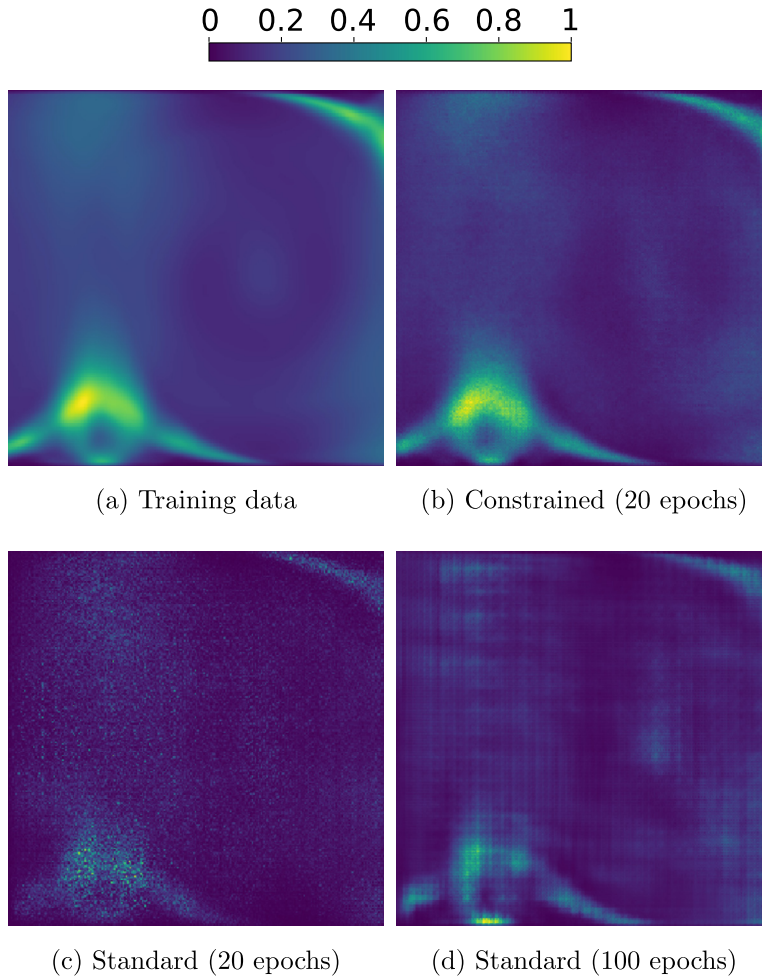


Fig. 17. The comparison of turbulent kinetic energy of 2-D Rayleigh-Bénard convection at high Rayleigh number from (a) training data, (b) constrained GAN at 20 epochs, (c) standard GAN at 20 epochs and (d) standard GAN at 100 epochs.

work. However, it is worthwhile to point out that the Nusselt number is an important dimensionless number to characterize Rayleigh-Bénard convection, and it should be a primary diagnostic quantity when velocity and temperature fields are trained and generated in a coupled manner.

In addition, the main purpose of statistical constraints is to assist the training of GANs better achieving global optimal solution by constraining the generator with known statistics in the training data. Although the predictive capability is not the focus of this work, this work can be extended to conditional GANs [30] in order to incorporate the trained GANs as closures of real applications with steady time-averaged velocity field. In terms of applications with unsteady time-averaged velocity field, further extension to learn the temporal correlation is needed and related studies can be found in [27]. Physical constraints [30,28] such as conservation laws can also be incorporated into the trained GANs to ensure obtaining physically correct closure models when applying to real applications of closure problems. There are also works [29] incorporating constraints to enhance interpolation and extrapolation capabilities of GANs. Considering that the above works focus on specific types of constraints, a comprehensive comparison is still lacking among these works and the proposed statistical constrained GAN as well, which deserves more investigation to further enhance the capability of GANs for science and engineering applications.

4. Conclusion

Recently, generative adversarial networks (GANs) have shown great potential of emulating and even predicting the solutions of PDEs-governed physical systems by training on some high-fidelity simulation data. However, it is known that GANs are much more difficult to train than regular neural networks, preventing its application to more complex and realistic physical systems. In this work, we introduce a statistical constrained approach to improve the robustness of training GANs. Specifically, the difference between the covariance structures of the generated samples and the training data is quan-

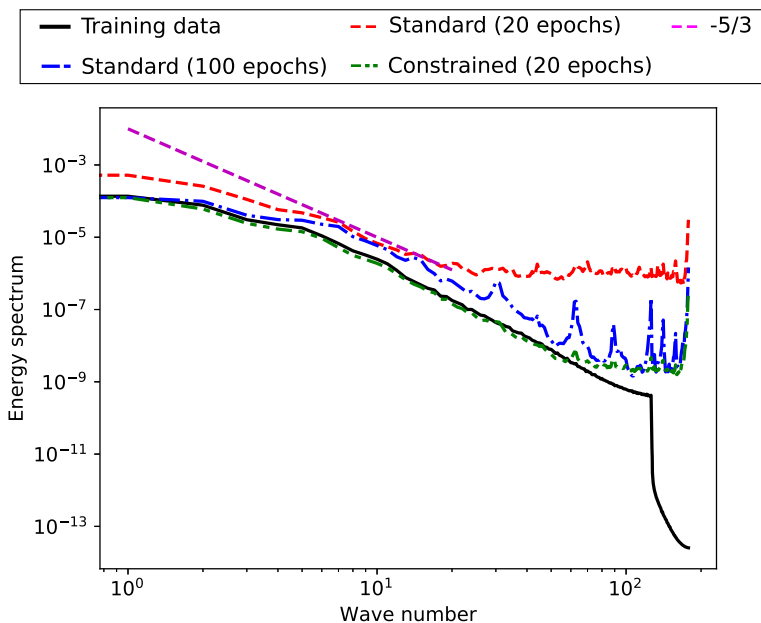


Fig. 18. Comparison of turbulent kinetic energy spectrum of 2-D Rayleigh-Bénard convection with a high Rayleigh number.

tified and incorporated into the original loss function of the generator in GANs as a penalty term. The performance of the constrained GAN is evaluated against the standard GANs by studying the Gaussian random field and the Rayleigh-Bénard convection. The results demonstrated that the proposed statistical regularization leads to more robust training than standard GANs. Even though similar quality of results can be achieved by carefully tuning the hyper-parameters of standard GANs, the constrained GAN significantly reduced (by up to 80%) training cost to reach the solution with similar or even better quality. With the growth of high-fidelity simulation databases of physical systems, this work has a great potential of being an alternative to the explicit modeling of closures or parameterizations for unresolved physics. By better preserving high-order statistics from existing high-fidelity simulations, this work potentially leads to a class of promising application of GANs in simulating chaotic physical systems, e.g., turbulence or Earth's climate.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Appendix A. More results of velocity distribution

The velocity distributions have been compared in Figs. 14 and 15 by fitting Gaussian distributions. However, the actual distributions show non-Gaussian pattern, and thus more detailed comparison is presented in Figs. A.19 and A.20 by studying histograms of velocity. Specifically, both horizontal velocity U_x and vertical velocity U_y at the corner point of the training domain are investigated. It can be seen in Fig. A.19 that the velocity histogram of standard GAN at 20 epochs demonstrate significant difference with the training data. Although the mean and the scattering of the histogram show better agreement with the training data for the standard GAN at 100 epochs, the high density peak is still not well captured. On the contrary, the histogram from constrained GAN at 20 epochs shows even better agreement with the training data. Similarly, it can be seen in Fig. A.20 that the histogram of vertical velocity U_y from constrained GAN at 20 epochs also outperform the results from standard GAN at 100 epochs, in terms of capturing the non-Gaussian pattern of the velocity distribution. In addition to the non-Gaussian pattern of the velocity distribution, it is well known that distributions of velocity increment also show non-Gaussian structures for turbulent flows. Although such kind of non-Gaussian structures is not investigated in this work, references about distributions of velocity increment can be found in [54,55].

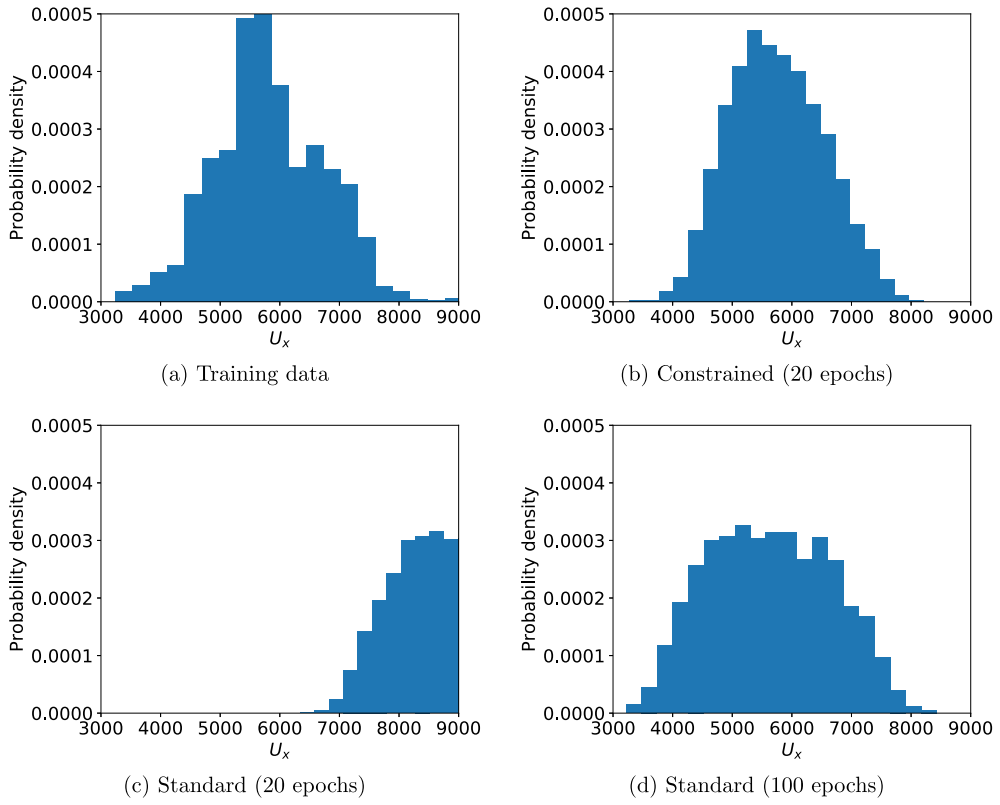


Fig. A.19. Histograms of horizontal velocity U_x at the corner point of the training domain.

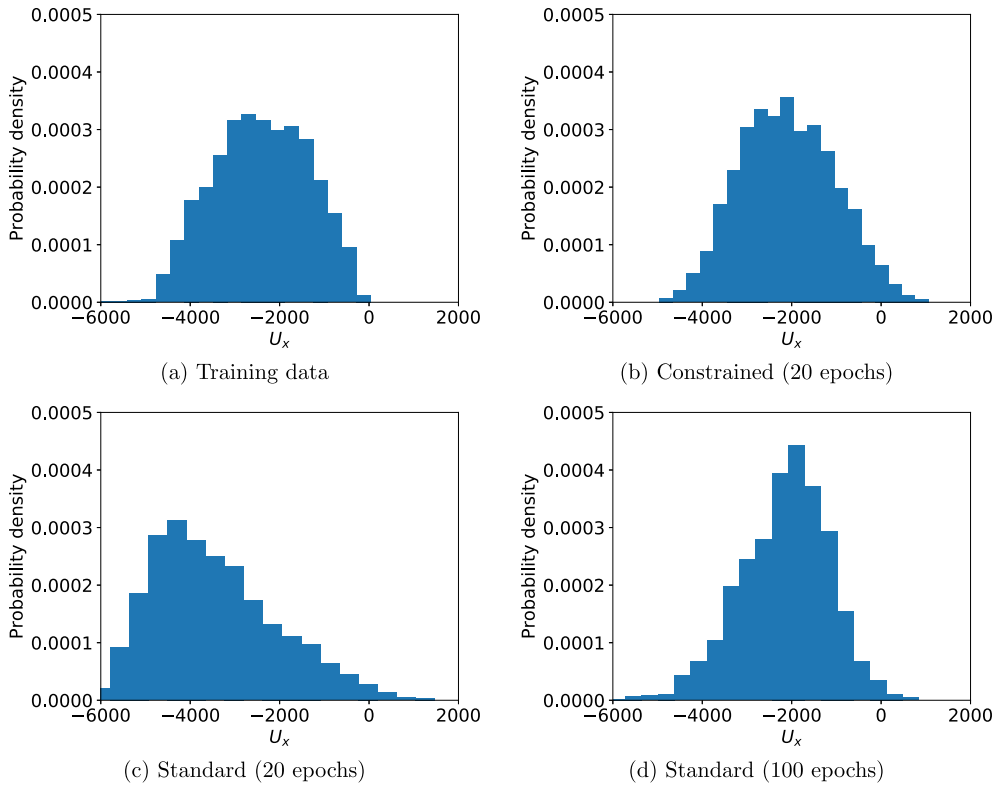


Fig. A.20. Histograms of vertical velocity U_y at the corner point of the training domain.

Table B.2

Detailed network architectures the generator and discriminator for the Gaussian process case. Acronyms used: C, number of channels; KS, kernel size; BN, batch normalization.

	Generator	Discriminator
Layer 1	Deconvolutional, C:32, KS: 4×4 , BN, LeakyReLU	Convolutional, C:16, KS: 5×5 , BN, LeakyReLU
Layer 2	Deconvolutional, C:16, KS: 5×5 , BN, LeakyReLU	Convolutional, C:32, KS: 5×5 , BN, LeakyReLU
Layer 3	Deconvolutional, C:1, KS: 5×5	Convolutional, C:1, KS: 4×4

Table B.3

Detailed network architectures the generator and discriminator for the Rayleigh-Bénard convection case at a low Rayleigh number. Acronyms used: C, number of channels; KS, kernel size; BN, batch normalization.

	Generator	Discriminator
Layer 1	Deconvolutional, C:16, KS: 8×8 , BN, LeakyReLU	Convolutional, C:16, KS: 5×5 , BN, LeakyReLU
Layer 2	Deconvolutional, C:8, KS: 5×5 , BN, LeakyReLU	Convolutional, C:32, KS: 5×5 , BN, LeakyReLU
Layer 3	Deconvolutional, C:1, KS: 5×5	Convolutional, C:1, KS: 8×8

Table B.4

Detailed network architectures the generator and discriminator for the Rayleigh-Bénard convection case at a high Rayleigh number. Acronyms used: C, number of channels; KS, kernel size; BN, batch normalization.

	Generator	Discriminator
Layer 1	Deconvolutional, C:16, KS: 8×8 , BN, LeakyReLU	Convolutional, C:16, KS: 5×5 , BN, LeakyReLU
Layer 2	Deconvolutional, C:16, KS: 5×5 , BN, LeakyReLU	Convolutional, C:16, KS: 5×5 , BN, LeakyReLU
Layer 3	Deconvolutional, C:16, KS: 5×5	Convolutional, C:16, KS: 5×5
Layer 4	Deconvolutional, C:1, KS: 5×5	Convolutional, C:1, KS: 8×8

Appendix B. Detailed settings of GANs

The detailed settings of GANs for different cases are summarized in Tables B.2 to B.4. All neural network architectures in this work are implemented on the machine learning platform `TensorFlow` [37]. The current implementation has no special treatment for different boundary conditions. Special treatment of boundaries is needed if GANs are used for tasks in which the inputs of generator are fields (e.g., for super-resolution of physical fields), and conditional GANs are required to take additional physical fields or boundary conditions as inputs. In these applications, it is important to design padding differently for different boundary conditions.

References

- [1] J.-X. Wang, J.-L. Wu, H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, *Phys. Rev. Fluids* 2 (3) (2017) 034603.
- [2] J.-L. Wu, H. Xiao, E. Paterson, Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework, *Phys. Rev. Fluids* 3 (7) (2018) 074602.
- [3] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *J. Fluid Mech.* 807 (2016) 155–166.

- [4] N.D. Brenowitz, C.S. Bretherton, Prognostic validation of a neural network unified physics parameterization, *Geophys. Res. Lett.* 45 (12) (2018) 6289–6298.
- [5] S. Rasp, M.S. Pritchard, P. Gentine, Deep learning to represent subgrid processes in climate models, *Proc. Natl. Acad. Sci. USA* 115 (39) (2018) 9684–9689.
- [6] D.J. Gagne, A. McGovern, M. Xue, Machine learning enhancement of storm-scale ensemble probabilistic quantitative precipitation forecasts, *Weather Forecast.* 29 (4) (2014) 1024–1043.
- [7] P.D. Dueben, P. Bauer, Challenges and design choices for global weather and climate models based on machine learning, *Geosci. Model Dev.* 11 (10) (2018) 3999–4009.
- [8] T. Schneider, S. Lan, A. Stuart, J. Teixeira, Earth system modeling 2.0: a blueprint for models that learn from observations and targeted high-resolution simulations, *Geophys. Res. Lett.* 44 (24) (2017).
- [9] Y. Liu, E. Racah, J. Correa, A. Khosrowshahi, D. Lavers, K. Kunkel, M. Wehner, W. Collins, et al., Application of deep convolutional neural networks for detecting extreme weather in climate datasets, arXiv preprint arXiv:1605.01156.
- [10] E. Racah, C. Beckham, T. Maharaj, S.E. Kahou, M. Prabhat, C. Pal, ExtremeWeather: a large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events, in: *Advances in Neural Information Processing Systems*, 2017, pp. 3402–3413.
- [11] M. Prabhat, J. Biard, S. Ganguly, S. Ames, K. Kashinath, S. Kim, S. Kahou, T. Maharaj, C. Beckham, T. O'Brien, et al., ClimateNet: a machine learning dataset for climate science research, in: *AGU Fall Meeting Abstracts*, 2017.
- [12] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436.
- [13] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, P. Riley, Tensor field networks: rotation- and translation-equivariant neural networks for 3D point clouds, arXiv preprint arXiv:1802.08219.
- [14] A. Karpayne, G. Atluri, J.H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar, Theory-guided data science: a new paradigm for scientific discovery from data, *IEEE Trans. Knowl. Data Eng.* 29 (10) (2017) 2318–2331.
- [15] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [16] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: a Navier-Stokes informed deep learning framework for assimilating flow visualization data, arXiv preprint arXiv:1808.04327.
- [17] C. Ma, J. Wang, et al., Model reduction with memory and the machine learning of dynamical systems, arXiv preprint arXiv:1808.04258.
- [18] B. Lusch, J.N. Kutz, S.L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, *Nat. Commun.* 9 (1) (2018) 4950.
- [19] R. King, P. Graf, M. Chertkov, From deep to physics-informed learning of turbulence: diagnostics, *Bull. Am. Phys. Soc. USA* 62 (2017).
- [20] R. King, O. Hennigh, A. Mohan, M. Chertkov, From deep to physics-informed learning of turbulence: diagnostics, in: *Workshop on Modeling and Decision-Making in the Spatiotemporal Domain, NIPS 2018*, 2018, arXiv preprint arXiv:1810.07785.
- [21] M. Mustafa, D. Bard, W. Bhimji, Z. Lukić, R. Al-Rfou, J.M. Kratochvil, CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks, arXiv preprint arXiv:1706.02390.
- [22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [23] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised pixel-level domain adaptation with generative adversarial networks, in: *The IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017, pp. 3722–3731.
- [24] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434.
- [25] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: *International Conference on Machine Learning*, 2017, pp. 214–223.
- [26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, in: *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.
- [27] Y. Xie, E. Franz, M. Chu, N. Thurey, tempoGAN: a temporally coherent, volumetric GAN for super-resolution fluid flow, *ACM Trans. Graph.* 37 (4) (2018) 95.
- [28] L. Yang, D. Zhang, G.E. Karniadakis, Physics-informed generative adversarial networks for stochastic differential equations, arXiv preprint arXiv:1811.02033.
- [29] P. Stinis, T. Hagge, A.M. Tartakovsky, E. Yeung, Enforcing constraints for interpolation and extrapolation in Generative Adversarial Networks, *J. Comput. Phys.* 397 (2019) 108844.
- [30] Y. Zeng, J.-L. Wu, H. Xiao, Enforcing deterministic constraints on generative adversarial networks for emulating physical systems, preprint arXiv:1911.06671.
- [31] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, *CoRR*, arXiv:1511.06434, <http://arxiv.org/abs/1511.06434>.
- [32] Y. Lu, A. Zhong, Q. Li, B. Dong, Beyond finite layer neural networks: bridging deep architectures and numerical differential equations, in: J. Dy, A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, PMLR, Stockholm, Sweden, in: *Proceedings of Machine Learning Research*, vol. 80, 2018, pp. 3282–3291, <http://proceedings.mlr.press/v80/lu18d.html>.
- [33] O. Hennigh, Lat-net: compressing lattice Boltzmann flow simulations using deep neural networks, arXiv preprint arXiv:1705.09036.
- [34] Z. Long, Y. Lu, X. Ma, B. Dong, PDE-net: learning PDEs from data, in: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2018.
- [35] J. Weickert, H. Hagen, *Visualization and Processing of Tensor Fields*, Springer Science & Business Media, 2005.
- [36] A. Cherian, S. Sra, A. Banerjee, N. Papanikolopoulos, Efficient similarity search for covariance matrices via the Jensen-Bregman LogDet divergence, in: *2011 IEEE International Conference on Computer Vision, ICCV*, IEEE, 2011, pp. 2399–2406.
- [37] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., TensorFlow: a system for large-scale machine learning, in: *OSDI*, vol. 16, 2016, pp. 265–283.
- [38] S. Azadi, C. Olsson, T. Darrell, I. Goodfellow, A. Odena, Discriminator rejection sampling, arXiv preprint arXiv:1810.06758.
- [39] J.-L. Wu, C. Michelen, H. Xiao, Physics-informed covariance kernel for model-form uncertainty quantification with application to turbulent flows, *Comput. Fluids* (2019) 104292.
- [40] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of Wasserstein GANs, in: *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.
- [41] N. Kodali, J.D. Abernethy, J. Hays, Z. Kira, How to train your DRAGAN, *CoRR*, arXiv:1705.07215, <http://arxiv.org/abs/1705.07215>.
- [42] L.M. Mescheder, S. Nowozin, A. Geiger, The numerics of GANs, *CoRR*, arXiv:1705.10461, <http://arxiv.org/abs/1705.10461>.
- [43] C. Wang, C. Xu, X. Yao, D. Tao, Evolutionary generative adversarial networks, *CoRR*, arXiv:1803.00657, <http://arxiv.org/abs/1803.00657>.
- [44] L. Rayleigh LIX, On convection currents in a horizontal layer of fluid, when the higher temperature is on the under side, *Philos. Mag. Ser. 6* 32 (192) (1916) 529–546, <https://doi.org/10.1080/14786441608635602>.
- [45] J. Wang, D. Wang, P. Lallemand, L.-S.S. Luo, Lattice Boltzmann simulations of thermal convective flows in two dimensions, *Comput. Math. Appl.* 65 (2) (2013) 262–286, <https://doi.org/10.1016/j.camwa.2012.07.001>.

- [46] D.B. Chirila, Towards Lattice Boltzmann Models for Climate Sciences: The GeLB Programming Language with Applications, Ph.D. thesis, University of Bremen, 2018, <https://elib.suub.uni-bremen.de/peid/D00106468.html>.
- [47] E.P. van der Poel, R.J.A.M. Stevens, D. Lohse, Comparison between two- and three-dimensional Rayleigh–Bénard convection, *J. Fluid Mech.* 736 (2013) 177–194, <https://doi.org/10.1017/jfm.2013.488>.
- [48] O. Shishkina, R.J.A.M. Stevens, S. Grossmann, D. Lohse, Boundary layer structure in turbulent thermal convection and its consequences for the required numerical resolution, *New J. Phys.* 12 (7) (2010) 075022, <https://doi.org/10.1088/1367-2630/12/7/075022>.
- [49] P. Mishra, A. De, M. Verma, V. Eswaran, Dynamics of large-scale flow in Rayleigh–Bénard convection, *J. Fluid Mech.* 668 (2011) 480–499.
- [50] H. Johnston, C.R. Doering, A comparison of turbulent thermal convection between conditions of constant temperature and constant heat flux boundaries, *Phys. Rev. Lett.* 102 (6) (2009) 064501, <https://doi.org/10.1103/PhysRevLett.102.064501>.
- [51] M. Herdin, N. Czink, H. Ozcelik, E. Bonek, Correlation matrix distance, a meaningful measure for evaluation of non-stationary MIMO channels, in: 2005 IEEE 61st Vehicular Technology Conference, vol. 1, IEEE, 2005, pp. 136–140.
- [52] Y. Zhang, Y.-X. Huang, N. Jiang, Y.-L. Liu, Z.-M. Lu, X. Qiu, Q. Zhou, Statistics of velocity and temperature fluctuations in two-dimensional Rayleigh–Bénard convection, *Phys. Rev. E* 96 (2) (2017) 023105.
- [53] A. Pandey, J.D. Scheel, J. Schumacher, Turbulent superstructures in Rayleigh–Bénard convection, *Nat. Commun.* 9 (1) (2018) 2118.
- [54] P. Kailasnath, K. Sreenivasan, G. Stolovitzky, Probability density of velocity increments in turbulent flows, *Phys. Rev. Lett.* 68 (18) (1992) 2766.
- [55] W. Sosa-Correa, R. Pereira, A. Macêdo, E. Raposo, D. Salazar, G. Vasconcelos, Emergence of skewed non-Gaussian distributions of velocity increments in isotropic turbulence, *Phys. Rev. Fluids* 4 (6) (2019) 064602.