JAMES | Journal of Advances in Modeling Earth Systems®

RESEARCH ARTICLE
10.1029/2021MS002490

Key Points:

- Artificial Neural Networks (ANNs) can parameterize exact closure terms of a Variational Multiscale method without a priori model form assumptions
- The ANN parameterization outperforms state-of-the-art schemes in offline tests
- Instabilities in the numerical model comprising the ANN parameterization are amendable through data set enrichment

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:
S. J. Hulshoff,
s.j.hulshoff@tudelft.nl

Citation:
Janssens, M., & Hulshoff, S. J. (2022). Advancing Artificial Neural Network parameterization for atmospheric turbulence using a variational multiscale model. *Journal of Advances in Modeling Earth Systems*, *14*, e2021MS002490. https://doi.org/10.1029/2021MS002490

Received 12 FEB 2021
Accepted 1 OCT 2021

Advancing Artificial Neural Network Parameterization for Atmospheric Turbulence Using a Variational Multiscale Model

M. Janssens[1,2] **and S. J. Hulshoff**[1]

[1]Faculty of Aerospace Engineering, TU Delft, Delft, The Netherlands, [2]Meteorology and Air Quality Department, Wageningen University & Research, Wageningen, The Netherlands

Abstract Data-driven parameterizations offer considerable potential for improving the fidelity of General Circulation Models. However, ensuring that these remain consistent with the governing equations while still producing stable simulations remains a challenge. In this paper, we propose a combined Variational-Multiscale (VMS) Artificial Neural Network (ANN) discretization which makes no a priori assumptions on the model form, and is only restricted in its accuracy by the precision of the ANN. Using a simplified problem, we demonstrate that good predictions of the required closure terms can be obtained with relatively compact ANN architectures. We then turn our attention to the stability of the VMS-ANN discretization in the context of a single implicit time step. It is demonstrated that the ANN parameterization introduces nonphysical solutions to the governing equations that can significantly affect or prevent convergence. We show that enriching the training data with nonphysical states from intra-time step iterations is an effective remedy. This indicates that the lack of representative ANN-induced errors in our original, exact training inputs underpin the observed instabilities. In turn, this suggests that data set enrichment might aid in resolving instabilities that develop over several time steps.

Plain Language Summary Computer models of weather and climate have coarse resolutions, which prevent them from accurately predicting the effect of small atmospheric motions, for instance in low-lying clouds, on the global climate. Recent studies indicate that improvements might be had by using models for the small motions that are informed purely by data, for example Artificial Neural Network (ANN) models. However, capitalizing on this potential is challenging in practice, since ANNs can introduce instabilities to the numerical model for larger-scale motions. In this study, we zero in on these instabilities by introducing a model framework in which prediction accuracy is only limited by an ANN's ability to predict the net influence of small atmospheric motions. We contextualize this framework by contrasting it to more commonly used approaches. Using a simple test case based on the motions underpinning the development of low clouds, we illustrate that standard training procedures do not prepare ANN models for their interaction with the rest of the numerical model, and that this can contribute to numerical instability. We demonstrate an effective remedy for a specific type of instability, and suggest how this technique might also be used to treat other types of instabilities prompted by ANN models.

1. Introduction

Intercomparison studies of existing atmospheric General Circulation Models (GCMs) exhibit a large spread in their predictions of atmospheric CO_2 concentration at which a 2 K temperature rise with respect to pre-industrial times is reached (Boucher et al., 2013). This uncertainty imposes a considerable cost on society (Hope, 2015).

The largest contributor to this uncertainty concerns the response of low clouds to warming (Dufresne & Bony, 2008), as their impact is large (Boucher et al., 2013), but the turbulent phenomena that drive much of their evolution lie far below the resolutions that computational limits will allow GCMs to resolve in the coming decades (Schneider, Teixeira, et al., 2017). Such clouds are currently approximated by phenomenological unresolved scales models, "parameterizations", of considerably lower fidelity than the resolved simulation.

It has been proposed that improvements might be had by considering data-driven parameterizations (Schneider, Lan, et al., 2017). Over the past few years, several promising flavors of such models have been developed (Duraisamy et al., 2019; Kochkov et al., 2021; Maulik & San, 2017; Yuval & O'Gorman, 2020). Among these flavors, replacing the original parameterization by Artificial Neural Networks (ANNs) appears particularly encouraging for improving the accuracy of both idealized fluid flow calculations (Beck et al., 2019; Guan et al., 2021; Park & Choi, 2021; Stoffer et al., 2020) and global atmospheric models (Brenowitz & Bretherton, 2018; Krasnopolsky et al., 2013; Rasp et al., 2018; Yuval et al., 2021). However, ensuring that an ANN adheres to the laws of physics and returns stable, generalizable simulations remains an appreciable challenge (Gentine et al., 2018). Additionally, embedding machine learning frameworks in GCMs does not resolve the conceptual inconsistencies of model formulations that prevent the correct interaction between the smaller scales of clouds and the larger scales of climate (e.g., Majda & Grooms, 2014). The effort of discovering and treating the drawbacks that currently prevent data-driven models from competing with the state of the art is just beginning.

We contribute to this effort by considering ANN approximations for the exact closure terms that couple the resolved and unresolved scales of a discretized mathematical atmospheric model. These terms arise naturally as projections of unresolved processes onto a discrete basis when analyzed through the lens of a Variational Multiscale (VMS) framework (Hughes et al., 2018). In contrast to previous work, a numerical VMS model with an ANN representing its closure terms makes no a priori assumptions on the parameterization's modeling form and is only restricted in its accuracy of reconstructing the resolved scales by the ability of an ANN to approximate the closure terms. It is therefore a rather general and clear platform to assess the ultimate potential of ANN parameterization, and to search for sources of the instabilities that it instils in numerical simulations.

In this paper, we therefore introduce a combined VMS-ANN modeling framework, where the resolved scales of the VMS model drive ANN approximations of exactly measured closure terms at each time step of the numerical model. In the context of dry, statistically stationary convective boundary layer turbulence, we train and test relatively simple ANN structures outside a time stepping loop (offline), where they promise excellent potential. However, even for such a simple flow, our VMS-ANN encounters a loss of energy conservation over several time steps in forward simulations of the model problem (an "online" model evaluation setting), similar to that found in studies of similarly simple (e.g., Beck et al., 2019; Stoffer et al., 2020) and more complex (e.g., Brenowitz & Bretherton, 2018; Krasnopolsky et al., 2013) situations. In contrast to those studies, which employed explicit time marches, we investigate the use of an implicit method. For the numerical model, this consistently allows the ANN to account for the time-propagation of discretization error, and could unlock the use of longer, stable time steps. However, this choice will lead to a second, previously undescribed mode of instability. By exposing the reasons underpinning this instability and indicating a direction for improving it, we will suggest a broader strategy for making data-driven parameterizations viable for online use.

In all, this paper has three objectives. First, Section 2 very generally outlines a coupled VMS-ANN model and discusses its relation to several other modeling approaches. Second, we define our minimal test case for the model in Section 3, and use it to illustrate the approach's potential to outperform state-of-the-art VMS model closures in an offline setting in Section 4. Finally, we expose and amend instabilities within an implicit time step in Section 5, and discuss how lessons from this example may pave the way for efficient, stable online running with data-driven parameterizations in Section 6.

2. Variational Multiscale-Artificial Neural Network Model

2.1. Variational Multiscale Model

The framework of VMS modeling offers a perspective on how to develop stabilized spectral and finite element methods for problems with a large range of impactful scales, such as atmospheric dynamics (Hughes, 1995; Hughes et al., 1998). It has been successfully applied to various aerodynamic (Hughes et al., 2000) and atmospheric (Marras et al., 2013a, 2013b) problems. We employ the framework here, as it allows for a particularly clear exposition of what is required from an unresolved scales model.

High-resolution GCMs require a mathematical model appropriate to phenomena of global scale (e.g., Klein, 2010). Initial studies aimed at showing the potential of ANN parameterizations (e.g., Krasnopolsky et al., 2013; Rasp et al., 2018) and more recent work aimed at uncovering their effects on large-scale processes (Brenowitz et al., 2020) therefore concentrated on global, fully process-equipped models. While these studies successfully accomplished their objective, their highly complex models are not ideal to advance our understanding of how ANN parameterizations fundamentally interact with numerical models. Therefore, we propose to start at the other end, by focusing on the free turbulence in the atmospheric boundary layer that underpins the development of low clouds. While ignoring many processes and scales, this is arguably the simplest imaginable setting relevant for studying cloud parameterization for GCMs. Therefore, we consider it a suitable testbed to begin building an understanding of an ANN parameterization's basic behavior.

More precisely, we consider a fluid governed by the Boussinesq equations without moisture, Coriolis forces or molecular diffusion in three spatial dimensions, on a horizontally homogeneous domain Ω:

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}\left(u_j u_i\right) + \frac{\partial \pi}{\partial x_i} - g\frac{\theta - \theta_0}{\theta_0}\delta_{i3} = 0 \tag{2}$$

$$\frac{\partial \theta}{\partial t} + \frac{\partial}{\partial x_j}\left(\theta u_j\right) = S_\theta \tag{3}$$

Where u_i are the three components of velocity, θ is (dry) potential temperature, θ_0 is a reference temperature, π are pressure fluctuations normalised by a reference density ρ_0, g is acceleration due to gravity and S_θ is a (radiative) source. The conditions imposed on the spatial boundary of Ω, Γ, and on the initial time boundary, are discussed in Section 3.

One can derive a variational multiscale model for solving equations such as these by following e.g., Hughes et al. (2000). Here, we will only give a minimal sketch of this derivation, and refer the interested reader to for example, (Codina et al., 2018; Hughes et al., 2018) for more detailed overviews. We begin by decomposing the solution $\boldsymbol{u} = \left[\pi, u_i, \theta\right]$ into a sum of known basis functions $\boldsymbol{\psi} = \left[\psi_\pi, \psi_{u_i}, \psi_\theta\right]$ with unknown amplitudes $\boldsymbol{a_i}$:

$$\boldsymbol{u} = \sum_{i=0}^{m} \boldsymbol{\psi}_i \boldsymbol{a}_i \tag{4}$$

Here, m defines the potentially infinite number of solution components needed to represent the solution exactly. To solve for the unknown $\boldsymbol{a_i}$, we pose m "weak" equations, constructed by inserting Equation 4 into Equations 1–3, multiplying the equation set with m weighting functions of choice, and integrating each resulting equation over Ω (Equations 1–3 themselves are commonly referred to as "strong" equations). Each term in a weak equation is thus an inner product, denoted here as $\left(\cdot,\cdot\right)_\Omega$.

To obtain a square system, we choose the set of weighting functions to be equal to the set of solution basis functions, $\boldsymbol{\psi}$. Hence, both the weighting functions and solution components exist in the function space $\boldsymbol{\psi} \in \mathcal{V}$, which is defined such that the functions in this space can be integrated and differentiated twice, while always returning finite values. With these definitions in place, we can pose Equations 1–3 as the following semi-discrete, variational problem:

Find $\boldsymbol{u} \in \mathcal{V}$ such that $\forall \boldsymbol{\psi} \in \mathcal{V}$:

$$A\left(\boldsymbol{\psi}, \boldsymbol{u}\right) = 0 \tag{5}$$

$$B_1\left(\boldsymbol{\psi}, \boldsymbol{u}\right) + B_2\left(\boldsymbol{\psi}, \boldsymbol{u}, \boldsymbol{u}\right) = 0 \tag{6}$$

$$C_1\left(\boldsymbol{\psi}, \boldsymbol{u}\right) + C_2\left(\boldsymbol{\psi}, \boldsymbol{u}, \boldsymbol{u}\right) = 0 \tag{7}$$

Where:

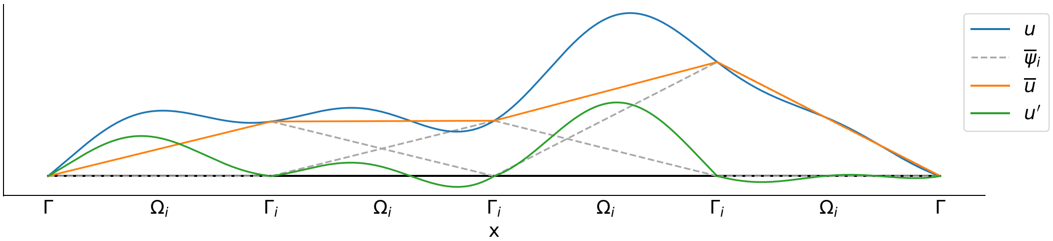

Figure 1. Simplified representation of a one-dimensional solution $u(x)$ and its lower dimensional representation $\overline{u}$, constructed by projecting u onto piecewise linear bases $\overline{\psi}_i$ over finite elements with domain Ω_i and boundary Γ_i, such that $\overline{u}(\Gamma_i) = u(\Gamma_i)$ and the projection is nodally exact. This projection defines u' as illustrated.

$$\begin{cases} A(\boldsymbol{\psi},\boldsymbol{u}) &= -\sum_{i=1}^{m}\left(\frac{\partial\psi_{\pi}}{\partial x_j},u_j\right)_{\Omega_i} \\ B_1(\boldsymbol{\psi},\boldsymbol{u}) &= \left(\psi_{u_i},\frac{\partial u_i}{\partial t}\right)_{\Omega} - \sum_{i=1}^{m}\left(\frac{\partial\psi_{u_i}}{\partial x_i},\pi\right)_{\Omega_i} - \left(\psi_{u_i},g\frac{\theta-\theta_0}{\theta_0}\delta_{i,3}\right)_{\Omega} \\ B_2(\boldsymbol{\psi},\boldsymbol{u},\boldsymbol{u}) &= -\sum_{i=1}^{m}\left(\frac{\partial\psi_{u_i}}{\partial x_j},u_iu_j\right)_{\Omega_i} \\ C_1(\boldsymbol{\psi},\boldsymbol{u}) &= \left(\psi_{\theta},\frac{\partial\theta}{\partial t}\right)_{\Omega} \\ C_2(\boldsymbol{\psi},\boldsymbol{u},\boldsymbol{u}) &= -\sum_{i=1}^{m}\left(\frac{\partial\psi_{\theta}}{\partial x_j},u_j\theta\right)_{\Omega_i} \end{cases}$$

In the above, we have integrated several terms by parts, introduced integration over the sub-domain Ω_i on which $\boldsymbol{\psi}_i$ is non-zero, and it is assumed that $\boldsymbol{\psi}$ satisfies the boundary conditions to be imposed. Note that we elect to express Equations 5–7 in terms of "bilinear" forms which depend linearly on $\boldsymbol{u}$ (e.g., $A(\boldsymbol{\psi},\boldsymbol{u})$), and "trilinear" forms, which depend on non-linear products of the solution vector (e.g., $B_2(\boldsymbol{\psi},\boldsymbol{u},\boldsymbol{u})$). While hiding several details of the equations themselves, this notation emphasizes the scale interactions in $\boldsymbol{\psi}$ and $\boldsymbol{u}$ that we wish to focus on, and keeps the equations compact throughout the paper.

Attaining exact solutions for **u** requires solving Equations 5–7 for an infinite number of trial solutions weighting an infinite number of basis functions, i.e., $m \to \infty$. To make the problem tractable, we will only solve for a finite number, n_{el}, of solution components. We will contain these finite degrees of freedom in the space $\overline{\mathcal{V}}$, which satisfies the direct sum decomposition $\oplus$ of $\mathcal{V}$:

$$\mathcal{V} = \overline{\mathcal{V}} \oplus \mathcal{V}' \tag{8}$$

In this context, $\overline{\mathcal{V}}$ is the space of resolved, "large" scales, while $\mathcal{V}'$ contains unresolved, "small" scales. This scale separation identifies the resolved solution $\overline{\boldsymbol{u}}$, its degrees of freedom $\overline{\boldsymbol{a}_i}$ and its basis $\overline{\boldsymbol{\psi}}$:

$$\overline{\boldsymbol{u}} = \sum_{i=0}^{n_{el}} \overline{\psi}_i \overline{\boldsymbol{a}}_i \tag{9}$$

For a given one-dimensional state u, this definition of $\overline{u}$ directly identifies u'. This is illustrated in Figure 1, where $\overline{u}$ is constructed using piecewise linear $\overline{\psi}$. It is important to note that when computing coarse-mesh solutions, the objective is to compute $\overline{\boldsymbol{u}}$, the desired projection of $\boldsymbol{u}$ onto the discrete space, rather than $\boldsymbol{u}$ itself. For the current work, the desired $\overline{\boldsymbol{u}}$ was chosen as the nodally exact projection of $\boldsymbol{u}$, although other projections could be used.

By also projecting Equations 5–7 onto $\overline{\mathcal{V}}$ and $\mathcal{V}'$, we arrive at a finite-dimensional set of resolved scales equations for $\overline{\boldsymbol{u}}$ (Equations 10–12) and an infinite-dimensional set of unresolved scales equations for $\boldsymbol{u}'$ (Equations 13–15). Our numerical model's objective will then be reduced to:

Find $\overline{\boldsymbol{u}} \in \overline{\mathcal{V}}$ such that $\forall \overline{\boldsymbol{\psi}} \in \overline{\mathcal{V}}$:

$$\underbrace{A\left(\bar{\boldsymbol{\psi}},\bar{\boldsymbol{u}}\right)}_{\text{i}}+\underbrace{A\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}'\right)}_{\text{iii}}=0 \tag{10}$$

$$\underbrace{B_1\left(\bar{\boldsymbol{\psi}},\bar{\boldsymbol{u}}\right)+B_2\left(\bar{\boldsymbol{\psi}},\bar{\boldsymbol{u}},\bar{\boldsymbol{u}}\right)}_{\text{i}}+\underbrace{B_2\left(\bar{\boldsymbol{\psi}},\bar{\boldsymbol{u}},\boldsymbol{u}'\right)+B_2\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}',\bar{\boldsymbol{u}}\right)}_{\text{ii}}+\underbrace{B_1\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}'\right)+B_2\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}',\boldsymbol{u}'\right)}_{\text{iii}}=0 \tag{11}$$

$$\underbrace{C_1\left(\bar{\boldsymbol{\psi}},\bar{\boldsymbol{u}}\right)+C_2\left(\bar{\boldsymbol{\psi}},\bar{\boldsymbol{u}},\bar{\boldsymbol{u}}\right)}_{\text{i}}+\underbrace{C_2\left(\bar{\boldsymbol{\psi}},\bar{\boldsymbol{u}},\boldsymbol{u}'\right)+C_2\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}',\bar{\boldsymbol{u}}\right)}_{\text{ii}}+\underbrace{C_1\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}'\right)+C_2\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}',\boldsymbol{u}'\right)}_{\text{iii}}=0 \tag{12}$$

Three types of terms appear in Equations 10–12: (i) interactions between resolved scales, (ii) direct interactions between resolved and unresolved scales, projected on the resolved basis $\bar{\boldsymbol{\psi}}$ and (iii) projections of pure unresolved scale interactions on the resolved scales. In the language of the Large Eddy Simulation community, terms (ii) and (iii) can be interpreted as analogs of the Cross and Reynolds stresses of the filtered momentum equations. Note that *if* one could reproduce these terms exactly, one could reproduce $\bar{\boldsymbol{u}}$ exactly. Hence, in spite of having discretized the equations, we have yet to introduce any approximation. In practice, this will of course be necessary, as terms (ii) and (iii) can only be identified once the infinite dimensional system Equations 13–15 is solved for $\boldsymbol{u}'$:

$$\begin{aligned} A\left(\boldsymbol{\psi}',\boldsymbol{u}'\right) &= -A\left(\boldsymbol{\psi}',\bar{\boldsymbol{u}}\right) \\ &= \left(\boldsymbol{\psi}',-\bar{\mathcal{R}}_{s,c}\right) \end{aligned} \tag{13}$$

$$\begin{aligned} B_1\left(\boldsymbol{\psi}',\boldsymbol{u}'\right)+B_2\left(\boldsymbol{\psi}',\bar{\boldsymbol{u}},\boldsymbol{u}'\right)+B_2\left(\boldsymbol{\psi}',\boldsymbol{u}',\bar{\boldsymbol{u}}\right)+B_2\left(\boldsymbol{\psi}',\boldsymbol{u}',\boldsymbol{u}'\right) &= -\left(B_1\left(\boldsymbol{\psi}',\bar{\boldsymbol{u}}\right)+B_2\left(\boldsymbol{\psi}',\bar{\boldsymbol{u}},\bar{\boldsymbol{u}}\right)\right) \\ &= \left(\boldsymbol{\psi}',-\bar{\mathcal{R}}_{s,m,i}\right) \end{aligned} \tag{14}$$

$$\begin{aligned} C_1\left(\boldsymbol{\psi}',\boldsymbol{u}'\right)+C_2\left(\boldsymbol{\psi}',\bar{\boldsymbol{u}},\boldsymbol{u}'\right)+C_2\left(\boldsymbol{\psi}',\boldsymbol{u}',\bar{\boldsymbol{u}},\right)+C_2\left(\boldsymbol{\psi}',\boldsymbol{u}',\boldsymbol{u}'\right) &= -\left(C_1\left(\boldsymbol{\psi}',\bar{\boldsymbol{u}}\right)+C_2\left(\boldsymbol{\psi}',\bar{\boldsymbol{u}},\bar{\boldsymbol{u}}\right)\right) \\ &= \left(\boldsymbol{\psi}',-\bar{\mathcal{R}}_{s,h}\right) \end{aligned} \tag{15}$$

In a variational multiscale framework, it is common to approximate the unresolved scales equations using evolution equations for $\boldsymbol{u}'$ (Codina et al., 2018). When most of the energy in the flow resides in the resolved scales, one may expect $\boldsymbol{u}'$ to be small and local enough to justify its approximation using element Green's functions (Hughes et al., 2018). Further assuming that the unresolved scales react instantaneously to their resolved counterparts results in algebraic approximations of the form:

$$\boldsymbol{u}' \approx -\boldsymbol{\tau}\bar{\boldsymbol{\mathcal{R}}} \tag{16}$$

Where $\boldsymbol{\tau}$ is a matrix of approximate element Green's functions and $\bar{\boldsymbol{\mathcal{R}}}=\left[\bar{\mathcal{R}}_{s,c}(\bar{\boldsymbol{u}}),\bar{\mathcal{R}}_{s,m,i}(\bar{\boldsymbol{u}}),\bar{\mathcal{R}}_{s,h}(\bar{\boldsymbol{u}})\right]$ are the residuals of the strong form of the continuity, momentum and energy equations, evaluated using the resolved portion of the solution, $\bar{\boldsymbol{u}}$. However, the best possible $\bar{\boldsymbol{u}}$ does not require the best possible model for $\boldsymbol{u}'$, but only for the closure terms (ii) and (iii) in Equations 10–12; we call these closure terms $\mathcal{C}$ from here on:

$$\mathcal{C}=\begin{cases} A\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}'\right) \\ B_2\left(\bar{\boldsymbol{\psi}},\bar{\boldsymbol{u}},\boldsymbol{u}'\right)+B_2\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}',\bar{\boldsymbol{u}}\right)+B_1\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}'\right)+B_2\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}',\boldsymbol{u}'\right) \\ C_1\left(\bar{\boldsymbol{\psi}},\bar{\boldsymbol{u}},\boldsymbol{u}'\right)+C_2\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}',\bar{\boldsymbol{u}}\right)+C_1\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}'\right)+C_2\left(\bar{\boldsymbol{\psi}},\boldsymbol{u}',\boldsymbol{u}'\right) \end{cases} \tag{17}$$

Since the terms in $\mathcal{C}$ are integrated over a volume associated with the typical grid spacing, they are likely to be less locally variable than $\boldsymbol{u}'$ itself, and the prospect of predicting them is somewhat less daunting. Therefore, we will attempt to pose models for $\mathcal{C}$ directly, given insights from the structure of Equations 13–15.

2.2. Contrast to LES and Superparameterization

In contrast to other modeling frameworks, the VMS formulation subsumes both discretization errors and modeling errors in its unresolved scales equations. Therefore, the resolved scales equations remain exact equations for $\bar{\boldsymbol{u}}$. As a result, it is not necessary to consider the numerical structure of flux operators in finite volume codes to construct exact models for $\mathcal{C}$.

However, the VMS model form is closely related to more familiar atmospheric modeling frameworks that have been coupled to early ANN parameterizations. By replacing terms (ii) and (iii) in Equations 10–12 with a diffusive term scaled by an eddy-viscosity, classical LES models are attained (Hughes et al., 2000), albeit in weak form. Closing such models with ANNs was considered in a finite volume setting by e.g., Beck et al. (2019).

Conversely, Rasp et al. (2018) replaced the Cloud Resolving Models (CRMs) that close superparameterization formulations of GCMs with ANNs. The general structure of SP models can also easily be attained from the VMS framework by considering a three-scale decomposition of the trial solution and weighting function spaces:

$$\mathcal{V} = \bar{\mathcal{V}} \oplus \tilde{\mathcal{V}} \oplus \mathcal{V}' \tag{18}$$

where $\bar{\mathcal{V}}$ would be the "large" scales space of the GCM, $\tilde{\mathcal{V}}$ the "small" scales space of an individual CRM and $\mathcal{V}'$ the still infinite-dimensional unresolved scales space. The three sets of governing equations that emerge from this decomposition and the range of scale interaction terms therein are included in Appendix A for the interested reader.

For SP, it is common to introduce further assumptions based on scale separation arguments in these equations (the validity of such assumptions will not be discussed here). For instance, in the context of Equations 10–12, it is often assumed that terms (ii) are 0, such that the equations on the large and small resolved scales can be separately stepped in time and adjusted to each other in each large-scale time step (Grabowski, 2001). The locality of the small resolved scales can be controlled by the locality of its basis and can (but does not need to be) grid-column contained, as is commonly assumed for SP (Grabowski, 2001, 2016; Grabowski & Smolarkiewicz, 1999). Hence, if it is too computationally expensive to generate training data for a global model's ANN parameterization with a model that resolves the entire range of scales of interest, the VMS model form can easily be cast in the form of SP, rather than the more general, "full" coupling presented in Equations 10–12 and considered in the rest of this work.

In summary, the VMS model form is general, flexible and can be adapted as one would like to couple large and small-scale models, in cases where an explicit scale separation assumption must be made. It can thus be used as proposed by Rasp et al. (2018) on the global scales, or on much smaller scales. This provides ample playing room for the ANN parameterization defined next.

2.3. Artificial Neural Networks

Machine learning techniques in general and ANNs in particular are set to become an increasingly attractive alternative to human reasoning for formulating models for unresolved processes in fluid flow problems (Gentine et al., 2018; Kutz, 2017). ANNs form a class of supervised machine learning techniques that consist of sequential layers of non-linear functions, connected by weights and additive constants that are adjusted during a "training" phase to stochastically minimize a "loss" function based on provided examples (LeCun et al., 2015). This architecture endows them with exceptional skill at inferring complex, non-linear relationships in the presence of large data sets (Krizhevsky et al., 2012), at potentially lower computational cost than directly resolving all such phenomena. In particular, ANNs have recently shown potential to improve the modeling of 3D homogeneous isotropic turbulence (Beck et al., 2019), simple boundary flow emulations (Srinivasan et al., 2019), and atmospheric convection (e.g., Brenowitz & Bretherton, 2018, 2019; Krasnopolsky et al., 2013; Rasp et al., 2018).

We use the VMS model defined by Equations 10–12 to investigate the ANNs' ability to infer and predict $\mathcal{C}$, based on a set of resolved-flow quantities. This can be interpreted as a generalization of the studies quoted above that removes *all* heuristic modeling and defers the evaluation of consistent interaction between the resolved and unresolved scales spaces of a discretized fluid flow problem to ANNs. The errors in our simulations' resolved scales are thus solely tied to how well an ANN can infer $\mathcal{C}$, assuming they are trained on "true" data. This increases the potential accuracy of the method to the maximum achievable accuracy of an ANN. More importantly, it cleanly allows us to assess the impact of an ANN on a running simulation.

It is important to note that in this setup, the ANN model for $\mathcal{C}$ is allowed to act both as a source and sink in Equations 10–12. While this allows the ANN to backscatter energy to the resolved simulation where necessary, there is no enforced control mechanism to maintain the energy in the domain. In other frameworks, similar approaches have resulted in numerical instabilities (Beck et al., 2019; Brenowitz & Bretherton, 2018; Stoffer et al., 2020).

Table 1
DALES Settings and Characteristics for the Convective Boundary Layer Model

Parameter	Value
Domain size $[N_x \times N_y \times N_z]$	$128 \times 128 \times 96$
Grid cell size $[h_x \times h_y \times h_z]$	40m × 40m × 20m
Simulation time [hr]	6
Spin-up time [hr]	1
Average time step [s]	10
Radiative sink S_θ [Ks^{-1}]	−0.075
Surface heat flux $\langle w''\theta''\rangle_s$ [Kms^{-1}]	0.06
Average inversion height z_{i_0} [m]	1,000

2.4. Model Framework

The general structure of the VMS-ANN framework we propose consists of four steps. First, we run an expensive, high-resolution simulation that explicitly resolves phenomena over a large range of scales for a particular problem. Second, we project the high-dimensional solution field onto a basis $\overline{\boldsymbol{\psi}}$ that contains significantly fewer degrees of freedom, but remains exactly equal to the high-resolution solution at predefined nodes. This defines the scale separation Equation 8 from which $\overline{\boldsymbol{u}}$, $\boldsymbol{u}'$ and $\mathcal{C}$ directly follow. At this stage, flow features F that are used to train ANNs are also constructed from $\overline{\boldsymbol{u}}$ and $\frac{\partial \overline{\boldsymbol{u}}}{\partial t}$. Third, we train ANNs to represent the unknown relation that maps F to $\mathcal{C}$ (Note that this relation is not guaranteed to exist). We do this outside the time stepping loop of dynamic simulations (offline). This allows us to conduct rapid training in an existing software environment (Chollet et al., 2015) after running our high-dimensional ground-truth runs. However, offline training on features constructed from a nodally exact solution do not inform an ANN of how errors that it will inevitably introduce to a simulation will feed back in subsequent steps of a time march, as noted i.a. by Rasp (2020). The fourth and final step of the framework, running numerical simulations with a trained ANN embedded as the unresolved scales model, aims to assess the impact of such concerns.

In summary, VMS-ANN models constitute a general, clear framework for data-driven unresolved scales modeling. However, they share many of the features that challenged the robustness of other initial ANN unresolved scales models in fluid flow problems: A lack of control over the evolution of energy in the resolved simulation and the reliance on ANNs trained outside a time march on exact data. In this light, the following sections discuss a numerical experiment in the framework that highlights these issues and offers perspectives on how one might improve on them.

3. A Numerical Experiment

3.1. Model Problem

The VMS-ANN model that emerges from the previous section is studied in the context of a highly simplified model problem, as a minimal model for the small-scale boundary layer turbulence that underpins the development of shallow clouds. The case we consider is a dry convective boundary layer as described in van Driel and Jonker (2010). This case considers a well-mixed layer of θ with an inversion height and strength that are all constant in time. Such a layer is developed from the balance of a constant heat flux imposed on the lower domain boundary and a radiative sink that one may interpret as a scaled subsidence. Hence, statistics of the solution in the vertical dimension are stationary in time. This reduces the amount of data needed to train ANNs relative to non-equilibrium cases, rendering this case appropriate as a first test.

3.2. High-Resolution Simulations

The turbulence that drives this case is simulated with the Dutch Atmospheric Large Eddy Simulation (DALES) model (Heus et al., 2010). DALES solves filtered versions of Equations 1–3 in finite volumes that fill a 3D domain with homogeneous, horizontal dimension, periodic boundary conditions imposed on the domain sides and standard boundary conditions on the top and bottom. A set of 10 DALES simulations are run with the parameters quoted in Table 1, but with different initial fields. Hence, after a 1 h spinup, this gives a rich set of different realizations of the same statistical turbulence across horizontal coordinate, time and simulation. This forms the data set that our ANNs will be trained upon.

The DALES simulations do not resolve the full range of turbulent scales and are therefore subject to the errors of any LES. Nevertheless, the LES results will be considered here as the ground truth data that our ANNs will be trained on and compared against. This is justified by training and evaluating the ANNs in

VMS runs at resolutions that are significantly coarser than these DALES simulations, such that our ANNs must capture the effects of an appreciable range of energetic scales from the original LES.

To further promote insight into the mechanics of ANN unresolved scales models, we only consider a forced, 1D inviscid Burgers equation for our experiments. This can be derived from Equation 3 as Equations 19 and 20:

$$\frac{\partial w}{\partial t} + \frac{\partial}{\partial z}\left(ww\right) = f \tag{19}$$

$$f = -\frac{\partial}{\partial x}\left(uw\right) - \frac{\partial}{\partial y}\left(vw\right) - \frac{\partial \pi}{\partial z} + \frac{g}{\theta_0}\left(\theta - \theta_0\right) \tag{20}$$

Where f can be identified from DALES simulations a posteriori in an individual grid column. This reduces the resolved-scales Equations 10–12 to:

Find $\overline{w}(z,t)$ such that $\forall \overline{\psi} \in \overline{\mathcal{V}}$:

$$\underbrace{\left(\overline{\psi},\frac{\partial \overline{w}}{\partial t}\right)_\Omega - \left(\frac{\partial \overline{\psi}}{\partial z},\overline{w}^2\right)_\Omega}_{\text{Galerkin terms}} + \underbrace{\underbrace{\left(\overline{\psi},\frac{\partial w'}{\partial t}\right)_\Omega}_{w'_t\text{ projection}} - \underbrace{\left(\frac{\partial \overline{\psi}}{\partial z},2\overline{w}w'\right)_\Omega}_{\text{Cross term}} - \underbrace{\left(\frac{\partial \overline{\psi}}{\partial z},w'^2\right)_\Omega}_{\text{Reynolds term}}}_{\mathcal{C}} = \underbrace{\left(\overline{\psi},f\right)_\Omega}_{\text{Forcing}} \tag{21}$$

Where the large underbraces gather the resolved contributions to the resolved solution that result directly from a Galerkin projection, the forcing and the three closure terms that comprise $\mathcal{C}$: The projections of the tendency of unresolved scales, the Cross term and the Reynolds term. The latter three will be estimated by the ANN. Letting $\langle \cdot \rangle$ denote a time average, this problem retains the vertical profile of the time-averaged vertical velocity flux $\langle w^2 \rangle(z)$ of the original convective boundary layer problem; the goal of a VMS-ANN model for this problem would be to reproduce such statistics. The exact $\overline{w}(z,t)$, F and $\mathcal{C}$ required for ANN training of this problem are constructed by sampling individual columns of DALES simulations and projecting the resulting $w(z,t)$ onto a nodally exact, piecewise linear basis with a constant element length that is up to 6 times coarser than the original DALES simulation's grid spacing (see Text S1 for pseudocode of this procedure).

3.3. ANN Training

We train ANNs to predict $\mathcal{C}(z,t)$ based on three types of inputs: (i) $\overline{w}(x_s,t_s)$, (ii) $\frac{\partial \overline{w}}{\partial t}(x_s)$ and (iii) $\overline{\mathcal{R}_I}(x_s)$; this input stencil is illustrated in Figure 2. x_s contains the node at which the weighting function is one, and its left and right neighbors, while t_s samples at the current and previous two time steps of a simulation. $\overline{\mathcal{R}_I}(x_s)$ is the quantity defined in Equations 13–15, integrated over the elements that span x_s:

$$\overline{\mathcal{R}_\mathcal{I}}(x_s) = \int_{x_s}\left(f - \frac{\partial \overline{w}}{\partial t} - \frac{\partial}{\partial z}\left(\overline{w}^2\right)\right)dz \tag{22}$$

Input (i) contains the local scales of the problem's solution in space and time, input ii) is observed to improve the prediction and input iii) represents the forcing of the resolved scales on the unresolved scales in Equations 13–15.

Using the Keras API to TensorFlow (Abadi et al., 2016; Chollet et al., 2015), we then search for an ANN that minimizes the mean L_2 norm of the difference between the predicted and exact $\mathcal{C}$. Note that this is equivalent to exactly satisfying the weak equations in the L_2 norm when all resolved contributions are exactly represented, and therefore indirectly equivalent to driving the ANNs toward closing the exact, discrete problem in that norm.

The ANN input features, architecture and optimizer are selected through hyperparameter optimisation (see Tables S1–S4 for details). In this procedure, we gauge an ANN configuration by its ability to minimize our loss function over a data set that consists of columns of the DALES simulations that lie outside the horizontal correlation length of w from any column that has been used in the training of an ANN ("validation"

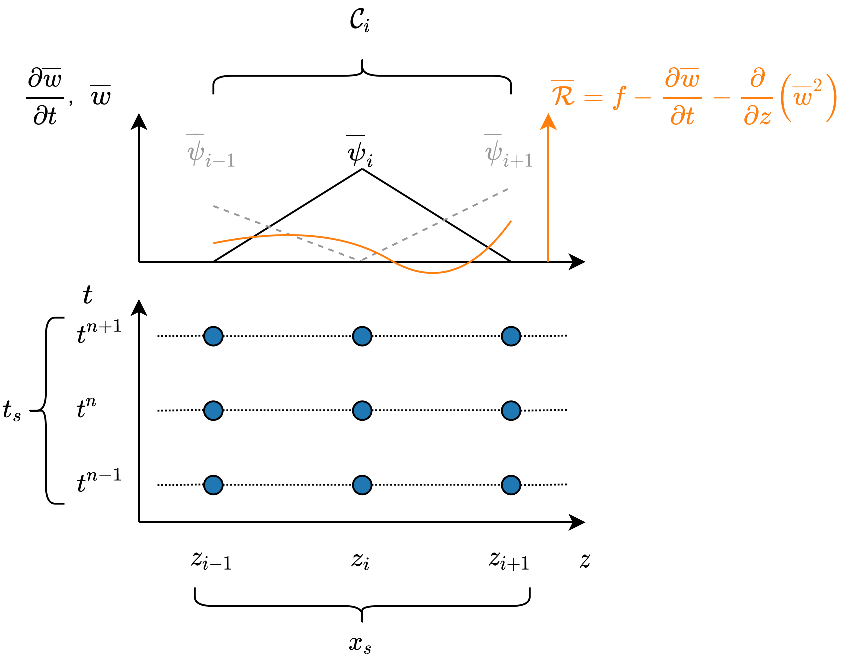
$\mathcal{C}_i$
$\frac{\partial \overline{w}}{\partial t}$, $\overline{w}$
$\overline{\psi}_{i-1}$
$\overline{\psi}_i$
$\overline{\psi}_{i+1}$
$\overline{\mathcal{R}} = f - \frac{\partial \overline{w}}{\partial t} - \frac{\partial}{\partial z}\left(\overline{w}^2\right)$
t
t^{n+1}
t_s
t^n
t^{n-1}
z_{i-1}
z_i
z_{i+1}
z
x_s

Figure 2. The i^{th} weak equation, solved for $\overline{\psi}_i$'s unknown amplitude at time level $n+1$ (t^{n+1}), is closed by $\mathcal{C}_i$, which is defined by an ANN taking the inputs $\overline{w}$, $\frac{\partial \overline{w}}{\partial t}$ and $\overline{\mathcal{R}}_I$, defined on a spatial stencil x_s and temporal stencil t_s. $\overline{\mathcal{R}}_I = \int_{x_s} \overline{\mathcal{R}} dz$, where $\overline{\mathcal{R}}$ is the residual of the resolved portion of Equation 19.

data set). In practice, this requires subsampling DALES columns with an interval of 4. The hyperparameter optimisation results in a densely connected network with two hidden layers of 512 neurons.

Finally, a "test" data set that is decorrelated from the validation examples is kept hidden during the tuning of ANN hyperparameters, but is used for our assessment of the ANN's offline performance (see Section 4). Note that our evaluation strategy gauges the ANN's ability to handle different realizations of the same statistical turbulence. Investigating how this ability would translate to statistically *different* turbulence is crucial for establishing the method's practical viability in the future, but is not part of our work.

3.4. Online Simulations

To test the ANN models in online simulations, they are directly substituted for $\mathcal{C}$ in the semi-discrete Equation 21. The time dimension of this equation is discretized with the following second-order scheme:

$$\frac{\partial \overline{w}}{\partial t}^{n+1} \approx \frac{3\overline{w}^{n+1} - 4\overline{w}^{n} + \overline{w}^{n-1}}{2\Delta t} \tag{23}$$

Where Δt is the time step and superscript $n+1$ denotes the discrete time level at which the solution is unknown; n and $n-1$ are the previous two time levels.

Our implicit time march is hardly the standard in operational atmospheric models. However, in contrast to multi-step explicit or hybrid time marches, implicit schemes express the problem solution's tendency at the next time level $\frac{\partial \overline{w}}{\partial t}^{n+1}$ clearly and consistently in terms of the solution at the next time level $\overline{w}^{n+1}$. This is useful, because if one can exactly predict $\mathcal{C}^{n+1}$, it remains possible to retain an exact prediction of $\overline{w}^{n+1}$. This is not possible if one uses an explicit scheme, where $\frac{\partial \overline{w}}{\partial t}^{n+1}$ is a function of $\mathcal{C}^{n}$. Since our objective is precisely to defer all modeling error to the ANN's ability to model $\mathcal{C}$, this time march is a particularly convenient choice for our VMM-ANN.

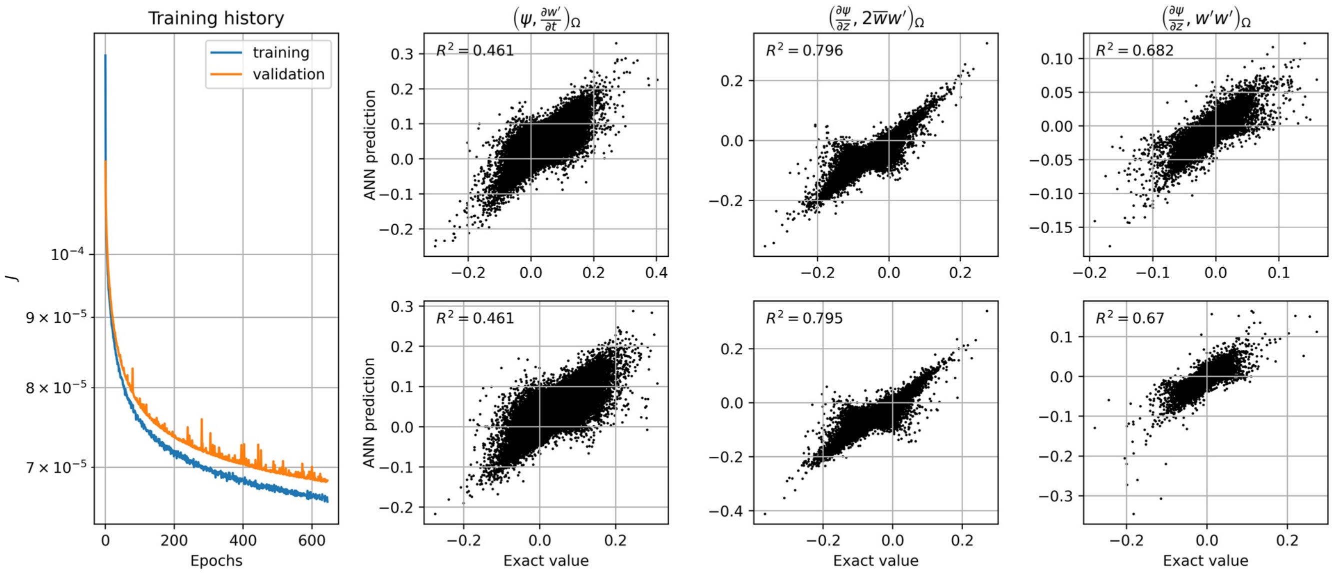

Figure 3. Training history on training and validation data and R^2 correlation on validation data (upper row) and test data (lower row) of each of the three closure terms for the best set of hyperparameters and input features.

Since the formulation is implicit, it results in a non-linear system of coupled, algebraic equations at t^{n+1}. We solve the system by driving the L_2 norm of the residual of Equation 21 to zero with a predictor-corrector scheme, where the corrector passes are Newton-Raphson iterations. This implies that the ANN will be called to predict $\mathcal{C}$ at each corrector pass, and that it must remain stable both within a time step and over many time steps.

4. Offline Results

Figure 3 shows the results of the training of our best ANN, for a case where $\overline{w}$ is defined on a basis with an element length that is 6 times as coarse as that of the DALES simulation and $\Delta t = 2\Delta t_{DALES}$. The training is conducted on $14.4 \cdot 10^6$ individual examples and run over 638 epochs during which the loss J is consistently reduced. The upper row displays the ANN's predictions of the three components of $\mathcal{C}$ on a held-out set of $1.07 \cdot 10^6$ randomly sampled validation examples used during all training runs to prevent the model from overfitting its training data. This data set has also been used to tune the model's hyperparameters. The bottom row displays results on the previously hidden $1.07 \cdot 10^6$ test examples. Finally, the plots show the coefficients of determination R^2 of the linear fit of the ANN prediction relative to the exact $\mathcal{C}$ for both data sets. This measure varies at most by 0.01 between the validation and test data sets, indicating that the ANN is able to generalize its predictions of $\mathcal{C}$ very well to previously unseen realizations of the same statistical turbulence.

The figure also exemplifies a pattern that broadly transfers to all ANNs we trained: The cross term, which is linear in w', is the easiest to learn, the non-linear Reynolds term is more challenging and the unresolved scales' time derivative projection is in most settings the most difficult term to learn.

To contextualize this performance, we compare the ANN's predictions of vertical profiles of $\langle \|\mathcal{C}\|_2 \rangle$ to (a) their exact counterparts and (b) those resulting from a standard, algebraic VMS scheme for u' in Burgers' problems of the form of Equation 16 (Shakib et al., 1991). Similar models tend to perform at least as proficiently as modern LES closures in full turbulence simulations (Bazilevs et al., 2007), and are thus a good representation of the state-of-the-art. These profiles are shown for the test data set in Figure 4.

The algebraic model reproduces the vertical profile of $\langle w^2 \rangle$ well in online simulations of this problem, even at $h/h_{DALES} = 6$ (not shown). However, it does not account for $\left(\overline{\psi}, \dfrac{\partial w'}{\partial t}\right)_\Omega$ and is unable to reproduce the statistics of the non-linear Reynolds term; its approximations only allow for decent predictions of the cross

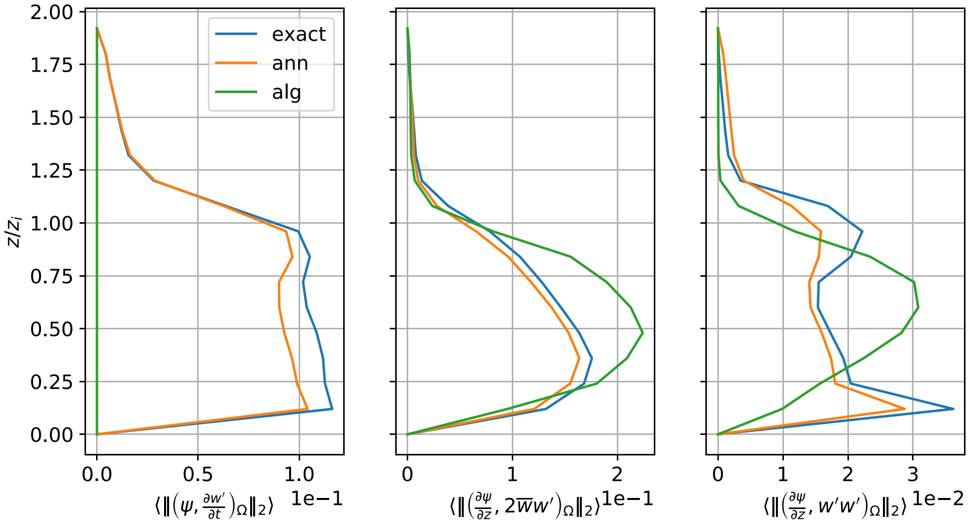

Figure 4. Comparison of time-averaged spatial distributions of the L_2 norms of the closure terms, as predicted by exact, ANN and algebraic models, at $h/h_{DALES} = 6$ and $\Delta t/\Delta t_{DALES} = 2$, over 16 columns from a different DALES simulation than was trained on.

term. The ANN dramatically improves the representation of all three terms, even with a relatively simple architecture and local input stencil, offering remarkable potential to improve the accuracy of traditional unresolved-scales models for this particular case.

5. Toward Online Applicability

Our offline results do not indicate to which extent our ANNs can generalize their performance to different flows or discretization levels. Similarly, these results do not guarantee that an ANN's performance will generalize to an implicit time march where the ANN is subjected to input from (a) an iterative solution procedure and (b) a time history and state that contain accumulated model errors, neither of which were seen during training. We do not reject the importance of verifying that the ANN models for $\mathcal{C}$ generalize to different flows and discretizations. However, we find our framework to give rise to instabilities in online simulations of our model problem. Therefore, we choose to first focus on generalization to the online dimension in this work. Specifically, we concentrate on understanding the mechanisms behind these instabilities, and thus will not assess the framework's ability to reproduce accurate statistics of $\langle\overline{w}^2\rangle$.

We find two distinct modes of instability, which correspond to the two dimensions of the time march of which our ANNs were unaware during training. First, we observe energy accumulation in the simulation's smallest, resolved scales over several time steps, in similar fashion to what is reported by Beck et al. (2019); Brenowitz and Bretherton (2018); Stoffer et al. (2020). However, we also see instabilities arise within a time step during corrector passes of the implicit time march. This second instability mode is both novel and instructive for informing a broader approach to treat ANN-induced instabilities. Therefore, we will elaborate on it here.

Instabilities within our implicit time march arise from the same aspect of the ANNs that makes them attractive: Their non-linear character. This can be illustrated by considering the L_2 norm of the weak residual across the domain, $\mathcal{R}_w$:

$$\mathcal{R}_w = \left\| \left(\overline{\psi}, \frac{\partial \overline{w}}{\partial t}\right)_\Omega - \left(\frac{\partial \overline{\psi}}{\partial z}, \overline{w}^2\right)_\Omega + \mathcal{C} - \left(\overline{\psi}, f\right)_\Omega \right\|_2 \tag{24}$$

During each time step, it is the objective of our Newton-Raphson scheme to drive $\mathcal{R}_w$ to 0. The Galerkin terms in this equation are quadratic functions of the problem's degrees of freedom a_i defined in Equation 9. Hence, if the model for the closure terms is also not more than a second-degree polynomial in the degrees of freedom, $\mathcal{R}_w$ will at most be a coupled quadratic in a_i with a maximum number of roots governed by

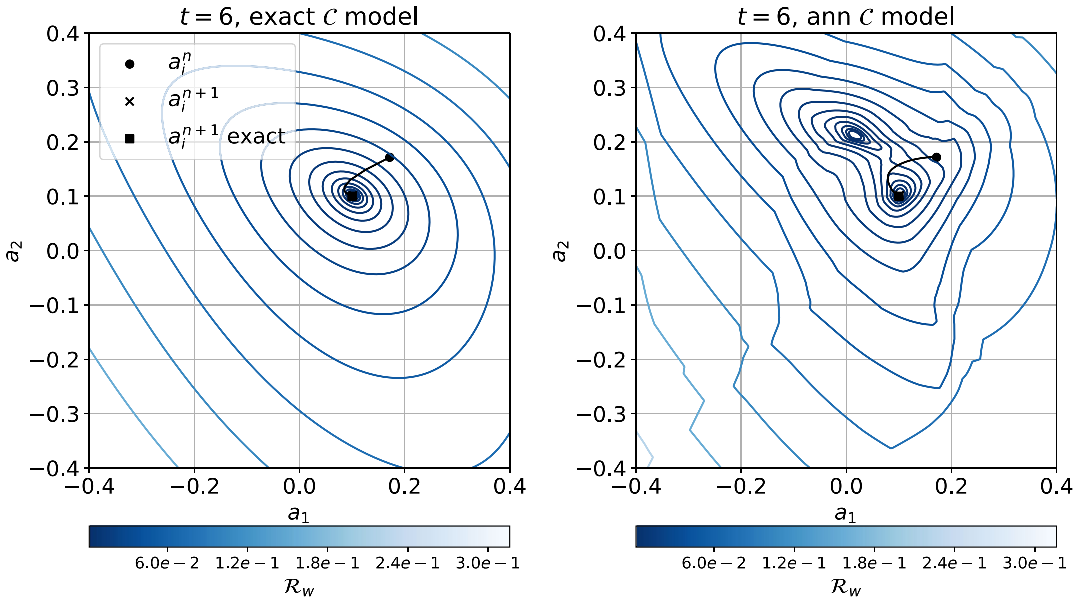

Figure 5. Response surfaces and convergence trajectories of $\mathcal{R}_{w_i}$ for an implicit time step of a manufactured solution problem with two degrees of freedom a_1 and a_2, using the exact $\mathcal{C}$ (left) and an ANN model for $\mathcal{C}$ (right). They derive from simulations where $\Delta t = 2$, $h = \frac{1}{3}$ and $\alpha = 1.2$.

Bézout's theorem (Séroul, 2000). However, the ANN models of the closure terms are highly non-linear in a_i, since they themselves are a function of $\overline{w}$ and a_i. Hence, large departures from a quadratic-shaped space for $\mathcal{R}_w$ can be expected for such models.

The implications of this can already be visualized in simulations that discretize the entire 1D domain using two degrees of freedom, a_1 and a_2. Figure 5 displays contours of $\mathcal{R}_w(a_1,a_2)$ in a given time step for such a model when (a) the exact $\mathcal{C}$ are inserted and (b) an ANN is employed. $\mathcal{R}_w$ has the expected, quadratic functional dependence on a_1 and a_2 when $\mathcal{C}$ is not a function of a_1 and a_2 and displays only a single root in this space. Moreover, only a few iterations are required to converge to this root during the displayed time step (left subfigure). In contrast, when our ANN predicts $\mathcal{C}$, $\mathcal{R}_w$ takes on a highly distorted, non-linear character (right subfigure). Despite accurately predicting the combination of a_1 and a_2 that correctly satisfies the weak form, the model contains a second, spurious root, surrounded by a second basin of attraction. This would be of little practical concern if the second attractor would be far removed from its physical counterpart. However, during the plotted time step, the model almost spills into the spurious attractor, which would immediately destabilize the simulation. Additionally, a large number of corrector passes are required to converge this model through a landscape of non-smooth gradients. Attractor switches, convergence to spurious solutions and stagnating convergence of the Newton-Raphson scheme are frequently observed for this model. Conventional measures such as underrelaxation or more accurate Jacobians do not improve matters: It is the highly non-linear character of $\mathcal{R}_w$, brought about by the ANN, that underpins this instability. In all, we conclude that inserting ANNs directly into an implicit time march can make the numerical scheme non-unique and ill-posed, resulting in instability.

We observe that the onset of instability is closely related to Δt, expressed relative to a characteristic velocity scale w^* (we take $w^* = \max_\Omega w$) and h through the Courant number α:

$$\alpha = \frac{w^*\Delta t}{h} \tag{25}$$

This can be seen in Figure 6, which shows $\mathcal{R}_w(a_1,a_2)$ for two different α in the first time step where the solutions' L_2 error rises above a preset threshold in a sequence of time steps that leads to the model's divergence. Note that this onset of instability arises progressively earlier for increasing α, as the problem becomes increasingly ill-posed.

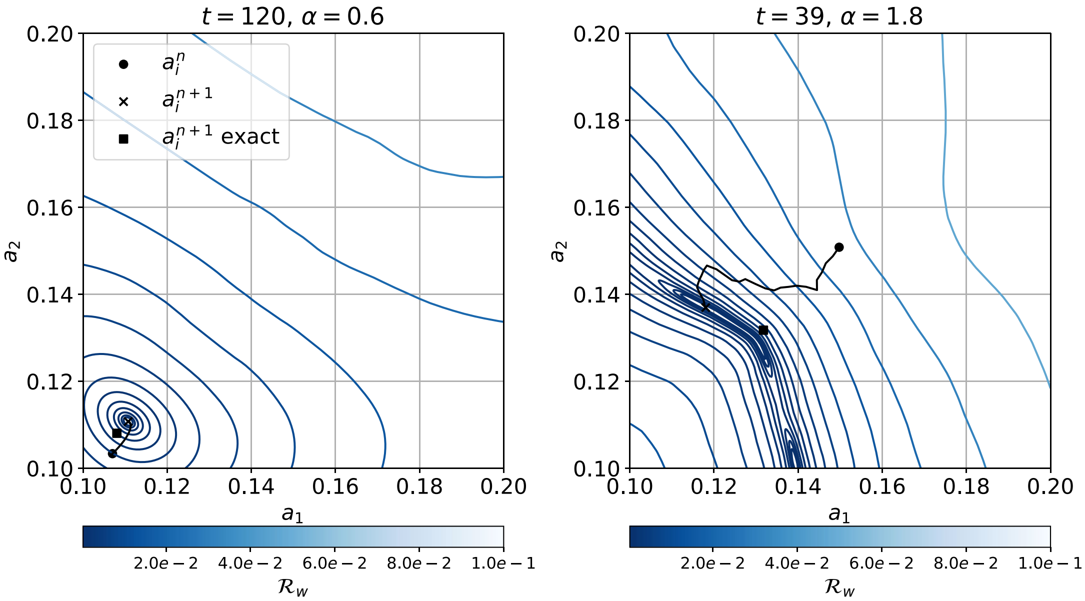

Figure 6. Response surfaces and convergence trajectories of $\mathcal{R}_w$ at the time step when the solution's L_2 error first exceeds $1 \cdot 10^{-4}$, for increasing α. $h = \frac{1}{3}$, $\Delta t = [1,3]$.

In particular, for increasing α, the initial prediction of a_1 and a_2 will on average be farther from its correct solution, requiring the model to traverse progressively long distances through the $\mathcal{R}_w$ space to find a root. This increases the likelihood of encountering (a) a spurious attractor along the way, or (b) regions where deficient gradient predictions prevent convergence. Additionally, the weak residual space itself becomes less workable at larger Δt: In the right half of Figure 6, multiple, clustered roots appear and the gradient predictions become increasingly erratic. Note that also the simulation at $\alpha = 0.6$ is unstable over several time steps in the small-scale, energy-accumulating pattern described before.

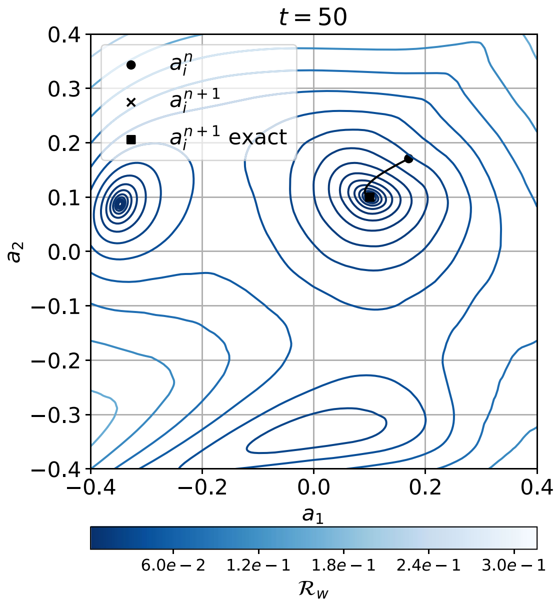

Figure 7. Response surface and convergence trajectories of $\mathcal{R}_w$ for the ANN model trained on data sampled from the convergence paths of exact models for $\mathcal{C}$, at $t = 50$ s for a simulation with $\Delta t = 2$, $h = 0.333$, $C = 1.2$.

One may hypothesize that the $\mathcal{R}_w$ space is ill-posed because the ANNs are trained to represent only the $\mathcal{C}$ that results in $\mathcal{R}_w = 0$. As a consequence, they have no knowledge of the space in $\mathcal{R}_w$ around this point and are unable to make consistent extrapolations to it. To test this hypothesis, we incorporate samples from the $\mathcal{R}_w$ space in our training data sets, by running online simulations with the exact $\mathcal{C}$ embedded in the numerical model (such as in the left portion of Figure 5) and, for each time step, sampling states of $\overline{w}$ that this model visits along convergence paths to the point where $\mathcal{R}_w = 0$ (see Text S2 for pseudocode of the creation of this enriched data set). These additional inputs all still point to the same, correct $\mathcal{C}$, to effectively over-constrain our training. Figure 7 shows that including these samples significantly improves the model's stability. The model now remains highly accurate and stable throughout the numerical experiment, evidenced by the reduced non-linear character of $\mathcal{R}_w$ and the broad physical basin of attraction, even after a much larger number of time steps has passed than in the previous experiments. As a result, many fewer corrector passes are also required to converge the model each time step. In all, this demonstrates that for an offline-trained ANN to be able to generalize to the space it is asked to navigate during an implicit time march, an exact data set does not suffice. Instead, the ANN's predictions in the partially converged time march space must be constrained, for instance by enriching the input data with examples from this space that the ANN is likely to traverse in online simulation.

At this point, it is worth pausing to note that both data set enrichment in particular (Cireşan et al., 2010; Simard et al., 2003) and promoting the regularity of an ANN's response to all its possible inputs in general (Kingma & Welling, 2013; Miyato et al., 2018) are popular means for enhancing the generalization capabilities of a broad class of ANNs. In fact, adding noise to an ANN's input data acts explicitly as a regularization on the network's response to deviations in its input (An, 1996). Such regularization can also be imposed directly in the network structure, for instance through spectral normalization (Shi et al., 2019), which might also improve the distorted character of the $\mathcal{R}_w(a_1, a_2)$ response surface shown in Figure 5. We view the data set enrichment employed here as a particularly targeted example of such regularization, as it adds data from portions of the input space that the ANN is likely to encounter.

The example of data set enrichment presented here also offers encouragement to resolving instabilities that develop over several time steps. As it turned out to be overly optimistic to assume that an ANN trained on exact F and $\mathcal{C}$ combinations could handle erroneous F in the early stages of a predictor-corrector scheme, it may also be too ambitious to expect that it will handle erroneous F that are caused by ANN prediction errors of $\mathcal{C}$ over several time steps. Such inputs may lead an ANN into non-physical regimes it never encountered during training, enlarging the prediction error and potentially instigating a positive feedback loop in the error. If this is the case, the long-term stability of the algorithm might be improved in a similar manner as the intra-time step instability encountered here: By including in the ANN's training the errors it generates over time when operating within the numerical model.

Rasp (2020) suggests implementing this idea by training ANN parameterizations *online* alongside a ground-truth model, which continually teaches the ANN how to cope with errors it feeds back onto itself. The drawback of such online learning is that it is potentially extremely costly, depending on how many ground-truth runs are necessary to adequately train such models. Fortunately, our results suggest that fully online training might not be necessary to account for the online dimension of prognostic problems: By resolving the intra-time step instabilities in our implicit time march through data set enrichment, we managed to maintain the offline training paradigm, requiring only a single run of our numerical model. This approach can be generalised to account for errors that develop over several time steps. By assembling statistics of an ANN's prediction errors over time, one can immediately create similar artificially corrupted, over-constrained training data sets as successfully employed here. Retraining an ANN on such a data set might strike a useful balance between the robustness of online training and the economy of offline training. Similar approaches have shown promise for comparable problems (Subel et al., 2021); our preliminary experiments using this approach have also been encouraging (Pusuluri, 2020).

6. Conclusions and Outlook

In this paper we have endeavored to clarify several aspects related to the use of data-driven parameterizations in global climate models. First, it was demonstrated that a variational-multiscale framework is useful in this context, as its basic form ensures the only source of error is due to the deficiencies in the employed parameterization, in this study an ANN. It was also demonstrated how traditional Large Eddy Simulation and Superparameterization methods can be derived from this framework by introducing additional modeling assumptions. Next, results from a simplified model problem revealed that some of the unresolved-scales closure terms are easier to parameterize than others. Specifically, the unresolved time-derivative term was found to be the most challenging, although good approximations were still obtained using relatively compact ANN architectures. This bodes well for the potential accuracy of the proposed technique.

The performance of the method was then examined for a single time step of an implicit time march. In contrast to the more difficult problem of studying instabilities over a large number of time steps, this approach provided a contained setting in which unstable behavior produced by ANN deficiencies could be clearly examined. Visualizations of corrector-pass residual response surfaces clearly indicated that nonphysical solutions produced by ANN parameterizations can significantly affect or prevent convergence. The severity of the problem increases with the size of the time step.

Noting that the residual response surface convergence paths required evaluations with inputs not encountered in training, it was theorized that the observed lack of convergence was partly due to a lack of generalization of the considered ANNs to this online setting. It was thus proposed to enrich the training data set with nonphysical corrector pass input/state combinations. This was found to substantially modify the residual response surface and proved to be an effective remedy, leading to a robust algorithm for implicit time-step evaluations. While enriching the data set with intra-time step examples did not treat instabilities which develop over several time steps, such instabilities might be postponed by enriching an ANN's training data set with statistics of the solution errors produced by an ANN parameterization over several time steps in online simulations, without requiring the training itself to take place online.

More broadly, this leads us to wonder how much data of their interaction with the temporal dimension data-driven parameterizations must be exposed to in order to robustly succeed in online calculations. Presumably, the answer to this question will play a role in determining the practical utility of data-driven parameterization, but simultaneously depend on the complexity of the dynamical system under consideration, which timescales of prediction are important and the details of the machine learning. Therefore, we recommend investigating this question broadly: Both in clear frameworks and simple situations that facilitate improved understanding of the fundamental interactions between the data-driven parameterization and numerical model, and in the complex, global models that we eventually wish to improve.

Appendix A: Three-Scale Decomposition of the Governing Equations

The infinite-dimensional system described by Equations 5–7 can be decomposed into three sets of trial and weighting solution function spaces, as suggested in Section 2.2:

$$\mathcal{V} = \bar{\mathcal{V}} \oplus \tilde{\mathcal{V}} \oplus \mathcal{V}' \tag{A1}$$

In this relation, $\bar{\mathcal{V}}$ is the space of large resolved scales, $\tilde{\mathcal{V}}$ is the space of small resolved scales and $\mathcal{V}'$ is the infinite-dimensional unresolved scales space. For a simulation framework such as Superparameterization (SP), $\bar{\mathcal{V}}$ represents the space in which a GCM would operate, $\tilde{\mathcal{V}}$ a space in which a CRM would operate and $\mathcal{V}'$ the unresolved scales below the CRM discretization. This scale decomposition gives rise to three sets of solutions, defined on each of these three function spaces:

$$\boldsymbol{u} = \bar{\boldsymbol{u}} + \tilde{\boldsymbol{u}} + \boldsymbol{u}' \tag{A2}$$

Such that the equations of motion at each scale become:

Find $\bar{\mathbf{u}} \in \bar{\mathcal{V}}$ such that:

$$\underbrace{A(\bar{\psi},\bar{\boldsymbol{u}})}_{\text{i}} + \underbrace{A(\bar{\psi},\tilde{\boldsymbol{u}})}_{\text{iv}} + \underbrace{A(\bar{\psi},\boldsymbol{u}')}_{\text{vi}} = 0 \tag{A3}$$

$$\begin{aligned}\underbrace{\boldsymbol{D_1}(\bar{\psi},\bar{\boldsymbol{u}}) + \boldsymbol{D_2}(\bar{\psi},\bar{\boldsymbol{u}},\bar{\boldsymbol{u}})}_{\text{i}} + \underbrace{\boldsymbol{D_2}(\bar{\psi},\bar{\boldsymbol{u}},\tilde{\boldsymbol{u}}) + \boldsymbol{D_2}(\bar{\psi},\tilde{\boldsymbol{u}},\bar{\boldsymbol{u}})}_{\text{ii}} +& \\ \underbrace{\boldsymbol{D_2}(\bar{\psi},\bar{\boldsymbol{u}},\boldsymbol{u}') + \boldsymbol{D_2}(\bar{\psi},\boldsymbol{u}',\bar{\boldsymbol{u}})}_{\text{iii}} + \underbrace{\boldsymbol{D_1}(\bar{\psi},\tilde{\boldsymbol{u}}) + \boldsymbol{D_2}(\bar{\psi},\tilde{\boldsymbol{u}},\tilde{\boldsymbol{u}})}_{\text{iv}} +& \\ \underbrace{\boldsymbol{D_2}(\bar{\psi},\tilde{\boldsymbol{u}},\boldsymbol{u}') + \boldsymbol{D_2}(\bar{\psi},\boldsymbol{u}',\tilde{\boldsymbol{u}})}_{\text{v}} + \underbrace{\boldsymbol{D_1}(\bar{\psi},\boldsymbol{u}') + \boldsymbol{D_2}(\bar{\psi},\boldsymbol{u}',\boldsymbol{u}')}_{\text{vi}} &= 0\end{aligned} \tag{A4}$$

Find $\tilde{\boldsymbol{u}} \in \tilde{\mathcal{V}}$ such that:

$$\underbrace{A(\tilde{\psi},\bar{\boldsymbol{u}})}_{\text{i}} + \underbrace{A(\tilde{\psi},\tilde{\boldsymbol{u}})}_{\text{iv}} + \underbrace{A(\tilde{\psi},\boldsymbol{u}')}_{\text{vi}} = 0 \tag{A5}$$

$$\begin{aligned}\underbrace{\boldsymbol{D_1}(\tilde{\psi},\bar{\boldsymbol{u}}) + \boldsymbol{D_2}(\tilde{\psi},\bar{\boldsymbol{u}},\bar{\boldsymbol{u}})}_{\text{i}} + \underbrace{\boldsymbol{D_2}(\tilde{\psi},\bar{\boldsymbol{u}},\tilde{\boldsymbol{u}}) + \boldsymbol{D_2}(\tilde{\psi},\tilde{\boldsymbol{u}},\bar{\boldsymbol{u}})}_{\text{ii}} +& \\ \underbrace{\boldsymbol{D_2}(\tilde{\psi},\bar{\boldsymbol{u}},\boldsymbol{u}') + \boldsymbol{D_2}(\tilde{\psi},\boldsymbol{u}',\bar{\boldsymbol{u}})}_{\text{iii}} + \underbrace{\boldsymbol{D_1}(\tilde{\psi},\tilde{\boldsymbol{u}}) + \boldsymbol{D_2}(\tilde{\psi},\tilde{\boldsymbol{u}},\tilde{\boldsymbol{u}})}_{\text{iv}} +& \\ \underbrace{\boldsymbol{D_2}(\tilde{\psi},\tilde{\boldsymbol{u}},\boldsymbol{u}') + \boldsymbol{D_2}(\tilde{\psi},\boldsymbol{u}',\tilde{\boldsymbol{u}})}_{\text{v}} + \underbrace{\boldsymbol{D_1}(\tilde{\psi},\boldsymbol{u}') + \boldsymbol{D_2}(\tilde{\psi},\boldsymbol{u}',\boldsymbol{u}')}_{\text{vi}} &= 0\end{aligned} \tag{A6}$$

and find $\boldsymbol{u}' \in \mathcal{V}'$ such that:

$$\underbrace{A(\psi',\bar{\boldsymbol{u}})}_{\text{i}} + \underbrace{A(\psi',\tilde{\boldsymbol{u}})}_{\text{iv}} + \underbrace{A(\psi',\boldsymbol{u}')}_{\text{vi}} = 0 \tag{A7}$$

$$\begin{aligned}&\underbrace{\boldsymbol{D_1}\left(\psi',\bar{\boldsymbol{u}}\right) + \boldsymbol{D_2}\left(\psi',\bar{\boldsymbol{u}},\bar{\boldsymbol{u}}\right)}_{\text{i}} + \underbrace{\boldsymbol{D_2}\left(\psi',\bar{\boldsymbol{u}},\tilde{\boldsymbol{u}}\right) + \boldsymbol{D_2}\left(\psi',\tilde{\boldsymbol{u}},\bar{\boldsymbol{u}}\right)}_{\text{ii}} + \\ &\underbrace{\boldsymbol{D_2}\left(\psi',\bar{\boldsymbol{u}},\boldsymbol{u}'\right) + \boldsymbol{D_2}\left(\psi',\boldsymbol{u}',\bar{\boldsymbol{u}}\right)}_{\text{iii}} + \underbrace{\boldsymbol{D_1}\left(\psi',\tilde{\boldsymbol{u}}\right) + \boldsymbol{D_2}\left(\psi',\tilde{\boldsymbol{u}},\tilde{\boldsymbol{u}}\right)}_{\text{iv}} + \\ &\underbrace{\boldsymbol{D_2}\left(\psi',\tilde{\boldsymbol{u}},\boldsymbol{u}'\right) + \boldsymbol{D_2}\left(\psi',\boldsymbol{u}',\tilde{\boldsymbol{u}}\right)}_{\text{v}} + \underbrace{\boldsymbol{D_1}\left(\psi',\boldsymbol{u}'\right) + \boldsymbol{D_2}\left(\psi',\boldsymbol{u}',\boldsymbol{u}'\right)}_{\text{vi}} = 0\end{aligned} \tag{A8}$$

To maintain some brevity in these equations, we have subsumed the momentum and energy equations in vectors $\boldsymbol{D_1} = [B_1, C_1]^T$ and $\boldsymbol{D_2} = [B_2, C_2]^T$. This should not distract from the main message, which is that equations on three sets of scales should contain models of the closure terms of each of those scales on their respective bases. This means that in the "GCM space" $\overline{\mathcal{V}}$, it becomes necessary to consistently parameterize several more sets of terms than in the two-scale decomposition discussed in the main text, as indicated by the braces in Equations A3 and A4:

ii) Interactions between large, resolved and small resolved scales projected onto $\overline{\mathcal{V}}$
iii) Interactions between large, resolved and unresolved scales projected onto $\overline{\mathcal{V}}$
iv) Interactions between small, resolved scales projected onto $\overline{\mathcal{V}}$
v) Interactions between small, resolved scales and unresolved scales projected onto $\overline{\mathcal{V}}$
vi) Interactions between unresolved scales projected onto $\overline{\mathcal{V}}$

Similarly, all terms but those in (iv) represent scale interactions that must be parameterized or imposed onto $\tilde{\mathcal{V}}$ in Equations A5 and A6. SP derives its computational efficiency from scale separation assumptions that set terms (ii), (iii), (v) and (vi) to zero in Equations 10 and 11 and terms (ii) and (iii) to zero in Equations 13 and 14. Yet, it is prudent to recognize that these terms do in fact exist and that their magnitude should be thoroughly quantified to assess the sacrifice of accuracy that this computational efficiency promotion demands.

Data Availability Statement

The data and scripts underlying the figures presented in this paper are available at https://doi.org/10.6084/m9.figshare.13675816.v2, along with the code required to preprocess data and train ANNs. Researchers that are interested in the code behind the numerical model used here are encouraged to contact the authors.

Acknowledgments

We gratefully acknowledge Pier Siebesma and Harm Jonker for their considerable contribution to shaping this study and for selecting an appropriate test case; we also thank Harm for running the DALES simulations. Finally, we acknowledge three anonymous reviewers for their insightful, constructive reviews, which have improved the manuscript's clarity.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: A system for large-scale machine learning. In *Osdi* (Vol. 16, pp. 265–283). https://doi.org/10.1007/s40899-016-0056-5

An, G. (1996). The effects of adding noise during backpropagation training on a generalization performance. *Neural Computation*, *8*(3), 643–674. https://doi.org/10.1162/neco.1996.8.3.643

Bazilevs, Y., Calo, V., Cottrell, J., Hughes, T., Reali, A., & Scovazzi, G. (2007). Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, *197*(1–4), 173–201. https://doi.org/10.1016/j.cma.2007.07.016

Beck, A., Flad, D., & Munz, C.-D. (2019). Deep neural networks for data-driven LES closure models. *Journal of Computational Physics*, *398*, 108910. https://doi.org/10.1016/j.jcp.2019.108910

Boucher, O., Randall, D., Artaxo, P., Bretherton, C., Feingold, G., Forster, P., et al. (2013). Clouds and aerosols. In *Climate change 2013: The physical science basis. Contribution of working group I to the fifth assessment report of the intergovernmental panel on climate change* (pp. 571–657). Cambridge University Press.

Brenowitz, N. D., Beucler, T., Pritchard, M., & Bretherton, C. S. (2020). Interpreting and stabilizing machine-learning parametrizations of convection. *Journal of the Atmospheric Sciences*, *77*(12), 4357–4375. https://doi.org/10.1175/jas-d-20-0082.1

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, *45*(12), 6289–6298. https://doi.org/10.1029/2018gl078510

Brenowitz, N. D., & Bretherton, C. S. (2019). Spatially extended tests of a neural network parametrization trained by coarse-graining. *Journal of Advances in Modeling Earth Systems*, *11*(8), 2728–2744. https://doi.org/10.1029/2019ms001711

Chollet, F., et al. (2015). Keras. Retrieved from https://keras.io

Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, *22*(12), 3207–3220.

Codina, R., Badia, S., Baiges, J., & Principe, J. (2018). Variational multiscale methods in computational fluid dynamics. In *Encyclopedia of computational mechanics* (2nd ed., pp. 1–28).

Dufresne, J.-L., & Bony, S. (2008). An assessment of the primary sources of spread of global warming estimates from coupled atmosphere–ocean models. *Journal of Climate*, *21*(19), 5135–5144. https://doi.org/10.1175/2008jcli2239.1

Duraisamy, K., Iaccarino, G., & Xiao, H. (2019). Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, *51*, 357–377. https://doi.org/10.1146/annurev-fluid-010518-040547

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, *45*(11), 5742–5751. https://doi.org/10.1029/2018gl078202

Grabowski, W. W. (2001). Coupling cloud processes with the large-scale dynamics using the cloud-resolving convection parameterization (CRCP). *Journal of the Atmospheric Sciences*, *58*(9), 978–997. https://doi.org/10.1175/1520-0469(2001)058<0978:ccpwtl>2.0.co;2

Grabowski, W. W. (2016). Towards global large eddy simulation: Super-parameterization revisited. *Journal of the Meteorological Society of Japan Series II*, *94*(4), 327–344. https://doi.org/10.2151/jmsj.2016-017

Grabowski, W. W., & Smolarkiewicz, P. K. (1999). CRCP: A cloud resolving convection parameterization for modeling the tropical convecting atmosphere. *Physica D: Nonlinear Phenomena*, *133*(1–4), 171–178. https://doi.org/10.1016/s0167-2789(99)00104-9

Guan, Y., Chattopadhyay, A., Subel, A., & Hassanzadeh, P. (2021). Stable a posteriori les of 2d turbulence using convolutional neural networks: Backscattering analysis and generalization to higher re via transfer learning. *arXiv preprint arXiv:2102.11400*.

Heus, T., van Heerwaarden, C. C., Jonker, H. J., Siebesma, A. P., Axelsen, S., vanden Dries, K., et al. (2010). Formulation of and numerical studies with the Dutch Atmospheric Large-Eddy Simulation (DALES). *Geoscientific Model Development*, *3*, 415–444. https://doi.org/10.5194/gmd-3-415-2010

Hope, C. (2015). The $10 trillion value of better information about the transient climate response. *Philosophical Transactions of the Royal Society A*, *373*(2054), 20140429.

Hughes, T. J. (1995). Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, *127*(1–4), 387–401. https://doi.org/10.1016/0045-7825(95)00844-9

Hughes, T. J., Feijóo, G. R., Mazzei, L., & Quincy, J.-B. (1998). The variational multiscale method - A paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, *166*(1–2), 3–24. https://doi.org/10.1016/s0045-7825(98)00079-6

Hughes, T. J., Mazzei, L., & Jansen, K. E. (2000). Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, *3*(1–2), 47–59. https://doi.org/10.1007/s007910050051

Hughes, T. J., Scovazzi, G., & Franca, L. P. (2018). Multiscale and stabilized methods. In *Encyclopedia of computational mechanics* (2nd ed., pp. 1–64).

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Klein, R. (2010). Scale-dependent models for atmospheric flows. *Annual Review of Fluid Mechanics*, *42*, 249–274. https://doi.org/10.1146/annurev-fluid-121108-145537

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021). Machine learning accelerated computational fluid dynamics. *arXiv preprint arXiv:2102.01010*.

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2013). Using ensemble of neural networks to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud resolving model. *Advances in Artificial Neural Systems*, *2013*, 5. https://doi.org/10.1155/2013/485913

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kutz, J. N. (2017). Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, *814*, 1–4. https://doi.org/10.1017/jfm.2016.803

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Majda, A. J., & Grooms, I. (2014). New perspectives on superparameterization for geophysical turbulence. *Journal of Computational Physics*, *271*, 60–77. https://doi.org/10.1016/j.jcp.2013.09.014

Marras, S., Moragues, M., Vázquez, M., Jorba, O., & Houzeaux, G. (2013a). Simulations of moist convection by a variational multiscale stabilized finite element method. *Journal of Computational Physics*, *252*, 195–218. https://doi.org/10.1016/j.jcp.2013.06.006

Marras, S., Moragues, M., Vázquez, M., Jorba, O., & Houzeaux, G. (2013b). A variational multiscale stabilized finite element method for the solution of the Euler equations of nonhydrostatic stratified flows. *Journal of Computational Physics*, *236*, 380–407. https://doi.org/10.1016/j.jcp.2012.10.056

Maulik, R., & San, O. (2017). A neural network approach for the blind deconvolution of turbulent flows. *Journal of Fluid Mechanics*, *831*, 151–181. https://doi.org/10.1017/jfm.2017.637

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Park, J., & Choi, H. (2021). Toward neural-network-based large eddy simulation: Application to turbulent channel flow. *Journal of Fluid Mechanics*, *914*, A16. https://doi.org/10.1017/jfm.2020.931

Pusuluri, A. (2020). *Noise-augmented offline training of ANN unresolved-scale models*. MSc. Thesis, Aerospace Engineering. TU Delft. Retrieved from http://resolver.tudelft.nl/uuid:03b345f6-464a-4dd2-ac50-4fa9c880c839

Rasp, S. (2020). Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: General algorithms and Lorenz 96 case study (v1. 0). *Geoscientific Model Development*, *13*(5), 2185–2196. https://doi.org/10.5194/gmd-13-2185-2020

Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39), 9684–9689. https://doi.org/10.1073/pnas.1810286115

Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017). Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, *44*(24), 12396–12417. https://doi.org/10.1002/2017gl076101

Schneider, T., Teixeira, J., Bretherton, C. S., Brient, F., Pressel, K. G., Schär, C., & Siebesma, A. P. (2017). Climate goals and computing the future of clouds. *Nature Climate Change*, *7*(1), 3–5. https://doi.org/10.1038/nclimate3190

Séroul, R. (2000). The bézout theorem. In *Programming for mathematicians* (p. 10). Springer Verlag.

Shakib, F., Hughes, T. J., & Johan, Z. (1991). A new finite element formulation for computational fluid dynamics: X. the compressible euler and navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, *89*(1–3), 141–219. https://doi.org/10.1016/0045-7825(91)90041-4

Shi, G., Shi, X., O'Connell, M., Yu, R., Azizzadenesheli, K., Anandkumar, A., & Chung, S.-J. (2019). Neural lander: Stable drone landing control using learned dynamics. In *2019 international conference on robotics and automation (icra)* (pp. 9784–9790). https://doi.org/10.1109/icra.2019.8794351

Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar* (Vol. 3).

Srinivasan, P., Guastoni, L., Azizpour, H., Schlatter, P., & Vinuesa, R. (2019). Predictions of turbulent shear flows using deep neural networks. *Physical Review Fluids*, *4*(5), 054603. https://doi.org/10.1103/physrevfluids.4.054603

Stoffer, R., van Leeuwen, C. M., Podareanu, D., Codreanu, V., Veerman, M. A., Janssens, M., & van Heerwaarden, C. C. (2020). Development of a large-eddy simulation subgrid model based on artificial neural networks: A case study of turbulent channel flow. *Geoscientific Model Development Discussions*, *14*, 3796–3788.

Subel, A., Chattopadhyay, A., Guan, Y., & Hassanzadeh, P. (2021). Data-driven subgrid-scale modeling of forced burgers turbulence using deep learning with generalization to higher Reynolds numbers via transfer learning. *Physics of Fluids*, *33*(3), 031702. https://doi.org/10.1063/5.0040286

van Driel, R., & Jonker, H. J. (2010). Convective boundary layers driven by non-stationary surface heat fluxes. *Journal of the Atmospheric Sciences*, *68*, 272–738.

Yuval, J., & O'Gorman, P. A. (2020). Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature Communications*, *11*(1), 1–10. https://doi.org/10.1038/s41467-020-17142-3

Yuval, J., O'Gorman, P. A., & Hill, C. N. (2021). Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision. *Geophysical Research Letters*, *48*(6), e2020GL091363. https://doi.org/10.1029/2020gl091363