Subgrid-scale parametrization of unresolved scales in forced Burgers equation using generative adversarial networks (GAN)

Received: 11 November 2020 / Accepted: 21 July 2021 / Published online: 15 November 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract  Stochastic subgrid-scale parametrizations aim to incorporate effects of unresolved processes in an effective model by sampling from a distribution usually described in terms of resolved modes. This is an active research area in fluid dynamics where processes evolve on a wide range of spatial and temporal scales. We propose a data-driven framework where resolved modes are defined as local spatial averages and deviations from these averages are the unresolved degrees of freedom. The proposed approach is applicable to a wide range of finite volume and finite difference numerical schemes commonly used to discretize many realistic problems in fluid dynamics. In this study, we evaluate the performance of conditional generative adversarial network (GAN) in parametrizing subgrid-scale effects in a finite difference discretization of stochastically forced Burgers equation. We train a Wasserstein GAN (WGAN) conditioned on the resolved variables to learn the distribution of the subgrid flux and, thus, represent the effect of unresolved scales. The resulting WGAN is then used in an effective model to reproduce the statistical features of resolved modes. We demonstrate that various stationary statistical quantities such as spectrum, moments and autocorrelation are well approximated by this effective model.

Keywords  Subgrid-scale parametrization · Generative adversarial network · Burgers equation

1 Introduction

Many physical processes in mathematics, engineering and other applied sciences are represented by differential equations. In practical numerical applications, partial differential equations are discretized in space and time. Often finite difference or finite volume numerical schemes are used for spatial discretization which results in a system of ordinary differential equations or difference equations. However, simulating discretized equations on a very fine mesh might be too costly computationally. Therefore, it is desirable to introduce coarse mesh (resolved) variables and develop closed-form effective equations for coarse quantities of interest. Spatial filtering is often used to define coarse variables in large Eddy simulation models. We adopt this strategy here as well. When developing an effective model for coarse variables, some of the physical processes may not be
fully resolved on the coarse spatial grid. Either some processes evolve on a much smaller spatial (subgrid) scale compared to the coarse mesh size or there are inherent properties of the system (e.g., nonlinearity, chaotic behavior) that cannot be captured by the chosen coarse discretization. The problem of incorporating effects of unresolved processes (or subgrid-scale processes) in an effective model is called parametrization (or subgrid-scale parametrization, in our context).

Developing accurate and fast parametrizations has been an active area of research in atmosphere–ocean community for many decades. For deterministic parametrization schemes, a function relating the unresolved modes to every state of the resolved modes is constructed by using either known physics or existing high-resolution data. This leads to incorporating the mean impact of subgrid processes on resolved modes. To be realistic, some measure of variability of the impacts of these processes must also be taken into account as proposed by Hasselman [24] in a visionary paper in 1976. Introduction of randomness leads to stochastic parametrizations. Stochasticity arises naturally due to an infinite number of possible valid realizations for subgrid processes. Therefore, some form of stochasticity needs to be used to represent this variability [35].

In recent decades, there have been a considerable number of studies focusing on stochastic parametrizations. Many methods have been developed which aim to parametrize effects of subgrid scales by incorporating stochastic terms with the goal of reproducing the statistical behavior of resolved scales. To mention a few, there are techniques which use regression models (e.g., [27, 47]), add stochastic perturbations to the mean parametrization terms [38, 41], utilize data-driven methods such as NARMAX fitting [12, 31] or use conditional Markov chains [14, 26, 28]. There are also more rigorous approaches (see recent review paper [40]) including stochastic mode reduction strategy exploiting scale separation between the resolved and subgrid-scale processes [17, 18, 32–34], reduction techniques based on the Zwanzig–Mori formalism [9–11, 25] and turbulence closure techniques [19, 20, 43]. In particular, stochastic mode reduction has been applied recently to reproduce the statistical behavior of local spatial averages in the truncated Burgers–Hopf model [15, 16] and the shallow water equations [52]. A more relaxed scheme that relies on weak coupling and uses ideas from response theory has been developed in [48]. Another approach uses maximum entropy principle for estimating probability density of unresolved scales [42, 44]. Stochastic parametrizations have been proved to be extremely useful in weather and climate science for analyzing prediction uncertainty, representing subgrid processes, capturing response to forcing and inducing regime transitions (see, e.g., [4]).

One drawback in some of the semi-analytical methods for constructing subgrid parametrizations is that these methods often require certain assumptions on the structure of coarse and/or subgrid variables (e.g., Gaussian closures or strong scale separation). Such assumptions are often necessary to make the model analytically tractable. In contrast to these semi-analytical methods, machine learning (ML) techniques offer approaches that do not require any assumptions on the coarse and/or subgrid variables. Instead, ML methods are capable of learning the underlying structure of subgrid processes from data alone.

Machine learning has become increasingly popular in recent years offering possible data-driven solutions for problems in many areas of physical and natural sciences. Machine learning also received increased attention in applied and computational fluid dynamics as well. Recently, neural networks (NNs) has become one of the most popular areas of machine learning and there are several applications of NNs in various models relevant to atmospheric research. In particular, NNs were used to represent clouds [39] and convection [37] in a global circulation model. NNs were also used to develop a parametrization for residual heating and moistening in a global circulation model [5]. Various deep learning methods such as reservoir computing and LSTM neural networks were used to predict trajectories of chaotic dynamics (e.g., [7, 45]). Methods used in these papers are classified as supervised learning and typically result in deterministic parametrizations. Here we use generative models to develop NN-based stochastic parametrizations.

Application of generative models is one of the relatively new areas in machine learning. These are primarily techniques from semi-supervised learning in which a network is trained to produce samples from a given distribution. One popular generative model is generative adversarial networks (GAN) introduced in [22]. This model consists of two networks—generator and discriminator. Given a training set, these two networks compete with each other—the generator attempts to produce new samples consistent with the training set, and the discriminator evaluates them with the goal of distinguishing whether these samples came from the original distribution or are outputs of the generator. Thus, GAN aims to learn the underlying distribution for the training data in order to produce new samples that a discriminator qualifies to be coming from that distribution. It is easy to see that using GAN would result in a stochastic parametrization. A modified version of this approach was applied to Lorenz 96 model [21]. Recently, an alternative training metric for generative adversarial networks has been proposed. In particular, it was demonstrated that Wasserstein GAN (WGAN) might have certain...
advantages over the traditional GAN during training [3, 23, 50]. These advantages include better convergence and interpretability of the Wasserstein metric.

The main goal of this work is to examine how well GANs perform as stochastic parametrizations of subgrid-scale processes. The choice of ML method used here is motivated by the generality of GANs and a successful application of this approach in other areas, such as imaging. In addition, GANs have been designed to learn and reproduce arbitrary probability distributions, which is exactly the main task behind developing accurate and efficient closures for subgrid terms. Here we utilize the setup similar to the large Eddy simulations where we define resolved variables as local spatial averages and subgrid-scale variables as fluctuations. In this paper, a conditional WGAN is trained to learn the distribution of subgrid processes and then used to generate samples of unresolved modes based on the current state of the resolved variables. We apply this approach for parametrizing local residuals in a flux discretization of stochastically forced Burgers’ equation. Specifically, a WGAN conditioned on resolved modes is trained to produce samples from the distribution of subgrid fluxes to be used in an effective model for local averages. The proposed approach follows a general framework for parametrizing subgrid fluxes in a reduced system for local averages and is applicable to a wide range of problems where finite difference or finite volume schemes are used in computations.

The rest of the paper is organized as follows. In Sect. 2.1, we discuss general methodology for defining coarse variables and representing the effective equations using flux discretization. In Sect. 2.2, we discuss the application of this methodology to the forced Burgers–Hopf model. In Sect. 2.3, we discuss the configuration and training of the neural network used to represent the subgrid fluxes. In Sect. 3, we present numerical results comparing stationary statistics of the full and reduced models and discuss the performance of the WGAN parametrization with respect to changes in forcing and resolution. Final remarks are presented in Sect. 4.

2 Methods

2.1 Subgrid flux parametrization

The method developed in this paper applies to systems based on flux difference discretization of partial differential equations that can be written in conservative form. More precisely, consider a state field \( u(x, t) \) which can be discretized on a uniform fine mesh \( M_f = \{ i \Delta x, \quad i = 0, 1, 2, \ldots, N - 1 \} \) resulting in a system of ODEs

\[
\frac{d}{dt} u_i = - \frac{f_{i+1/2} - f_{i-1/2}}{\Delta x} + \rho_i,
\]

where \( \Delta x = L/N \) and \([0, L]\) is the domain of integration. We supplement the above equation with periodic boundary conditions \( u(x, t) = u(x + L, t) \), but this is not essential for the formulation of the method. We define mesh points as \( x_i = i \Delta x \). It is also useful to define the (staggered) grid points \( x_{i+1/2} = x_i + \Delta x/2 = (t + 1/2) \Delta x \) and the \( i \)th interval \( [x_{i-1/2}, x_{i+1/2}] \), which is a standard formulation in the finite volume literature (see, e.g., [30]). Then the value \( u_i(t) \) is the numerical approximation for the value of \( u(x, t) \) at the center of the \( i \)th interval (i.e., \( u_i(t) \approx u(x_i, t) \)) or for the average over the \( i \)th interval, i.e., \( u_i(t) \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t) dx \), depending on whether the numerical method used is a finite difference or a finite volume scheme, respectively. More applicable to finite volume methods, \( f_{i+1/2} \) and \( f_{i-1/2} \) are discrete approximations for the flux at the right and left ends of the interval \( i \), respectively. Here, \( \rho_i \) is forcing discretized in space, i.e., \( \rho_i \approx \rho(x_i, t) \).

Equation (1) represents the fine mesh discretization which can be used for direct numerical simulations (DNS) of the underlying pde. Here we do not address the numerical accuracy of the discretization. We consider the fine mesh discretization (1) as an accurate model for direct numerical simulations where one can make numerical errors sufficiently small by the choice of numerical method and grid size. Performing DNS on a fine mesh can be computationally expensive and it is often desirable to obtain a closed-form reduced model for large-scale modes. To this end, we want to define resolved variables using a spatial averaging filter and obtain a closed-form equation for these resolved variables alone. This idea is similar to derivation of LES models, except our starting point is the discretized Eq. (1).

In this work, we use the “box” filter and define resolved modes as spatial averages over \( n \) fine grid points

\[
U_I = \frac{1}{n} \sum_{i=nI}^{n(I+1)-1} u_i \quad \text{for each } I = 0, \ldots, N/n - 1,
\]
on a coarser mesh \( \mathcal{M}_c = \{ X_I, I = 0, 1, 2, N/n - 1 \} \) over the same domain \([0, L]\) with \( \Delta X = nL/N \equiv n\Delta x \). The coarse mesh is defined implicitly by the averaging operator, but it is also easy to explicitly express \( X_I \) in terms of fine mesh grid points. Thus, coarse grid points can be computed as midpoints of the \( I \)th coarse mesh interval, i.e., \( X_I = (x_{nI-1/2} + x_{n(I+1)-1/2})/2 \). The interval associated with \( U_I \) is simply the union of fine mesh intervals \( nI \) to \( n(I+1) - 1 \) (same as the intervals associated with \( u_i \)’s in (2)). The unresolved degrees of freedom are defined as the residuals or deviations \( y_i = u_i - U_I(i) \), where coarse index \( I(i) \) can be computed as \( I = \lfloor i/n \rfloor \). Finally, we also define vectors of fine-scale variables, coarse-scale variables and fluctuations as \( u = \{ u_i, i = 0, \ldots, N - 1 \} \), \( U = \{ U_I, I = 0, \ldots, N/n - 1 \} \) and \( y = \{ y_i, i = 0, \ldots, N - 1 \} \), respectively.

Some degree of scale separation can be expected between \( U \) and \( y \) by this construction, but it might depend on the number of averaged points, \( n \). Typically, vector \( u \) is high-dimensional, while \( \dim(U) \ll \dim(u) \). Note that introduction of timescale separation is implicit since it is a consequence of applying the averaging operator in (2), in contrast to an explicit introduction of a small parameter to distinguish timescales among group of modes.

The dynamics of \( U \) can be recast as a coupled system for resolved modes \( U \) and fluctuation \( y \). This leads to the coupled system

\[
\frac{d}{dt} U_I = -\frac{f_{n(I+1)-1/2} - f_{nI-1/2}}{n\Delta x} + \rho^{U_I},
\]

\[
\frac{d}{dt} y_i = -\frac{f_{i+1/2} - f_{i-1/2}}{\Delta x} + \frac{f_{n(I+1)-1/2} - f_{nI-1/2}}{n\Delta x} + \rho^{y_i},
\]

where \( \rho^{U_I} \) and \( \rho^{y_i} \) are projections of the forcing on the respective modes. Here it is implicitly assumed that the forcing is slowly varying in space, so that \( \rho^{y_i} \) is negligible. Equation (3) describes evolution of resolved variables on the coarse mesh, while Eq. (4) described the dynamics of subgrid scales on the fine mesh. Note that the fluxes in (3) are the fine mesh fluxes on the boundary of the \( I \)th coarse mesh cell. All fluxes in Eqs. (3) and (4) depend on both the resolved and unresolved variables and can be obtained by direct substitution \( u_i = U_I(i) + y_i \) into the flux \( f_{i+1/2}(u) \).

Next, we introduce a decomposition of fluxes in Eq. (3)

\[
f_{n(I+1)-1/2}(U, y) = F_{I+1/2}(U) + G_{I+1/2}(U, y) \\
fnI-1/2(U, y) = F_{I-1/2}(U) + G_{I-1/2}(U, y)
\]

where terms \( F_{I+1/2}(U) \) describe self-interactions of local averages on the coarse mesh. It is often natural to maintain the same flux form for the coarse variables as in the full Eq. (1). Therefore, the functional form for \( F_{I+1/2} \) is often chosen to be identical to \( f_{i+1/2} \). The term \( G_{I+1/2}(U, y) \) is referred to as the subgrid flux at the right boundary of cell \( I \) or simply the right subgrid flux. Physical processes which cannot be adequately resolved with the mesh size \( \Delta X = n\Delta x \) are represented by \( G_{I+1/2} \). Note that although we write the entire vector \( U \) and \( y \) as predictors, only few neighboring \(^1\) resolved and subgrid variables are actually used in the computations of the right-hand size of the equation for \( U_I \). Equation (5) can be considered as the definition of the subgrid flux \( G_{I+1/2}(U, y) \).

In this paper, we use the same expression for the coarse mesh flux as for the fine mesh flux, i.e., functional forms of the nonlinearity in \( f_{i+1/2}(u) \) and \( F_{I+1/2}(U) \) are the same. However, it does not have to be the case. It might be beneficial to use a different formula for computations of nonlinear terms in the coarse mesh fluxes (e.g., lower order to accelerate computations). In this case, the decomposition becomes \( f_{n(I+1)-1/2}(U, y) = \tilde{F}_{I+1/2}(U) + \tilde{G}_{I+1/2}(U, y) \), where \( \tilde{F}_{I+1/2}(U) \) is prescribed beforehand using some modeling and/or computational considerations. This leads to a different definition of subgrid terms. Thus, subgrid terms depend on a number of computational and modeling parameters in the problem. However, the GAN parametrization framework presented here is quite general and relies only on generated computational data. Therefore, it can be applied to many flux definitions, including higher-order finite volume schemes which use limiters.

In the most general sense, a parametrization aims to approximate \( G_{I+1/2}(U, y) \) for every cell \( I \) in terms of the resolved modes only so that the resulting closed system emulates, in some sense, the behavior of resolved variables \( U \) in the full dynamics. Note that given a state \( U \) of the system, there can be many possible values of the unresolved variables, possibly due to the nonlinearity, chaotic behavior of the underlying dynamics, etc.

\(^1\) Number depends on the stencil used to represent the numerical flux.
Thus, there is some stochasticity in $G_{I+1/2}$ when its dynamics is observed on the coarser mesh and $G_{I+1/2}(t)$ can be considered a random variable (see, e.g., [35]), possibly conditioned on the system state $U(t)$. Therefore, the main focus of this paper is on stochastic parametrizations where approximations to $G_{I+1/2}$ are drawn from the distribution of the subgrid flux conditioned on current values of large-scale variables $U(t)$. Once the subgrid flux is generated, it is used in a time-stepping method to predict the next value of dynamic variables. Thus, our model is Markovian. One can build more elaborate non-Markovian parametrizations using conditioning on past values of the large-scale data, i.e., using $U(t-k\Delta)$, $k = 1, \ldots, K$ with some lag $\Delta$. However, in this case the lag $\Delta$ might restrict the integration time step for the reduced model. For instance, in the context of conditional Markov chain parametrizations, the integration time step for the reduced model must be equal to $\Delta$ [13,36]. Thus, conditioning only on current values of large-scale variables allows to avoid additional constraints on the integration time step for the reduced model.

We would like to comment that the ansatz introduced in (3), (4) and (5) is only one possible decomposition of the system (1). We choose to maintain the flux difference formulation for several reasons. First, the reduced model expressed in flux difference form for subgrid terms enforces mass conservation (at least on average). Second, the flux difference scheme allows for better understanding of relative strengths of flux terms and potential introduction of empirical corrections. Third, if one tries to develop a parametrization for $G_{I+1/2} - G_{I-1/2}$ directly, then such parametrization would potentially depend on a larger “conditioning” stencil (e.g., at least on $U_{I-1}$, $U_I$ and $U_{I+1}$) and thus requires additional computational efforts to generate training data. In general, the number of conditions should be kept to a necessary minimum, since additional conditions require larger training datasets. In practice, conditions are implemented by binning the data into appropriate bins in $\mathbb{R}^d$, where $d$ is the number of conditions. Therefore, longer simulations are required for larger $d$ to have sufficient amount of data in bins used for training the parametrization. Fourth, flux formulation in (5) allows to apply flux limiters to subgrid parametrizations, if necessary. Flux limiters have become a standard computational tool for hyperbolic systems, and one can apply flux limiters to introduce physical constraints (e.g., positivity) for the solution of the reduced system. It is certainly possible that for a particular problem at hand, parametrizing the divergence of subgrid terms $G_{I+1/2} - G_{I-1/2}$ can be easier than parametrizing the fluxes $G_{I+1/2}$ and $G_{I-1/2}$ separately. However, for reasons outlined above, one should attempt to develop parametrizations for $G_{I+1/2}$ and $G_{I-1/2}$ independently first and only consider parametrizing the difference $G_{I+1/2} - G_{I-1/2}$ if such attempts fail.

Replacing the subgrid fluxes with a suitable parametrization

$$G_{I+1/2}(U, y) \approx \tilde{G}_{I+1/2}(U),$$

the effective (or reduced) equation for large-scale variables becomes

$$\frac{d}{dt} U_I = -\frac{F_{I+1/2} - F_{I-1/2}}{\Delta X} - \frac{\tilde{G}_{I+1/2} - \tilde{G}_{I-1/2}}{\Delta X} + \rho U_I,$$  \hspace{1cm} (6)

where $\Delta X = n \Delta x$. Here, $\tilde{G}_{I+1/2}$ for each $I$ is a random variable, possibly conditioned on some neighboring large-scale variables. Parametrization $\tilde{G}_{I+1/2}$ represent a mechanism for effectively sampling the distribution of $\tilde{G}_{I+1/2}$ and, thus, allows for an efficient numerical simulation of the effective equation in (6). The structure and complexity of subgrid fluxes $G_{I+1/2}$ is determined by flux discretization and decomposition done in (5). Knowing its properties is helpful in the construction of a parametrization scheme. However, in general, there is no restriction on the form of the parametrization. We also refer to the model (6) without subgrid corrections (i.e., $\tilde{G} \equiv 0$) as the bare truncation (BTR) model.

In this paper, we utilize a neural network to represent $\tilde{G}_{I+1/2}$. In particular, we use generative adversarial network (GAN) to generate samples from the distribution of $\tilde{G}_{I+1/2}$. Since the parametrization is inherently stochastic, we focus our attention on reproducing the statistical properties of the resolved variables. Emphasis on statistical properties is, for instance, motivated by necessity to develop effective model for ensemble forecasts in climate and weather predictions.

### 2.2 Forced Burgers’ equation

The viscous Burgers equation is one of the well-studied nonlinear partial differential equations and it is often used as a prototype model for turbulence [6]. However, there are also subtle differences between Burgers equations and more realistic turbulence models, such as non-locality in the incompressible Navier–Stokes
equations due to the presence of pressure. We consider a stochastically forced Burgers equation over a periodic domain $[0, L]$ given by

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left( \frac{u^2}{2} - \nu \frac{\partial u}{\partial x} \right) = \rho(x, t), \quad \text{(7)}$$

where $\nu$ is the viscosity coefficient and $\rho$ is a large-scale stochastic forcing. To produce a system admitting the form given in (1), we adopt the flux discretization used in [51] and the discrete approximation for the flux becomes

$$f_{i+1/2} = \frac{1}{6} \left( u_{i+1}^2 + u_{i+1}u_i + u_i^2 \right) - \frac{\nu}{\Delta x} (u_{i+1} - u_i). \quad \text{(8)}$$

Thus, the above equation combined with (1) is the fine mesh discretization of the PDE and we assume that (7) is sufficiently well resolved on this mesh. Large-scale forcing is applied to low wavenumbers $k \in K$. Following [8] (and also [15, 52]) the forcing term is chosen such that $\rho^{y_i} = 0$ for all $i$, and

$$\rho^{U_I} = \frac{A}{\sqrt{\Delta t}} \sum_{k \in K} \left[ \alpha_k \cos \left( 2\pi k \frac{n \Delta x}{L} \right) + \beta_k \sin \left( 2\pi k \frac{n \Delta x}{L} \right) \right],$$

where $\Delta t$ is the integration time step of the DNS, $A$ is the forcing amplitude, and $\alpha_k$ and $\beta_k$ are independent random coefficients sampled from $N(0, 1)$. For the discrete equation (1) with the flux discretization (8), the self-interactions of the resolved variables and the subgrid flux in (5) become

$$f_{n(I+1)-1/2} := F_{I+1/2} \left( U_{I+1}, U_I \right) + G_{I+1/2} \left( U_{I+1}, U_I, y_{I+1}^f, y_I^f \right), \quad \text{(9)}$$

where

$$F_{I+1/2} = \frac{1}{6} \left( U_{I+1}^2 + U_{I+1}U_I + U_I^2 \right) - \frac{\nu}{\Delta x} (U_{I+1} - U_I) \quad \text{(10)}$$

and the subgrid flux depends on resolved variables $U_I$ and $U_{I+1}$, as well as on the unresolved modes $y_{I+1}^f$ and $y_I^f$. Here $y_{I+1}^f$ and $y_I^f$ represent residuals located the right boundary of cell $I$ and left boundary of cell $I+1$, respectively. Thus, for simpler notations, instead of using fine mesh index, we use $l$ and $r$, signifying left and right, to refer to location of the residuals with respect to its coarse cell index subscript.

It is important to note that after applying the averaging operator in (2), the resulting flux difference scheme for coarse variables (10) results in a viscosity $n$ times larger than that in the full model (i.e., there is $\Delta x$ in the denominator in (10), not $\Delta X$). This is a direct consequence of applying the discrete averaging operator (2) to the equation (1) with flux definition in (8). This means that setting the subgrid flux corrections $G_{I+1/2}$ to zero in (6) (same as setting $G_{I+1/2} \equiv 0$ in (9)) does not result in a coarse mesh discretization of the Burgers model (7) with a low resolution $N/n$. Thus, the bare truncation (BTR) model is not identical to the low-resolution discretization of the Burgers equation. Therefore, the parametrization is not a correction to the additional numerical error incurred due to a larger mesh size. Instead, parametrization is a correction to both the nonlinear part and the diffusive part of the flux. Hence, the subgrid flux term can further be divided into the subgrid contribution to the nonlinear part and to the viscosity term. That is, we can decompose the subgrid flux as

$$G_{I+1/2} \left( U_{I+1}, U_I, y_{I+1}^f, y_I^f \right) = G_{I+1/2}^{(1)} \left( U_{I+1}, U_I, y_{I+1}^f, y_I^f \right) - \frac{\nu}{\Delta x} G_{I+1/2}^{(2)} \left( y_{I+1}^f, y_I^f \right), \quad \text{(11)}$$

where $G_{I+1/2}^{(1)}$ and $G_{I+1/2}^{(2)}$ are subgrid flux corrections to the nonlinear part and diffusion, respectively. There are several reasons for such decomposition. First, terms on the right-hand side of (11) have different complexity—the first term involves quadratic interaction between resolved and unresolved modes, and the second term only involves linear relationship between unresolved modes. Second, we can investigate balance different terms and scalability of the stochastic parametrization with respect to varying $n$.

**Reference setup** Given the discrete flux and forcing above, we use a hybrid approach for integrating the equation in (1). In particular, we use the third-order Runge–Kutta (RK3) time-stepping scheme for the deterministic terms and then add the forcing term using the explicit Euler method. The motivation for this
hybrid approach is that a higher-order method is essential for accurately representing important properties of the PDE in (7) (e.g., conservation of momentum and energy by nonlinear terms) and overall stability of simulations. On the other hand, we can expect that small numerical errors in the random forcing do not significantly affect statistical properties of solutions. Parameters in direct numerical simulations are presented in Table 1. We perform direct numerical simulations (DNS) of the full model in (7) and generate long stationary time series of \( u_i(t) \). We then compute the resolved variables using (2) and compute numerically their statistical properties, such as spectra, distribution, correlation function and kurtosis. The main emphasis of the current work is on how well the reduced model approximates the statistical properties of large-scale variables obtained in DNS.

To understand the need for stochastic parametrization, we must investigate the dependence of the subgrid fluxes \( G_{i+1/2}^{(i)} \) for \( i = 1, 2 \) on the resolved variables \( U_I \) and \( U_{I+1} \). A typical profile of \( U(t) \) obtained from the DNS is characterized by several shock-like regions (up to three for parameters considered here) propagating and merging over the domain. Closely observing the values of \( G_{i+1/2}^{(i)} \) reveals that its magnitude is often relatively higher near regions with large gradients. This is especially true for \( G_{1+1/2}^{(1)} \), the nonlinear subgrid correction. This is not surprising since the absence of subgrid fluxes in 6 results in a reduced model with viscosity \( n \) times larger than that in the full model. Hence, the reduced model without any subgrid corrections “smoothes” the solution, especially at regions with large gradients. Moreover, the variability in the values of \( G_{i+1/2}^{(i)} \) increases as the near-shock regions becomes “steeper.” These regions can be found effectively by simply noting the values of \( |U_I - U_{I+1}| \).

To illustrate the need for stochastic parametrization, we depict in Fig. 1 the marginal distribution (histogram) of the subgrid flux \( G_{1+1/2}^{(1)} \) conditional on the difference of large-scale variables \( |U_0 - U_1| \). We would like point out that here we are analyzing the behavior of the “true” subgrid flux computed directly from the DNS of the resolved model (1).

Figure 1 demonstrates two important points. First, this figure illustrates the need for a stochastic parametrization. The distribution of the subgrid flux \( G_{1+1/2}^{(1)} \) is quite broad, especially for larger values of \( |U_0 - U_1| \). Figure 12 The linear subgrid correction term \( G_{I+1/2}^{(2)} \) can be reasonably well approximated by a linear regression on \( U_I \) and \( U_{I+1} \); possibly because of its lower complexity compared to the nonlinear subgrid flux term.
suggests that a deterministic parametrization might be possible for smaller values of \(|U_0 - U_1|\), but for values of \(|U_0 - U_1| \geq 0.2\) a stochastic term must be included to account for the variability of the subgrid flux. Second, Fig. 1 clearly demonstrates dependence of the subgrid flux on the difference between the adjacent local averages. Therefore, it is essential to include both local averages \(U_1\) and \(U_{I+1}\) as an input for the stochastic parametrization scheme. Here we choose to condition the parametrization on \(U_1\) and \(U_{I+1}\) separately, i.e.,

\[
\tilde{G}_{I+1/2}^{(l)} \equiv \tilde{G}_{I+1/2}^{(l)}(U_1, U_{I+1})
\]

instead on conditioning only on the difference \(|U_1 - U_{I+1}|\) since conditioning on \(U_1\) and \(U_{I+1}\) is more general and does not significantly increase the computational complexity compared to conditioning only on \(|U_1 - U_{I+1}|\). We also tried to condition the parametrization using a larger stencil, but that did not amount to any significant changes in the performance of the effective model. We discuss the performance of the GAN parametrization and Fig. 1 further in Sect. 3.1.

### 2.3 WGAN parametrization

In this section, we present the structure of the network and describe the training procedure. We also describe data used for training and verification.

#### 2.3.1 GAN

Generative adversarial networks (GANs) are machine learning tools that given a dataset generated by sampling some probability distribution, attempt to learn the underlying distribution in order to generate additional samples. The framework of GANs consists of two networks, generator \(G\) and discriminator \(D\), which are in mutual competition. Although only the generator \(G\) is needed to generate new samples, training of both networks \(G\) and \(D\) have to performed simultaneously. The network architecture of \(G\) and \(D\) is completely independent and can involve various neural network constructions such as simple feed-forward networks and convolutional networks [46]. Convolutional networks are the most popular architecture for imaging applications because of a very high-dimensional input, and simple feed-forward multilayer networks are used most often in other applications where the dimension of input is relatively low. The generator accepts a multi-dimensional noise vector \(z\) from some specified distribution \(p_z\) (e.g., uniform or Gaussian) as an input and transforms it to a “fake” sample. The discriminator attempts to distinguish between the “real” sample \(x\) (one coming from training data with some unknown distribution \(p_x\)) and the generated “fake” sample \(G(z)\). Training is done by updating the network weights of the generator and the discriminator by minimizing the game-theoretic loss

\[
\mathbb{E}_{x \sim p_x} \left[ \log(D(x)) \right] + \mathbb{E}_{z \sim p_z} \left[ \log(1 - D(G(z))) \right].
\]

Note that expectations are taken over a batch of samples and in practice it is computed as an empirical sample mean. Training of both networks proceeds sequentially. First, the discriminator is trained for some number of iterations and then, with the discriminator’s parameters frozen, the generator is updated by minimizing the value of the above loss. The optimality is obtained when equilibrium is reached. Intuitively, it is achieved when the discriminator is indifferent whether a generated sample is “fake” or “real.” It can be shown that optimizing these losses is equivalent to minimizing the Jensen–Shannon divergence between the distributions of “real” and generated samples [22].

In this paper, we use conditional GAN. In conditional GANs, it is assumed that the underlying distribution \(p_x\) depends on some parameters (or labels with known values) and these labels are also used as an input to both the generator \(G\) and the discriminator \(D\). In particular, in our case it is reasonable to assume that the subgrid flux \(G_{I+1/2}(t)\) depends on the state of local averages \(U_I(t)\) and \(U_{I+1}(t)\). Hence, we use values of local averages at the current time, \(U_I(t)\) and \(U_{I+1}(t)\), as “labels.”

It is widely known that it can be difficult to train GANs because of stability and divergence issues. There are also problems such as mode collapse and vanishing gradients (see [2], for example). One improvement in training GANs is to use another metric called Wasserstein distance between distributions. This variation is called Wasserstein GAN or WGAN. In WGAN, the discriminator needs to satisfy some Lipschitz condition so additional constraint or penalty must be imposed. This can be done by adding a gradient penalty to the loss function [23]. This is known as WGAN-GP. Its loss function is

\[
\mathbb{E}_{z \sim p_z} [D(G(z))] - \mathbb{E}_{x \sim p_x} [(D(x))] + \lambda \mathbb{E}_{x \sim p_x} \left[ \left( \| \nabla_x D(\hat{x}) \|_2 - 1 \right)^2 \right],
\]
where \( \hat{x} \) are uniformly chosen from segments joining \( x \) and \( G(z) \). The reasons for the use of WGAN over GAN are the advantages in the training procedure, such as improved stability and meaningful loss function [3]. Note that the study does not aim to compare the performance between GAN and WGAN. The usual GAN with chosen parameters can converge and give good results as well. However, training GANs typically requires an additional offline criterion in order to determine how good the samples are, such as the use of metrics in [21]. This additional criterion is not necessary for WGANs since in WGAN training, smaller discriminator’s loss correlates with “smaller difference” between distributions of “fake” and “real” samples. We refer to [3,23] for thorough discussion on this matter.

2.3.2 Network

Our goal is to achieve a local parametrization, that is, produce approximate subgrid flux at each \( I + 1/2 \) conditioned only on immediate neighboring local averages \( U_I \) and \( U_{I+1} \). Local parametrizations have many attractive computational features such as, they require less data for training, they are more computationally efficient and easily parallelizable, they can be trained independently for non-homogeneous domains, etc. Thus we construct a local parametrization.

Figure 2 shows a schematic diagram of the conditional WGAN used. The generator accepts a vector \( z \) consisting of two random numbers independently drawn from the uniform distribution on \([-1,1]\) and the conditions \( U_I \) and \( U_{I+1} \) at a specific time \( t \). It outputs two values—the “fake” samples \( \tilde{G}^{(1)}_{I+1/2} \) and \( \tilde{G}^{(2)}_{I+1/2} \). The discriminator accepts values of \( G^{(1)}_{I+1/2} \) and \( G^{(2)}_{I+1/2} \), which are computed from DNS, conditions \( U_I \) and \( U_{I+1} \), and the generated \( \tilde{G}^{(1)}_{I+1/2} \) and \( \tilde{G}^{(2)}_{I+1/2} \). It then outputs scores to the inputs \( G^{(1,2)}_{I+1/2} \) and \( \tilde{G}^{(1,2)}_{I+1/2} \). Note that in vanilla GAN, discriminator outputs probabilities of being “real.” Here we use scores because the loss is WGAN corresponds to scores. The smaller the difference between the scores given to the generated samples and the real subgrid fluxes, the smaller the value of the loss. This implies smaller Wasserstein distance between the respective distributions.

Both the generator and the discriminator are fully connected feed-forward networks with 3 hidden layers, 16 neurons in each layer and Leaky ReLU activation. We would like to point out that network architecture does not play a significant role. At least two hidden layers are required to produce adequate results in the present application, and three hidden layers seem to be optimal since adding more layers does not significantly improve the performance of the reduced model.

Note that this architecture is somewhat basic, but still serves the purpose very well. As one would expect, adjustments might be needed when the complexity or dimension of the system is increased. In more complex systems, it might be necessary to add (non-traditional) layers or modify the loss to capture some physical properties as in [29,49]. In the present paper, the design of the parametrization only guarantees mass conservation. We did not make any modification of the loss function or hidden/output layers that imposes physical constraints.

2.3.3 Training

We perform high-resolution numerical simulation of the full model and sample the solution every \( \Delta t_{sample} = 50\Delta t \) with \( \Delta t = 0.01 \). However, the time step of the reduced model is not restricted to either \( \Delta t \) or \( \Delta t_{sample} \).

The only restriction on the time step of integration is the stability of the reduced model. This issue is discussed further in Sect. 3.1 To summarize, the approach developed here is applicable to a range of sampling time steps. The choice of the sampling time step is motivated by several factors—(i) correlation time of the variables of interest, (ii) desire to accelerate computations by selecting a bigger time step for the reduced model, (iii) stability and accuracy of the time-stepping scheme for the reduced model and (iv) generating sufficient amount of training data in a reasonable amount of time. The correlation time for the variables of interest (i.e., cell averages) in our case is \( CT \approx 150 \) (see Fig. 6), so consecutive data points are correlated at \( 50\Delta t \). Therefore, stability and accuracy of the time-stepping scheme for the reduced model are the primary consideration for restricting the time step. Also, increasing sampling time step is equivalent to performing direct numerical simulations for a longer time. Since GAN training typically requires large amount of data, this can be an important practical consideration in realistic problems in higher dimensions.

After a burnout of 10,000 model time units, time series of all four variables \( (U_0, U_1, G^{(1)}_{1/2} \) and \( G^{(2)}_{1/2} \) are saved until \( 6 \times 10^5 \) samples for each are obtained. The first \( 10^5 \) is used for training and the rest is for validation.
Fig. 2 Conditional WGAN structure for subgrid flux parametrization

The code for training and training data set is available at github (see Acknowledgments). We use testing data to ensure that the Wasserstein distance does not increase for values of coarse variables which were not used in training. In addition, we also use training data to compare various marginal distributions (e.g., Fig. 1).

Adam learning method is implemented for optimization with minibatch size of 400. Initial learning rate is set to $2 \times 10^{-5}$. During training, we keep track of the discriminator loss using the validation set. Note that this is a rough estimate of the Wasserstein distance between the distributions of the generated subgrid fluxes and the true data from DNS, in the sense that the 1−Lipschitz test functions is the family of discriminators obtained during training. Its value is offset by the gradient penalty, but since the penalty value tends to zero, this difference becomes irrelevant with the duration of the training. Hence, the discriminator loss might serve as the test error and the stopping criterion for selecting the optimized weights in the generator. Figure 3 shows the trajectory of discriminator loss on the validation set. We test optimized weights at epoch 100 in the reduced model for this study as the loss curve flattens around that training time. We also tested using weights at epoch 110 and 120, but the results does not change significantly.

Aside from the discriminator loss, as another way of (informal) verification, we also observe the contour plots of $U_0$ versus $\tilde{G}_{1/2}^{(i)}$ during training. Sample plots are presented in Fig. 4. This plots serve as a visual approximate for the joint distribution of local average and subgrid flux at the first coarse cell (i.e., $I = 0$).

Figure 4 depicts how generated subgrid fluxes tends to concentrate near the origin during training. On the right we also depict the “true” distribution of $(U_0, G_{1/2}^{(i)})$. The contour plot $(U_0, \tilde{G}_{1/2}^{(i)})$ at epoch 100 is a good approximation of the “true” distribution, so that the resulting GAN parametrization has a good performance. Training for 150 epochs takes around 1.5 h. This is a reasonable training time for one-dimensional model, and we can expect that training time would not a bottleneck for developing data-driven parametrizations in 2D and 3D.

3 Results

Aside from the bare truncation model (BTR) which integrates reduced Eq. (6) in time without any subgrid flux corrections, we also compare the performance of GAN parametrization with a polynomial regression with additive noise (POLY) similar to one used in [47]. This reduced model parametrizes subgrid scales by fitting a third-order polynomial with Gaussian additive noise estimated from regression residuals. The deterministic part of POLY parametrization is a multivariate polynomial for each $G_{1/2}^{(i)}$ ($i = 1, 2$) regressed against $U_0$ and

---

3 We did the same for $\tilde{G}_{1/2}^{(2)}$ but its contour plot resembles the “truth” very early during training, so we focus on $\tilde{G}_{1/2}^{(1)}$ instead.

4 Test is done using Tensorflow on a 6-core 32GB RAM 2.6 GHz computer with RTX 2070 GPU.
Subgrid-scale parametrization

Fig. 3 Discriminator loss on validation set

![Fig. 3 Discriminator loss on validation set](image)

Fig. 4 Left four subplots: contour plots of \((U_0, G_{1/2})\) at different epochs, right subplot: contour plot \((U_0, G_{1/2})\) using the training set

![Fig. 4 Left four subplots: contour plots of \((U_0, G_{1/2})\) at different epochs, right subplot: contour plot \((U_0, G_{1/2})\) using the training set](image)

\(U_1\). Coefficient estimation was done in MATLAB. The same polynomial parametrization (with independent noise realizations) is applied for each coarse cell.

**Time integration** For BTR, Runge–Kutta of order 3 (RK3) is used. For the other two reduced models (POLY and GAN), a hybrid scheme is implemented. That is, the deterministic part of the equation is integrated using RK3 and forcing and stochastic terms (additive noise or GAN-generated samples) are then added using Euler-type discretization. Although the resulting scheme is of low order, RK3 time stepping for the deterministic part can be beneficial for the stability of the time-stepping scheme for larger integration time steps. Time step used in simulations of the reduced model is not restricted to the sampling time step of the data. The only practical consideration is the stability of the reduced model. We investigated sensitivity of the reduced model with respect to the integration time step. Results for the stationary statistical properties of coarse variables in the numerical simulations of the reduced model with different time steps are presented in Table 5. These results demonstrate that the reduced model is not sensitive to changes in the integration time step and can be simulated numerically with a range of time steps. In particular, we were able to simulate the reduced model with a much bigger time step \(50\Delta t\) compared to direct numerical simulations on a fine mesh performed with time step \(\Delta t\). Robustness of the reduced model with respect to the integration time step allows for a considerable acceleration of simulations of the reduced model even if data are sampled with a very small time step (e.g., if \(\Delta t_{\text{sample}} = \Delta t\)).
Typically, it is more challenging to reproduce two-point statistical properties, compared to the one-point stationary statistics. Therefore, we also compute two-time stationary statistics and compare performance of the three reduced models to the results of the DNS. Two-time stationary correlation function and kurtosis are

Table 2 Mean and variance for conditional (on \(|U_0 - U_1| \in [a, b]\)) distribution of subgrid fluxes \(G_{1/2}^{(1)}\) and \(\tilde{G}_{1/2}^{(1)}\) computed from DNS and GAN parametrization, respectively. Corresponding distributions are depicted in Fig. 1

| \([a, b]\) | Mean DNS | Var DNS | Mean GAN | Var GAN |
|---|---|---|---|---|
| \([0.1, 0.2]\) | 0.0489 | 0.0063 | 0.0393 | 0.0058 |
| \([0.2, 0.3]\) | 0.0935 | 0.0154 | 0.0789 | 0.0137 |
| \([0.3, 0.4]\) | 0.1425 | 0.0277 | 0.1213 | 0.025 |
| \([0.4, 0.5]\) | 0.1842 | 0.0455 | 0.165 | 0.0411 |

Corresponding distributions are depicted in Fig. 1.

Table 3 Variance, fourth moment and correlation time of resolved modes in DNS, BTR model (resolution \(N/n\) with larger viscosity), LOW-RES model (resolution \(N/n\) with smaller viscosity) and reduced models (resolution \(N/n\) with polynomial (POLY) and GAN parametrizations.

| | Variance | Rel. error | Fourth moment | Rel. error | Corr. time | Rel. error |
|---|---|---|---|---|---|---|
| DNS | 0.0393 | | 0.0043 | | 114.26 | |
| BTR | 0.0355 | (9.7%) | 0.0037 | (14.1%) | 125.02 | 9.4% |
| LOW-RES | 0.0699 | 77% | 0.025 | 478% | 39.32 | (65.6%) |
| POLY | 0.0376 | (4.3%) | 0.0045 | 4.4% | 114.72 | 1.3% |
| GAN | 0.0393 | 0.1% | 0.0044 | 1.5% | 112.32 | (1.7%) |

3.1 Stationary statistics

Statistical properties of trajectories are often used to assess the performance of reduced models. To this end, for each model, we performed five long simulations with integration time step \(\Delta t = 0.01\) until time 160,000. Each simulation has different realizations of forcing, each also different from forcing realization used in generation of training and validation sets. Samples are taken every \(50\Delta t = 0.5\) after skipping \(t = 10,000\) to guarantee that the system is in stationary regime. All statistics are computed as ensemble and spatial averages (since the system is spatially homogeneous).

We also would like to point out that generated subgrid corrections exhibit the same variability pattern as for the DNS (see Fig. 1). PDFs of the generated \(\tilde{G}_{1/2}^{(1)}\) conditional on the value of \(|U_0 - U_1|\) from the test data are depicted in the right part of Fig. 1. In particular, we observe larger variability of the subgrid corrections for larger difference \(|U_0 - U_1|\). Therefore, magnitude of subgrid fluxes increases with \(|U_0 - U_1|\). This demonstrates that the GAN generator provides higher corrections at regions near shocks. In addition, mean and variance of distributions in Fig. 1 are also presented in Table 2. The GAN parametrization performs quite well in both regions with low and high gradients. The GAN parametrization correctly reproduces the overall trend for both mean and variance. We would like to point out that here we compare estimation of the right hand of the reduced equation with the “true” right-hand side computed by the DNS. Estimation of the right-hand side can be often mode challenging compared to the estimation of the statistical properties of reduced variables.

Energy spectrum and probability density of the resolved variables in the DNS of the full model and different reduced models are presented in Fig. 5. Stationary variance and fourth moment for \(U_1\) are also presented in Table 3. Energy spectra clearly demonstrate that a subgrid parametrization is necessary, since the bare truncation (BTR) underestimates the spectrum at high wavenumbers. The role of the parametrization is then to introduce corrections which mimic the subgrid fluxes and reproduce the statistical properties more accurately. Both polynomial (POLY) and neural network (GAN) reduced models reproduce energy spectrum quite accurately with GAN reduced model providing a slightly better approximation at the highest wavenumbers. Figure 5 shows that the PDF of the reduced variables is approximately normal and the shape is best reproduced by the GAN parametrization. In addition, the GAN parametrization also reproduces one-point statistics (see Table 3) slightly more accurately compared to the polynomial reduced model. Mean values and third moments of reduced variables are omitted as all of them are very close to zero, similar to the DNS results. Because of higher viscosity and absence of any corrections, bare truncation is expected to underestimate the variance and hence results in a narrower PDF. Polynomial model provides some improvement for one-point statistics when compared to the bare truncation, but still the PDF is slightly narrower than the results of the DNS. Typically, it is more challenging to reproduce two-point statistical properties, compared to the one-point stationary statistics. Therefore, we also compute two-time stationary statistics and compare performance of the three reduced models to the results of the DNS.
Subgrid-scale parametrization

Fig. 5 Energy spectrum (left) and PDFs (right) of resolved variables $U_i$ in direct numerical simulations (DNS) and simulations of low-resolution models. BTR denotes the bare truncation model, POLY denotes the polynomial parametrization, and GAN denotes the NN parametrization.

Fig. 6 Two-point stationary statistics of resolved variables in direct numerical simulations (DNS) and simulations of low-resolution models. BTR denotes the bare truncation model, POLY denotes the polynomial parametrization, and GAN denotes the NN parametrization. Left—correlation function. Right—kurtosis $K(s)$ depicted in Fig. 6. Kurtosis is a measure of non-Gaussianity and it is given by

$$K(s) = \frac{\langle U_i(t)^2 U_i(t + s)^2 \rangle}{\left( \langle U_i(t)^2 \rangle + 2 \langle U_i(t) U_i(t + s) \rangle \right)^2}.$$  

The value of $K(s)$ is exactly one for Gaussian models. All three reduced models reproduce stationary autocorrelation very well. The GAN parametrization predicts the non-Gaussian features better compared to POLY and BTR. Relative errors for $K(s)$ for the POLY model are approximately 8–10% compared to approximately 3% relative errors for $K(s)$ in the GAN model. This is particularly evident for short lags. Clearly, GAN reduced models performs very well in terms of capturing long-term statistical properties of local averages. Polynomial model also performs reasonably well. However, recall that polynomial model includes additive Gaussian noise. Stationary statistics indicates that local averages are almost Gaussian—we can see that the PDF is very close to a Gaussian distribution and kurtosis does not deviate significantly from one. We can expect that POLY parametrization should perform worse for stronger non-Gaussian models. On the other hand, due to the general nature of the noise in the GAN reduced model, we expect that this model should perform adequately for non-Gaussian processes as well. This will be investigated in subsequent papers. Also, there are particular realizations of forcing and noise that make the simulation of the reduced model with POLY parametrization unstable. We suspect that POLY parametrization might be overestimating the magnitude of the subgrid fluxes. This is verified when we increase forcing amplitudes which is discussed in later sections. We did not experience any stability problems with the GAN model for any realization during our test.

The GAN parametrization appears to be anti-diffusive. We performed additional simulations where we compared individual trajectories in DNS, bare truncation (no parametrization) and reduced model with GAN parametrization. We used exactly the same forcing realizations and initial conditions in these models and
compared the three resulting trajectories. Bare truncation model appears to be more diffusive compared to the DNS; this is particularly evident in regions near shocks (large curvature of the solution). GAN parametrization reproduces curvature of solutions better compared to bare truncation. Therefore, GAN parametrization is able to correct the diffusive nature of bare truncation.

Comparison between BTR and low-resolution model As pointed out in Sect. 2.2, the BTR model is not equivalent to the low-resolution (LOW-RES) model of the Burgers equation with \( N/n \) points. The flux for the low-resolution model has the form similar to the BTR model, except for a smaller diffusion, i.e., the flux in the low-resolution model is given by

\[
\text{LOW-RES:} F_{I+1/2} = \frac{1}{6} \left( U_{I+1}^2 + U_{I+1}U_I + U_I^2 \right) - \frac{\nu}{\Delta x} (U_{I+1} - U_I)
\]

compared to the BTR flux

\[
\text{BTR:} F_{I+1/2} = \frac{1}{6} \left( U_{I+1}^2 + U_{I+1}U_I + U_I^2 \right) - \frac{\nu}{\Delta x} (U_{I+1} - U_I),
\]

where \( \Delta x = n \Delta x \) and \( \Delta x \) are the coarse and the fine mesh resolutions, respectively. Dynamic equations for both LOW-RES and BTR are given by (6) with \( \tilde{G}_{I+1/2} \equiv 0 \). Performance of the low-resolution model is considerably worse compared to the BTR model. LOW-RES produces large errors for all stationary statistics (see Table 2). From the shape of the LOW-RES spectrum, it is also evident that the viscosity coefficient is too large in this model. Detailed comparison between the LOW-RES and BTR is presented in [1]. Since neither LOW-RES nor BTR is capable of accurately reproducing the results of the DNS, this demonstrates the necessity for the subgrid flux parametrization at the resolution \( N/n \).

Effect of noise and conditioning We also investigated the importance of stochasticity in the reduced model. In addition, we also compared performance of the GAN parametrization with and without conditions. In particular, we performed numerical experiments as above, but omit the noise term in POLY regression (denote POLY-D) and set the input noise vector in GAN model identical to zero for all times (denote GAN-D). We also test a GAN model with noise only (denote GAN-Z) by setting the conditions to zero for all times, since random variables in the input layer then undergo nonlinear transformations and “interact” with conditions. Stochasticity in GAN can be thought of as both additive and multiplicative. Therefore, the worse performance of the GAN-D parametrization compared to the POLY-D points to the potential importance of multiplicative noise term.

Note on integration time step Reduced models can potentially significantly accelerate computations since they involve fewer independent degrees of freedom an often can be simulated with a bigger time step compared to the DNS. In particular, reduced model with GAN parametrization simulated with the same time step as the DNS is three times faster than the DNS. Moreover, we also demonstrated that the reduced model can be simulated with a significantly larger time step compared to the DNS. Simulations of the reduced model in this section are performed with the time step \( 10 \Delta t = 0.1 \), that is, it is equal to ten times the time step of DNS. Thus, simulations of the reduced model are approximately 30 time faster than DNS for averaging window \( n = 16 \). The choice of the integration time step for the reduced model with GAN parametrization is not particularly related to the sampling time step. Due to the independence of the subgrid flux parametrization on past values of local averages or subgrid fluxes, any reasonable integration time step can be used in simulations as long as it is not too large to create stability issues. We also test the performance of the reduced model with GAN parametrization with larger time steps. In particular, we used integration time steps 25\( \Delta t \) and 50\( \Delta t \) for the reduced model. Data sampling time step for the training data is 50\( \Delta t \) in all cases. Summary of the one-point statistics and correlation times is presented in Table 5. Since the stationary statistics for the varied time steps are not significantly different, the results for the remaining subsections are obtained when reduced models are run at 10\( \Delta t \) with sampling done every 50\( \Delta t \).
Table 4 Variance and fourth moments of resolved modes in the DNS, full GAN parametrization, GAN parametrization without noise (GAN-D), GAN parametrization without input conditions (GAN-Z), polynomial parametrization (POLY) and polynomial parametrization without noise (POLY-D)

|                | DNS   | GAN   | GAN-Z  | GAN-D  | POLY  | POLY-D |
|----------------|-------|-------|--------|--------|-------|--------|
| Variance       | 0.0393| 0.0394| 0.0359 | 0.0373 | 0.0381| 0.0379 |
| Fourth moment  | 0.0043| 0.0044| 0.0038 | 0.0039 | 0.0046| 0.0046 |
| Correlation time| 114.26| 112.32| 119.98 | 117.34 | 115.72| 120.44 |

Fig. 7 Energy spectrum of resolved variables in simulations with increased forcing amplitude at 110% (left) and 120% (right)

3.2 Sensitivity to forcing

It is important to develop parametrizations which can be easily adapted to different parameter and forcing regimes without additional retraining. Parametrization’s ability to perform adequately with respect to changes in parameters, forcing or model resolution is an important practical factor. Practical significance of parametrization is significantly enhanced if it can perform well in a relatively wide parameter regime without refitting or retraining. Therefore, we test the performance of the GAN parametrization with respect to increase in the magnitude of the forcing. In particular, we increase the magnitude of the forcing, $A$, by 10% and 20% and compare the performance of the GAN reduced model with the DNS. We would like to emphasize that the GAN reduced model is not retrained, i.e., we use GAN reduced model trained using data generated with the original forcing amplitude. In addition, we also test the performance of the reduced model when the forcing acts on a different set of wavenumbers in Fourier space. In particular, the set of wavenumbers forced is changed to $\{4, 5, 6\}$. The GAN reduced model is not retrained in this case either.

(a) Increase in forcing amplitude Polynomial model is excluded from the comparison here because the polynomial reduced model often produced cases when the system’s energy explodes. With increased forcing amplitude, there are occurrences of local averages higher in magnitude than those in the data generated at the original forcing. This leads to incorrect estimation of subgrid fluxes by the polynomial regression and the polynomial model often diverges. The most likely reason for a poor performance of the polynomial reduced model is inadequate estimation of subgrid fluxes in regions with high gradients. When an under- or overestimation of subgrid flux corrections occurs in polynomial model, near discontinuities in regions with high gradients may not be sufficiently “tamed” by viscosity which leads to more instances of non-physical shocks or oscillations. Although this event can be hard to disprove in the case of GAN, it was observed that GAN did not encounter this problem. Numerical results for comparison of the GAN reduced model with the DNS in the regime with increased forcing are presented in Fig. 7 and Table 6. The GAN reduced model performs very well in both cases. We would like to point out that increase in forcing magnitude by 10% and 20% results in the increase in variance of the resolved variables by approximately 12% and 26%, respectively. One-point statistical quantities are reproduced perfectly by the GAN reduced model. Two-point statistics is also reproduced very well, including the kurtosis $K(s)$ and these results are not presented here only for the brevity of presentation.

(b) Change in wave numbers As in the increase in forcing amplitude, we compare the GAN reduced model with the DNS when the forcing is applied to a different set of wave numbers, particularly on $\{4, 5, 6\}$ instead
Table 5 Variance, fourth moment and correlation time of resolved modes in simulations of the reduced model with GAN parametrization with different integration time steps

| Time step | Variance | Rel. error | Fourth moment | Rel. error | Corr. time | Rel. error |
|-----------|----------|------------|---------------|------------|------------|------------|
| $10\Delta t$ | 0.0393 | 0.1% | 0.0044 | 1.5% | 112.32 | (1.7%) |
| $25\Delta t$ | 0.0390 | (0.9%) | 0.0043 | (0.2%) | 109.59 | (4.1%) |
| $50\Delta t$ | 0.0391 | (0.5%) | 0.0043 | (0.1%) | 112.36 | (1.7%) |

Fig. 8 Comparison of stationary statistical quantities in the DNS and reduced GAN model with forcing applied to wavenumbers $K = \{4, 5, 6\}$. The GAN reduced model is trained on data obtained from DNS with forcing wavenumbers $K = \{1, 2, 3\}$. Upper left: PDF, upper right: energy spectrum, lower left: ACF, lower right: lagged kurtosis of the first three wave numbers. We perform simulations of the DNS and the GAN reduced model having the same realization of forcing. Figure 8 depicts stationary statistics of the DNS and the GAN reduced model in this forcing regime. We can notice that PDE in the simulations of the GAN reduced model is broader than the PDF in the DNS. Thus, the variance of coarse variables is not reproduced as well as in previous experiments. In this regime $\text{Var}_{DNS} = 0.0069$, $\text{Var}_{GAN} = 0.0087$, Rel. error $= 26\%$. The energy spectrum is overestimated compared to the results of the DNS. We would like to point out that worse performance of the GAN reduced model might be related to the absence (or very short) of the inertial range in this regime. This will be investigated further for more realistic models. The GAN reduced model resolved 16 wavenumbers with $N/n = 32$ points in space. One can argue (from the shape of the energy spectrum) that viscosity becomes important at approximately wavenumbers 10–11. In fact, it might be a considerably more challenging regime for any parametrization, compared to forcing $K = \{1, 2, 3\}$. Nevertheless, GAN parametrization performs reasonably well in this regime. In particular, we do not encounter any unstable runs of the GAN reduced model.

Results in this section demonstrate that the GAN reduced model is able to reproduce the behavior of resolved variables in DNS with different forcing amplitudes. We would like to point out that changes in forcing amplitude resulted in substantial increase (more than 20%) in the variability of resolved variables. Therefore, the GAN reduced model proves capable of extrapolating subgrid fluxes outside of the parameter regime where it was trained.
3.3 Scalability

Another important practical property of parametrizations is scalability. Scalability reflects the ability to utilize the same parametrization without retraining in a reduced model with different spatial (and possibly temporal) resolution. Typically, if the spatial resolution is changed (e.g., coarser or finer mesh than originally designed is used), we would like to use the same parametrization scheme (possibly properly scaled) to perform simulations of reduced models on a new mesh without retraining or refitting. Resolution decrease is motivated by necessity to run fast crude simulations in, for example, ensemble predictions, and the increase in resolution is motivated by technological advances and future increase in computational capabilities. Thus, in the future it should be possible to run reduced models with higher spatial resolution, and it is important to verify that stochastic parametrizations utilized in reduced models can adequately respond without retraining.

Here we demonstrate that simple linear scaling in GAN parametrization is sufficient to adapt GAN reduced model with respect to changes in mesh size and perform simulations with a different spatial resolution. Of course, we cannot expect that the same parametrization would perform adequately if the spatial resolution is changed drastically. Therefore, we test several cases—when the mesh becomes twice finer and two and four times coarser. In particular, we consider the following setup. Recall that we denote the averaging window as $n$ (see (2)). If the averaging window is changed to $n'$, then we use a linear scaling and the generated subgrid fluxes (output of GAN) is multiplied by a factor $n/n'$. This linear scaling is motivated by adequate results in the context of stochastic mode reduction strategy applied to finite difference discretization of the Burgers–Hopf equation [15,16] and the shallow water model [52]. However, it is important to note that the scaling might be system-specific or, at least, it might depend on the dimension of the problem. It is rather difficult to obtain the scaling factor using either analysis or direct numerical simulations, since it is not clear how to relate input and output of the generator for different $n$. This is a consequence of the fact that typical magnitude of both the coarse variables and subgrid fluxes changes as $n$ is varied. Still, we tested the extent of applicability of linear scaling because of its ease in implementation and its performance in reduced systems derived using stochastic mode reduction mentioned earlier.

The reference setup in Sect. 3.1 corresponds to $n = 16$ and here we consider cases $n' = 4$, $n' = 8$ and $n' = 32$. Second and fourth moments in simulations with $n = 16$ and $n' = 4, 8, 32$ are presented in Table 7. In addition, in this Table we also present results for simulations with non-scaled (NS) GAN parametrization. The scaling introduced earlier improves the performance of the reduced model for all cases $n' = 4, 8, 32$. Numerical results for $n' = 4$ and $n' = 8$ are very close to the results for $n = 16$ and, thus, scaling of the

| Table 6 | Variance, fourth moment and correlation time of resolved modes in simulations of the reduced model with GAN parametrization with increased forcing by 10% and 20% |
|---------|--------------------------------------------------|
|         | Variance | Fourth moment |
| DNS (10%) | 0.0441 | 0.0055 |
| GAN (10%) | 0.0439 | 0.0055 |
| DNS (20%) | 0.0496 | 0.0069 |
| GAN (20%) | 0.0494 | 0.0071 |

| Table 7 | Variance and fourth moments of resolved modes in direct numerical simulations and simulations of the reduced model with different resolutions and GAN parametrization |
|---------|--------------------------------------------------|
|         | Variance | Rel. error | Fourth moment | Rel. error |
| DNS ($n' = 4$) | 0.03934 | (2.8%) | 0.004451 | (3.9%) |
| GAN-S ($n' = 4$) | 0.03991 | (1.3%) | 0.004490 | (2.9%) |
| DNS ($n' = 8$) | 0.03993 | | 0.004509 | |
| GAN-S ($n' = 8$) | 0.03842 | (3.8%) | 0.004253 | (5.7%) |
| DNS ($n = 16$) | 0.03934 | | 0.004385 | (2.7%) |
| GAN ($n = 16$) | 0.03936 | 0.1% | 0.004347 | 1.5% |
| DNS ($n' = 32$) | 0.03616 | | 0.003689 | |
| GAN-S ($n' = 32$) | 0.05928 | 63.9% | 0.007951 | 115.7% |
| GAN-NS ($n' = 32$) | 0.03718 | 2.8% | 0.004006 | 8.6% |

GAN-S and GAN-NS denote scaled and non-scaled version of the GAN parametrization trained with $n = 16$, respectively.
GAN parametrization does not produce a large effect. On the other hand, numerical results for $n' = 32$ clearly indicate that scaling of the GAN parametrization improves the performance of the reduced model. This is particularly important for simulations with the coarser resolution $n' = 32$. We would like to point out that variance of reduced variables does not increase significantly for cases $n' = 4$ and $n' = 8$ (only 2.7% and 1.5% relative increase, respectively), but for $n' = 32$ variance is reduced by approximately 8% compared with $n = 16$. Also, with wider averaging window in the case of $n' = 32$, the regime becomes quite different. Since the number of resolved wavenumbers is reduced for $n' = 32$, the inertial range becomes considerably shorter (or maybe even disappears). This is likely the reason why the reduced model for $n' = 32$ is more sensitive to the scaling of GAN parametrization compared to simulations with $n' = 4$ and $n' = 8$. Two-point stationary statistics are also reproduced very well by reduced models with scaled GAN parametrization. Overall, there is a good agreement between the direct numerical simulations and simulations of the reduced model with scaled GAN parametrization. In addition to results presented in Table 7, utilizing scaled GAN parametrization also improves the shape of the spectrum for high wavenumbers for $n' = 4$ and $n' = 8$ (see [1]). Simulations with unscaled GAN parametrization with $n' = 32$ completely fail to reproduce stationary spectrum and probability distribution of resolved variables (see [1] for details) and these results are not presented here only for the brevity of presentation.

4 Conclusions

A stochastic subgrid flux parametrization is developed here using conditional GAN. The overall approach is similar to the large Eddy simulations modeling and we use a “box” filter in physical space to define coarse variables. We demonstrated that Generative Adversarial Networks can be successfully trained using the Wasserstein metric to reproduce subgrid corrections for fluxes of resolved variables. The WGAN version of the Adversarial Networks significantly improves conversion during the training process. In addition, to estimate the subgrid flux between two neighboring cells, the GAN is conditioned on the current value of the coarse variables in these two cells. This conditioning is essential for reproducing the subgrid fluxes. Since, resolved variables are defined here as local spatial averages and conditioning for GAN is necessary only on their values in neighboring cells, the reduced model is local in space, i.e., only two neighboring cells are coupled by a GAN parametrization which estimates the subgrid flux between these two cells. Thus, if the model is non-homogeneous in space, the GAN parametrization can potentially be trained separately for different spatial regions.

Our aim here is to reproduce statistical properties of long-term turbulent simulations. In particular, we compare spectra, moments, and two-point temporal statistical quantities. We would like to point out that we simulate the reduced model for a long time and the reduced model does not blow up; all runs of the reduced model produce stable trajectories. The reduced model with GAN parametrization reproduces statistical properties of the full model quite well. In addition, we also performed a limited number of non-equilibrium simulations where we compared statistical properties of ensembles of solutions starting with a particular initial condition. However, performance of the GAN parametrizations degrades as initial conditions are taken further away from equilibrium (i.e., initial conditions are more than one order of magnitude larger than the standard deviation of the stationary data used for training). Overall, the GAN parametrization performs very well in a non-equilibrium setting producing timescales for the changes in the mean and variance of ensembles comparable with direct numerical simulations. This is reported in [1]. The Burgers–Hopf model used here has many common features with more realistic equations in fluid dynamics. In particular, Burgers–Hops equation exhibits solutions with large gradients, which are also present in, for instance, shallow water model. Applications of the proposed approach to more realistic models in fluid dynamics will be considered in subsequent papers. In foresight, extending to more complex systems may require introducing additional conditions and/or enforcing physical constraints. For instance, with the GAN formulation considered here, no physical constraints or qualitative behavior of the system is incorporated in either the network or in the training. It might be necessary for other systems to include various constrains explicitly in the network to improve the performance of the GAN parametrization, produce stable results, or accelerate computations. This can be achieved by including additional penalty terms in the loss functions, introducing additional layers which enforce symmetries or constraints in the system, etc. This question will be the subject of future research.

In addition, we investigate numerically the performance of the GAN parametrization with respect to changes in forcing and different spatial resolutions. To this end, we utilize the original parametrization (without retraining) to simulate reduced models with 10% and 20% increase in forcing. The GAN parametrization
performs remarkably well in these regimes. We also introduce linear scaling for the GAN parametrization to compute reduced models with different spatial resolutions. We demonstrate that this linear scaling is essential for the reduced model to accurately reproduce statistical properties of solutions. We would like to emphasize that the GAN parametrization is not retrained on the new data (e.g., simulations with increased forcing or changed resolution). Therefore, we demonstrate that the reduced models with the GAN parametrization can be used in a wide range of regimes without retraining the neural network. We would like to point out that empirical parametrizations (e.g., polynomial regression used here) often exhibit considerable difficulties when applied outside of the training regime (e.g., higher energy due to increased forcing). However, GAN parametrization performs extremely well in such test cases which makes GAN parametrization superior to many empirical approaches considered in the literature (e.g., polynomial regression, Markov chain parametrization, etc.)

The approach developed here is particularly suitable for finite volume (or finite difference) discretizations of partial differential equations since resolved variables are defined as local spatial averages and subgrid fluxes are represented using corresponding averages of the right-hand side of the equation. Although a particular “box” filter is used here to define resolved variables, we expect that our approach should be applicable to other types of filters as well as long as spatial filters induce a weak scale separation between the resolved and unresolved degrees of freedom. This is a typical situation often exploited to derive effective equations for reduced variables in LES-type models. We believe that this scale separation is not essential for the applicability of the GAN parametrization, but it quite useful for accelerating simulations of the effective model.

In this paper we demonstrate applicability of GANs for parametrizing subgrid scales in a prototype nonlinear model which exhibits complex turbulent behavior. Thus, this approach stands ready to be applied in more complex models of fluid dynamics. This will be investigated in subsequent papers.

Acknowledgements The model code for this project is publicly available online at github in the repository Burgers_GAN and on zinodo [1](https://doi.org/10.5281/zenodo.3952945).

References

1. Alcala, J.S.: Subgrid-scale parametrization of unresolved processes. Ph.D. Thesis, University of Houston (in preparation)
2. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. arXiv:1701.04862 (2017)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv:1701.07875v3 (2017)
4. Berner, J., Achatz, U., Batte, L., Bengtsson, L., de la Camara, A., Christensen, H.M., Colangeli, M., Coleman, D.R.B., Crommelin, D., Dolaptchiev, S.I., Franzke, C.L.E., Friederichs, P., Imkeller, P., Jarvinen, H., Juricke, S., Kitsios, V., Lott, F., Lucarini, V., Mahajan, S., Palmer, T.N., Penland, C., Sakradzija, M., von Storch, J.S., Weisheimer, A., Weniger, M., Williams, P.D., Yano, J.I.: Stochastic parametrization: towards a new view of weather and climate models. Bull. Am. Meteorol. Soc. 98, 565–588 (2017)
5. Brenowitz, N.D., Bretherton, C.S.: Spatially extended tests of a neural network parametrization trained by coarse-graining. J. Adv. Model. Earth Syst. 11, 2728–2744 (2019)
6. Burgers, J.: A mathematical model illustrating the theory of turbulence. Adv. Appl. Mech. 1, 171–179 (1948)
7. Chattopadhyay, A., Nabizadeh, E., Hassanzadeh, P.: Analog forecasting of extreme-causing weather patterns using deep learning. arXiv:1907.11617 (2019)
8. Chekhnov, A., Yakhot, V.: Kolmogorov turbulence in a random-force-driven Burgers equation: anomalous scaling and probability density functions. Phys. Rev. E 52(5), 5681–5684 (1995)
9. Chorin, A.J., Hald, O.H.: Stochastic Tools in Mathematics and Science. Springer, Berlin (2009)
10. Chorin, A.J., Hald, O.H., Kupferman, R.: Optimal prediction and the Mori-Zwanzig representation of irreversible processes. Proc. Natl. Acad. Sci. 97(7), 2968–2973 (2000). https://doi.org/10.1073/pnas.97.7.2968
11. Chorin, A.J., Hald, O.H., Kupferman, R.: Optimal prediction with memory. Physica D Nonlinear Phenomena 166(3), 239–257 (2002). https://doi.org/10.1016/S0167-2789(02)00446-3
12. Chorin, A.J., Lu, F.: Effects of stochastic parametrization in the Lorenz 96 system. Proc. Natl. Acad. Sci. USA 112(32), 9804–9809 (2015)
13. Crommelin, D., Vanden-Eijnden, E.: Fitting timeseries by continuous-time Markov chains: a quadratic programming approach. J. Comp. Phys. 217, 782–805 (2006)
14. Crommelin, D.T., Vanden-Eijnden, E.: Subgrid-scale parametrization with conditional Markov chains. J. Atmos. Sci. 65, 2661–2675 (2008)
15. Dolaptchiev, S.I., Achatz, U., Timofeyev, I.: Stochastic closure for local averages in the finite-difference discretization of the forced Burgers equation. Theor. Comput. Fluid Dyn. 27(3–4), 297–317 (2013)
16. Dolaptchiev, S.I., Timofeyev, I., Achatz, U.: Subgrid-scale closure for the inviscid Burgers–Hopf equation. Commun. Math. Sci. 11(3), 757–777 (2013)
17. Franzke, C., Majda, A.: Low-order stochastic mode reduction for a prototype atmospheric gcm. J. Atmos. Sci 63, 457–479 (2006)
18. Franzke, C., Majda, A., Vanden-Eijnden, E.: Low-order stochastic mode reduction for a realistic barotropic model climate. J. Atmos. Sci 62, 1722–1745 (2005)
19. Frederiksen, J.S., Davies, A.G.: Eddy viscosity and stochastic backscatter parameterizations on the sphere for atmospheric circulation models. J. Atmos. Sci. 54, 2475–2492 (1997). https://doi.org/10.1175/1520-0442(1997)
20. Frederiksen, J.S., Okane, T.J., Zdilkeri, M.J.: Subgrid modelling for geophysical flows. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 371(1982), 201266–201266 (2012). https://doi.org/10.1098/rsta.2012.0166
21. Gagne II, D.J., Christensen, H.M., Subramanian, A.C., Monahan, A.H.: Machine learning for stochastic parameterization: generative adversarial networks in the Lorenz ‘96 model. J. Adv. Model. Earth Syst. (2020). arXiv:1909.04711. https://doi.org/10.1029/2019MS001896
22. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Adv. Neural Inf. Process. Syst. 2672–2680 (2014). arXiv:1406.2661
23. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. arXiv:1704.00028 v3 (2017)
24. Hasselman, K.: Stochastic climate models. Part I. Theory. Tellus 28, 473–485 (1976)
25. Hijon, C., Espanol, P., Vanden-Eijnden, E., Delgado-Buscalioni, R.: Mori–Zwanzig formalism as a practical computational tool. Faraday Discuss. 144, 301–302 (2010)
26. Timofeyev, I.: Markov chain stochastic parametrizations of essential variables. SIAM Mult. Mod. Simul. 8(5), 2079–2096 (2010)
27. Kravtsov, S., Kondrashov, D., Ghil, M.: Multilevel regression modeling of nonlinear processes: derivation and applications. J. Climate 18, 4404–4424 (2005)
28. Kwastriok, F.: Data-based stochastic subgrid-scale parameterisation: an approach using cluster weighted modeling. Philos. Trans. R. Soc. A 370, 1061–1086 (2012)
29. L. Yang, D.Z., Karniadakis, G.E.: Physics-informed generative adversarial networks for stochastic differential equations. SIAM J. Sci. Comput. 42(1), A292–A317 (2020)
30. LeVeque, R.J.: Numerical Methods for Conservation Laws, 2nd edn. Birkhäuser, Basel (1992)
31. Lu, F., Lin, K.K., Chorin, A.J.: Data-based stochastic model reduction for the Kuramoto–Sivashinsky equation. Physica D 370, 46–57 (2017)
32. Majda, A., Timofeyev, I., Vanden-Eijnden, E.: A mathematical framework for stochastic climate models. Commun. Pure Appl. Math. 54, 891–974 (2001)
33. Majda, A., Timofeyev, I., Vanden-Eijnden, E.: Systematic strategies for stochastic mode reduction in climate. J. Atmos. Sci. 60, 1705–1722 (2003)
34. Majda, A., Timofeyev, I., Vanden-Eijnden, E.: Stochastic models for selected slow variables in large deterministic systems. Nonlinearity 19, 769–794 (2006)
35. Moser, R.D., Haering, S.W., Yalla, G.R.: Statistical properties of subgrid-scale turbulence models. Annu. Rev. Fluid Mech. 53(1), 1–22 (2021). https://doi.org/10.1146/annurev-fluid-060420-023755
36. Nimsaila, K., Timofeyev, I.: Markov chain stochastic parametrizations of essential variables. SIAM Mult. Mod. Simul. 8(5), 2079–2096 (2010)
37. OGorman, P.A., Dwyer, J.G.: Using machine learning to parameterizemoist convection: potential for modeling of climate, climate change, and extreme events. J. Adv. Model. Earth Syst. 10, 2548–2563 (2018)
38. Palmer, T., et al.: Stochastic parametrization and model uncertainty. Technical Report No. 596. ECMWF, Reading UK (2009)
39. Resseguier, V., Li, L., Jouan, G., Dérian, P., Mémin, E., Bertrand, C.: New trends in ensemble forecast strategy: uncertainty quantification for coarse-grid computational fluid dynamics. Arch. Comput. Methods Eng. 1–82 (hal–02558016) (2020)
40. Shutts, G., Palmer, T.: Convective forcing fluctuations in a cloud-resolving model: relevance to the stochastic parameterization problem. J. Clim. 20, 187–202 (2007)
41. Verkley, W.T.M.: A maximum entropy approach to the problem of parametrization. Q. J. R. Meteorol. Soc. 137, 1872–1886 (2011)
42. Verkley, W.T.M., Kalverla, P.C., Severijns, C.A.: A maximum entropy approach to the parametrisation of subgrid processes in two-dimensional flow. Q. J. R. Meteorol. Soc. 142(699), 2273–2283 (2016). https://doi.org/10.1002/qj.2817
43. Verkley, W.T.M., Kalverla, P.C., Severijns, C.A.: A maximum entropy approach to the parametrisation of subgrid processes in two-dimensional flow. Q. J. R. Meteorol. Soc. 142, 2273–2283 (2016)
44. Vlachas, P.R., Pathak, J., Hunt, B.R., Sapsis, T.P., Girvan, M., Ott, E., Koumoutsakos, P.: Forecasting of spatio-temporal chaotic dynamics with recurrent neural networks: a comparative study of reservoir computing and backpropagation algorithms. arXiv:1910.05266 (2019)
45. Wang, Z., She, Q., Ward, T.E.: Generative adversarial networks: a survey and taxonomy. arXiv:1906.01529 (2019)
46. Wilks, D.: Effects of stochastic parameterization in the Lorenz 96 system. Q. J. R. Meteorol. Soc. 131, 389–407 (2005)
47. Wouters, J., Dolaptchiev, S.I., Lucarini, V., Achatz, U.: Parameterization of stochastic multiscale trips. Nonlinear Process. Geophys. 23, 435–445 (2016)
48. Wu, J.L., Kashinath, K., Albert, A., Prabhat, D.C., Xiao, H.: Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. J. Comput. Phys. 406, 109–209 (2020)
49. Yang, L., Zhang, D., Karniadakis, G.E.: Physics-informed generative adversarial networks for stochastic differential equations. SIAM J. Sci. Comput. 42(1), A292–A317 (2020). https://doi.org/10.1137/18M1225409
50. Zabusky, N.J., Kruskal, M.D.: Interaction of solitons in a collisionless plasma and the recurrence of initial states. Phys. Rev. Lett. 15, 240–243 (1965)
51. Zacharuk, M., Dolaptchiev, S.I., Achatz, U., Timofeyev, I.: Stochastic subgrid-scale parameterization for one-dimensional shallow water dynamics using stochastic mode reduction. Q. J. R. Meteorol. Soc. 144(715), 1975–1990 (2018)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.