https://doi.org/10.5194/gmd-2020-289
Preprint. Discussion started: 10 November 2020

Development of a large-eddy simulation subgrid model based on artificial neural networks: a case study of turbulent channel flow

Robin Stoffer[1], Caspar M. van Leeuwen[2], Damian Podareanu[2], Valeriu Codreanu[2], Menno A. Veerman[1], Martin Janssens[1], Oscar K. Hartogensis[1], and Chiel C. van Heerwaarden[1]

[1]Meteorology and Air Quality Group, Wageningen University, Wageningen, The Netherlands
[2]SURFsara, Amsterdam, The Netherlands

Correspondence: Robin Stoffer (robin.stoffer@wur.nl)

Abstract. Atmospheric boundary layers and other wall-bounded flows are often simulated with the large-eddy simulation (LES) technique, which relies on subgrid-scale (SGS) models to parameterize the smallest scales. These SGS models often make strong simplifying assumptions. Also, they tend to interact with the discretization errors introduced by the popular LES approach where a staggered finite-volume grid acts as an implicit filter. We therefore developed an alternative LES SGS model based on artificial neural networks (ANNs) for the computational fluid dynamics code MicroHH (v2.0), which can be run in direct numerical simulation (DNS) and LES mode. We used a turbulent channel flow (with friction Reynolds number $Re_\tau = 590$) as a test case. The developed SGS model has been designed to require fewer simplifying assumptions, and to compensate for the instantaneous discretization errors introduced by the staggered finite-volume grid. We trained the ANNs based on instantaneous flow fields from a direct numerical simulation (DNS) of the selected channel flow. In general, we found excellent agreement between the ANN predicted SGS fluxes and the SGS fluxes derived from DNS for flow fields not used during training (with the correlation coefficient ρ mostly varying between 0.6 and 1.0), showing the potential ANNs may have to construct highly accurate SGS models. However, we observed an artificial build-up of turbulence kinetic energy at high wave modes when we directly incorporated our ANN SGS model into a LES simulation of the selected channel flow, eventually resulting in numeric instability. We hypothesized that error accumulation and aliasing errrors, were both important contributors to the observed instability. Several obstacles therefore remain before the a priori promise of our ANN LES SGS model, can be successfully leveraged in practical applications.

1 Introduction

Large-eddy simulation (LES) is an often used technique to simulate turbulent atmospheric boundary layers (ABLs) and other wall-bounded geophysical flows with high Reynolds numbers (e.g. rivers). These turbulent flows are challenging to simulate because of their strong non-linear dynamics and large ranges in spatial/temporal scales. LES explicitly resolves only the largest, most energetic, turbulent structures in these flows, while parameterizing the smaller ones with so-called subgrid-scale (SGS) models. This allows LES to keep the total computational effort feasible for today's high-performance computing systems, but makes the quality of the results strongly dependent on the chosen SGS model. As an SGS model based on physical principles alone does not exist, the SGS models used today typically rely on simplifying assumptions in combination with ad-hoc empirical corrections (Pope, 2001; Sagaut, 2006, e.g). We therefore wonder whether fully data-driven SGS models based on machine learning techniques, which require fewer simplifying assumptions, can become a viable alternative. Given the continuous increase in available computational power and recent advancements in applying machine learning to complex physical problems including turbulent flows, this is a timely question to address (e.g. Carleo et al., 2019; Brunton et al., 2020; Duraisamy et al., 2019; Kutz, 2017).

To briefly illustrate the effects simplifying assumptions can have, we take as an example the eddy-viscosity assumption used in the popular Smagorinsky model (Smagorinsky, 1963; Lilly, 1967) and several other SGS models. Crucially, the eddy-viscosity assumption introduces an alignment between the Reynolds stress and strain rate tensor that has not been verified in experimental data (Schmitt, 2007). This makes it impossible to produce both the correct Reynolds stresses and dissipation rates (Jimenez and Moser, 2000). As a consequence, eddy-viscosity SGS models often require ad-hoc manual corrections (e.g. tuning the Smagorinsky coefficient and/or implementing a wall-damping function) or multiple computationally expensive spatial filtering operations (e.g. scale-dependent dynamical Smagorinsky models (Bou-Zeid et al., 2005)) to achieve satisfactory results.

Data-driven machine learning techniques are, in contrast, much more flexible regarding their functional form. This is especially valid for artificial neural networks (ANNs): simple feed-forward ANNs with just one hidden layer are theoretically able to represent any continuous function on finite domains (i.e they are universal approximators; Hornik et al. (1989)). This makes them potentially very suitable for LES SGS modelling, where the 'real' SGS terms are unknown and highly non-linear.

Currently, in flow and turbulence modelling a wide effort is therefore undertaken to explore the potential for ANNs and other machine learning techniques (Brunton et al., 2020; Kutz, 2017; Duraisamy et al., 2019). In particular, multiple studies successfully modelled turbulence in Reynolds-Averaged Navier-Stokes (RANS) codes with machine learning techniques trained on high-fidelity direct numerical simulations (DNS) that resolve all relevant turbulence scales (e.g. Ling et al., 2016a, b; Wang et al., 2017; Wu et al., 2018; Singh et al., 2017).

Several other efforts in literature experimented with similar approaches in LES SGS modelling (Beck et al., 2019; Cheng et al., 2019; Gamahara and Hattori, 2017; Maulik et al., 2019; Milano and Koumoutsakos, 2002; Sarghini et al., 2003; Vollant et al., 2017; Wang et al., 2018; Xie et al., 2019; Yang et al., 2019; Zhou et al., 2019). Most of them used DNS fields as a basis, and subsequently applied a downscaling procedure to generate consistent pairs of coarse-grained fields (that are assumed to represent the fields that a LES code would generate) and the quantity of interest (e.g. the 'true' subgrid transport or the closure term itself). These pairs were then typically used to train ANNs in a supervised way. Some studies showed very promising results with this method, both in *a priori* tests (where the predicted quantity is directly compared to the ones derived from DNS) and *a posteriori* tests (where the trained ANN is directly incorporated as a SGS model into a LES simulation). However, these studies focused on test cases that are less complex than wall-bounded geoscientific flows: (in)compressible isotropic turbulence (Beck et al., 2019; Vollant et al., 2017; Wang et al., 2018; Zhou et al., 2019; Xie et al., 2019) and Kraichnan turbulence (Maulik et al., 2019).

There are also studies that attempted similar methods in cases that better represent ABLs. Some of them focused on LES wall modelling specifically (Milano and Koumoutsakos, 2002; Yang et al., 2019), which is challenging on its own because of the many unresolved near-wall motions that the wall model has to take into account. Sarghini et al. (2003) and Gamahara and Hattori (2017), in turn, focused on SGS modelling in the whole turbulent channel flow. Sarghini et al. (2003) used neural networks to predict the Smagorinsky coefficient in the Smagorinsky-Bardina SGS model (Bardina et al., 1980) reaching a computational time saving of about 20%. Gamahara and Hattori (2017) directly predicted the SGS turbulent transport with a neural network, using DNS during training. They got reasonable a priori results, but did not perform an a posteriori test. Finally, another important step towards application of these methods in realistic atmospheric boundary layers was taken by Cheng et al. (2019). They performed an extensive a priori test for an ANN-based LES SGS model covering a wide range of grid resolutions and flow stabilities (from neutral channel flow to very unstable convective boundary layers). We emphasize though that, for successful integration of ANN-based SGS models in practical applications, accurate and numerically stable a posteriori results are an important requirement. So far however, to the best of our knowledge, the a posteriori performance of ANN-based LES SGS models has not been documented in wall-bounded flows.

On top of that, the ANN LES SGS models, together with traditional eddy-viscosity models, do not directly reflect the popular LES approach where a staggered finite-volume numerical scheme acts as an implicit filter. Eddy-viscosity models are typically derived based on a generic filtering operation that does not consider the finite discrete nature of the used numerical grid (i.e. it is usually thought of as an analytical filter like a continuous top-hat filter), while the ANN SGS models so far did not attempt to compensate for all the discretization errors arising in simulations with staggered finite-volumes. These discretization errors, however, can strongly influence the resolved dynamics (e.g. Ghosal, 1996; Chow and Moin, 2003; Giacomini and Giometto, 2020), especially at the smallest resolved scales. Since the ANNs haves access to both the instantaneous DNS flow fields and corresponding coarse-grained field during training, the instantaneous discretization errors can in principle be taken into account.

Within this context, there have been a couple of noteworthy studies (Langford and Moser, 1999; Völker et al., 2002; Zandonade et al., 2004) that introduced the framework of *perfect* and *optimal* LES. Based on this framework, these studies approximated the full LES closure terms (that account for both the unresolved physics and all instantaneous discretization errors) with a data-driven approach based on DNS. The statistical method they used for this purpose though (i.e. stochastic estimation), still made additional assumptions about the functional form of the LES SGS model (e.g. linearity).A recent study by Beck et al. (2019) used therefore instead ANNs to approximate, in a similar way as the aforementioned studies, the full LES closure terms. To construct based on these ANNs an LES SGS model that is numerically stable a posteriori, they combined the ANNs with eddy-viscosity models. They did not specifically focus on the discretization errrors associated with staggered finite-volume grids, and did not consider wall-bounded turbulent flows like ABLs.

In the current study, we therefore made a first attempt to construct an ideally functioning ANN SGS model that i) compensates for the unresolved physics and all the instantaneous discretization errors introduced by implicit filtering with staggered finite-volume numerical schemes, ii) requires few simplifying assumptions, iii) works both near walls and in the bulk flow without the need for a wall model, iv) does not require ad-hoc empirical corrections, and v) produces numerically stable, accurate a posteriori results in wall-bounded flows. As a test case for our attempt we used wall-bounded turbulent channel flow ($Re_\tau = 590$), which can be seen as a simplified neutrally stratified atmospheric boundary layer. We used as a modelling tool the DNS and finite-volume LES code MicroHH (v2.0), which is designed for (near-)surface boundary layers and wall-bounded flows (van Heerwaarden et al., 2017).

In the following sections, we will first explain our methodology in more detail. To this end, we will start with a more detailed introduction of how we compensate for instantaneous discretization errors into our LES ANN SGS model (Sect. 2), followed by a description of our turbulent channel flow test case (Sect. 3), training data generation procedure (Sect. 4), ANN design (Sect. 5), and ANN training (Sect. 6). After that, we will focus on the performance and behaviour of our LES ANN SGS model. This includes both an a priori (Sect. 7) and a posteriori test (Sect. 8), together with an additional analysis that gives some insights into which input variables matter most to our model (Sect. 7.3). Finally, we will summarize our findings and give our main conclusions (Sect. 9).

2 Subgrid modelling for finite volume LES

One of our key objectives is to construct an ANN LES SGS model that accounts for the instantaneous discretization errors introduced by implicit filtering with staggered finite-volume numerical schemes (Sect. 1). To derive such an SGS model, we used as a starting point the Navier-Stokes momentum conservation equations for a Newtonian, incompressible fluid without buoyancy effects (which is appropriate for the test case used in this study, see Sect. 3):

$$\frac{\partial u_j}{\partial t} = -\frac{\partial u_i u_j}{\partial x_i} - \frac{1}{\rho_0}\frac{\partial P}{\partial x_j} + \nu \frac{\partial^2 u_j}{\partial {x_i}^2}, \tag{1}$$

where u_j (u, v, w) $[\mathrm{m\ s^{-1}}]$ is the wind velocity along the j-th direction, t [s] the time, x_i and x_j [m] the positions in the i-th direction and j-th direction respectively, ρ_0 $[\mathrm{kg\ m^{-3}}]$ the density, P [Pa] the pressure, and ν $[\mathrm{m^2\ s^{-1}}]$ the kinematic viscosity.

The governing LES equations are usually derived by applying a *generic*, unspecified filtering operation $\boldsymbol{G}$ to Eq. 1, which introduces a subgrid term $\tau_{ij} \equiv \overline{u_i u_j} - \overline{u_i}\ \overline{u_j}$ that has to be modeled (Pope, 2001; Sagaut, 2006). Traditional subgrid models like Smagorinsky (Smagorinsky, 1963; Lilly, 1967) attempt to model τ_{ij} associated with $\boldsymbol{G}$. However, by only considering the generic operation $\boldsymbol{G}$, they cannot directly compensate for the discretization errors arising on a specific finite numerical grid. Although the impact of the discretization errors can be reduced by adopting an explicit filtering technique (for instance by increasing the grid resolution compared to the filter width), this is in practice often not done because of the high computational cost (Sagaut, 2006). It may therefore be beneficial to develop a LES SGS model that directly compensates for the introduced discretization errors, ideally such that explicit filtering is not required anymore.

To this end we applied the *finite volume* filtering operation $\boldsymbol{G^{FV}}$, to Eq. 1 instead of the *generic* operation $\boldsymbol{G}$. $\boldsymbol{G^{FV}}$ is defined as a 3D top-hat filter sampled on a priori defined finite volume grid, where the finite sampling implicitly imposes an additional spectral cutoff filter (Langford and Moser, 1999; Zandonade et al., 2004). We used $\boldsymbol{G^{FV}}$ to derive an alternative set of LES equations (Eq. 3) that reflects the employed finite volume grid, and removes the need for commutation between the filtering and spatial differentiation operators (Eq. 3; Denaro (2011)). This allowed us to explicitly include many instantaneous discretization errors in the definition τ_{ij} (Eq. 5), making use of prior knowledge about the employed finite volume grid and numerical schemes.

Considering for the sake of clarity only equidistant LES grids, following Zandonade et al. (2004) the filtered velocity associated with $\boldsymbol{G^{FV}}$, $\overline{u_j}$, at a certain grid cell with indices $(\mathrm{l},\mathrm{m},\mathrm{n})$ can be written as:

$$\overline{u_j}(\mathrm{l},\mathrm{m},\mathrm{n}) = \frac{1}{\Delta x \Delta y \Delta z} \int\limits_{\Omega_j(\mathrm{l},\mathrm{m},\mathrm{n})} u_j(x,y,z)\ \mathrm{d}\boldsymbol{x}', \tag{2}$$

where $\Delta x, \Delta y, \Delta z$ are the equidistant filter widths in the three spatial directions, $\Omega_j(\mathrm{l},\mathrm{m},\mathrm{n})$ the grid cube/control volume for u_j at the considered grid cell, and $\boldsymbol{x}$ a vector indicating the position (x,y,z) in the flow domain. Since we focused in this study on *staggered* finite-volume grids (Table 1), the location of each control volume $\Omega_j(\mathrm{l},\mathrm{m},\mathrm{n})$ depends on the j-component considered (Sect. 4).

Applying the finite volume filter to Eq. 1, using the divergence theorem to convert the volume integrals to surface integrals, and combining the advection and viscous stress terms, for a certain grid cell we get an expression similar to that obtained by Zandonade et al. (2004):

$$\frac{\partial \overline{u_j}(\mathrm{l},\mathrm{m},\mathrm{n})}{\partial t} = -\frac{1}{\Delta x \Delta y \Delta z} \int\limits_{\partial\Omega_j(\mathrm{l},\mathrm{m},\mathrm{n})} (u_i u_j - \nu \frac{\partial u_j}{\partial x_i}) n_i\ \mathrm{d}\boldsymbol{x}' - \frac{1}{\Delta x \Delta y \Delta z}\frac{1}{\rho_0} \int\limits_{\partial\Omega_j(\mathrm{l},\mathrm{m},\mathrm{n})} p n_j\ \mathrm{d}\boldsymbol{x}', \tag{3}$$

where $\partial\Omega_j(\mathrm{l},\mathrm{m},\mathrm{n})$ is the surface area of the control volume $\Omega_j(\mathrm{l},\mathrm{m},\mathrm{n})$, and n_i, n_j the i-th and j-th component respectively of the outward pointing normal vector $\boldsymbol{n}$ corresponding to $\partial\Omega_j(\mathrm{l},\mathrm{m},\mathrm{n})$. Noteworthy is that, by invoking the divergence theorem, the divergence operator itself is effectively replaced by surface integrals, which removes the need for a commutative filter (Denaro, 2011) and avoids the truncation errors introduced by the discretization of the divergence operator on the finite grid.

The well-known closure problem does, off course, persist. In fact, none of the terms on the right-hand side of Eq. 3 can be determined exactly on the available finite LES grid and therefore have to be approximated. As argued however by Langford and Moser (2001) and Zandonade et al. (2004), an *optimal* formulation for the pressure term is impractical and barely more accurate than traditional finite-volume pressure schemes. We will therefore only consider the discretization errors in the advection and viscous stress terms. We further note that, in contrast to eddy-viscosity SGS models, the isotropic part of the transport terms does not have to be incorporated in a modified pressure term.

To approximate the advection and viscous stress terms on the finite LES grid, in this study we used second-order linear interpolations (Sect. 4, 8). If we then consider specifically i) the control volume of the u-component, and ii) the transport in vertical direction, we can rewrite the first term at the right-hand side of Eq. 3 as follows:

$$\begin{aligned}
&\frac{1}{\Delta x \Delta y \Delta z} \int\limits_{\partial\Omega_u^{in}(\mathrm{l,m,n})} \left(wu - \nu \frac{\partial u}{\partial z} \right) \, \mathrm{d}x'\mathrm{d}y' - \frac{1}{\Delta x \Delta y \Delta z} \int\limits_{\partial\Omega_u^{out}(\mathrm{l,m,n})} \left(wu - \nu \frac{\partial u}{\partial z} \right) \, \mathrm{d}x'\mathrm{d}y' \\
&= \frac{1}{\Delta z} \Bigg(\frac{\overline{w}(\mathrm{l,m,n}) + \overline{w}(\mathrm{l-1,m,n})}{2} \, \frac{\overline{u}(\mathrm{l,m,n}) + \overline{u}(\mathrm{l,m,n-1})}{2} - \nu \frac{\overline{u}(\mathrm{l,m,n}) - \overline{u}(\mathrm{l,m,n-1})}{\Delta z} \\
&\quad - \frac{\overline{w}(\mathrm{l,m,n+1}) + \overline{w}(\mathrm{l-1,m,n+1})}{2} \, \frac{\overline{u}(\mathrm{l,m,n}) + \overline{u}(\mathrm{l,m,n+1})}{2} + \nu \frac{\overline{u}(\mathrm{l,m,n+1}) - \overline{u}(\mathrm{l,m,n})}{\Delta z} \Bigg) \\
&\quad + \frac{1}{\Delta z} \left(\tau_{wu}^{in}(\mathrm{l,m,n}) - \tau_{wu}^{out}(\mathrm{l,m,n}) \right), \qquad (4)
\end{aligned}$$

where $\partial\Omega_u^{in}(\mathrm{l,m,n})$ and $\partial\Omega_u^{out}(\mathrm{l,m,n})$ are, respectively, the lower and upper boundaries of the control volume corresponding to the u-component $\Omega_u(\mathrm{l,m,n})$, representing two different subsets of the total control volume area $\partial\Omega_u(\mathrm{l,m,n})$.

$\tau_{wu}^{in}(\mathrm{l,m,n})$ and $\tau_{wu}^{out}(\mathrm{l,m,n})$, in turn, are unknown terms correcting for the unresolved physics and instantaneous discretization errors in the finite-volume advection and viscous stress terms at the lower and upper control volume boundaries respectively. Following Eq. 4, $\tau_{wu}^{in}(\mathrm{l,m,n})$ and $\tau_{wu}^{out}(\mathrm{l,m,n})$ are defined as:

$$\begin{aligned}
\tau_{wu}^{in}(\mathrm{l,m,n}) &\equiv \frac{1}{\Delta x \Delta y} \int\limits_{\partial\Omega_u^{in}(\mathrm{l,m,n})} \left(wu - \nu \frac{\partial u}{\partial z} \right) \, \mathrm{d}x'\mathrm{d}y' - \frac{\overline{w}(\mathrm{l,m,n}) + \overline{w}(\mathrm{l-1,m,n})}{2} \, \frac{\overline{u}(\mathrm{l,m,n}) + \overline{u}(\mathrm{l,m,n-1})}{2} \\
&\quad + \nu \frac{\overline{u}(\mathrm{l,m,n}) - \overline{u}(\mathrm{l,m,n-1})}{\Delta z}, \\
\tau_{wu}^{out}(\mathrm{l,m,n}) &\equiv \frac{1}{\Delta x \Delta y} \int\limits_{\partial\Omega_u^{out}(\mathrm{l,m,n})} \left(wu - \nu \frac{\partial u}{\partial z} \right) \, \mathrm{d}x'\mathrm{d}y' - \frac{\overline{w}(\mathrm{l,m,n+1}) + \overline{w}(\mathrm{l-1,m,n+1})}{2} \, \frac{\overline{u}(\mathrm{l,m,n}) + \overline{u}(\mathrm{l,m,n+1})}{2} \\
&\quad + \nu \frac{\overline{u}(\mathrm{l,m,n+1}) - \overline{u}(\mathrm{l,m,n})}{\Delta z}, \qquad (5)
\end{aligned}$$

The correction terms for the other control volumes and transport components are defined in a similar manner. In the remainder of the paper, we will denote the complete correction terms with the shorthand notation τ_{ij}^{in} and τ_{ij}^{out}. It is these complete terms we aim to predict with an ANN-based LES SGS model. We chose their definition such that the ANNs can replace traditional SGS models like Smagorinsky without large changes in the encompassing code (Sect. 8). To fully solve Eq. 3, *after* training our ANN SGS model only makes use of information available in actual finite-volume LES: for its inputs, it relies only on the resolved flow fields $\overline{u}$, $\overline{v}$, $\overline{w}$ and their boundary conditions (Sect. 5).

3 Direct numerical simulation test case

To generate the training data for the ANN, we relied on a DNS of incompressible neutral channel flow (with Friction Reynolds number Re_τ being equal to 590) based on Moser et al. (1999). We used the high-order DNS and finite-volume LES code MicroHH (v2.0) as simulation tool, which has been verified previously for this flow case (van Heerwaarden et al., 2017). A neutral channel flow is a turbulent flow bounded by walls at both the bottom and top of the domain (no-slip boundary conditions), with a mean flow characterized by a symmetric horizontally averaged vertical profile (Fig. (1)). In the horizontal directions, periodic boundary conditions were applied and a constant volume-averaged velocity ($U_f = 0.11\text{m s}^{-1}$) was enforced by dynamically adjusting the pressure gradient.

180 We stored in total 31 3D flow fields of the wind velocity fields u, v, w at time intervals of 60s after the flow reached steady state. This time interval was large enough to ensure that subsequent stored flow fields were (nearly) independent, which is preferable for the training and testing of the neural networks (Sect. 6). More details about the used simulation set-up and simulation code can be found in Table 1 and van Heerwaarden et al. (2017).

Table 1. Simulation specifications for direct numerical simulation of incompressible neutral channel flow test case we used to generate the training data (Sect. 4). Here, δ $[\text{m}]$ refers to the channel-half width. Additional details about the employed code (MicroHH v2.0) are given in van Heerwaarden et al. (2017).

Friction Reynolds number Re_τ	590
Boundary conditions	horizontal directions (x,y): periodic, vertical direction (z): no-slip
Domain size(x,y,z)	$2\pi\delta,\ \pi\delta,\ 2\delta$
Kinematic viscosity ν	$1.0 * 10^{-5}$ $[\text{m}^2\ \text{s}^{-1}]$
Prescribed volume-averaged velocity U_f	0.11 $[\text{m s}^{-1}]$
Grid resolution (x,y,z)	$768,\ 384,\ 256$ (stretched in vertical)
Employed grid	staggered Arakawa C-grid
Spatial discretization	fourth-order interpolation scheme
Time discretization	three-stage, third-order Runge-Kutta scheme

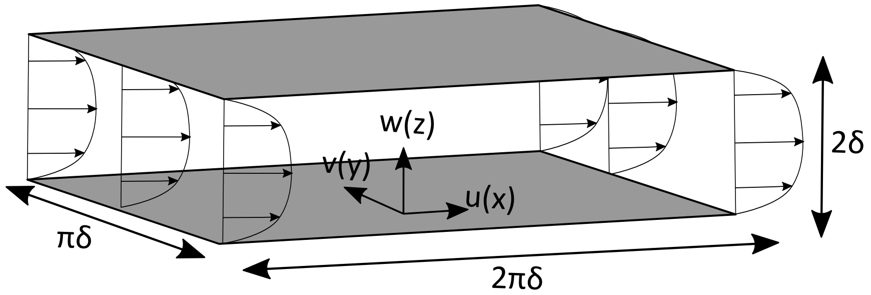


Figure 1. Sketch of simulated turbulent channel flow. Here, δ $[\text{m}]$ refers to the channel-half width.

4 Training data generation

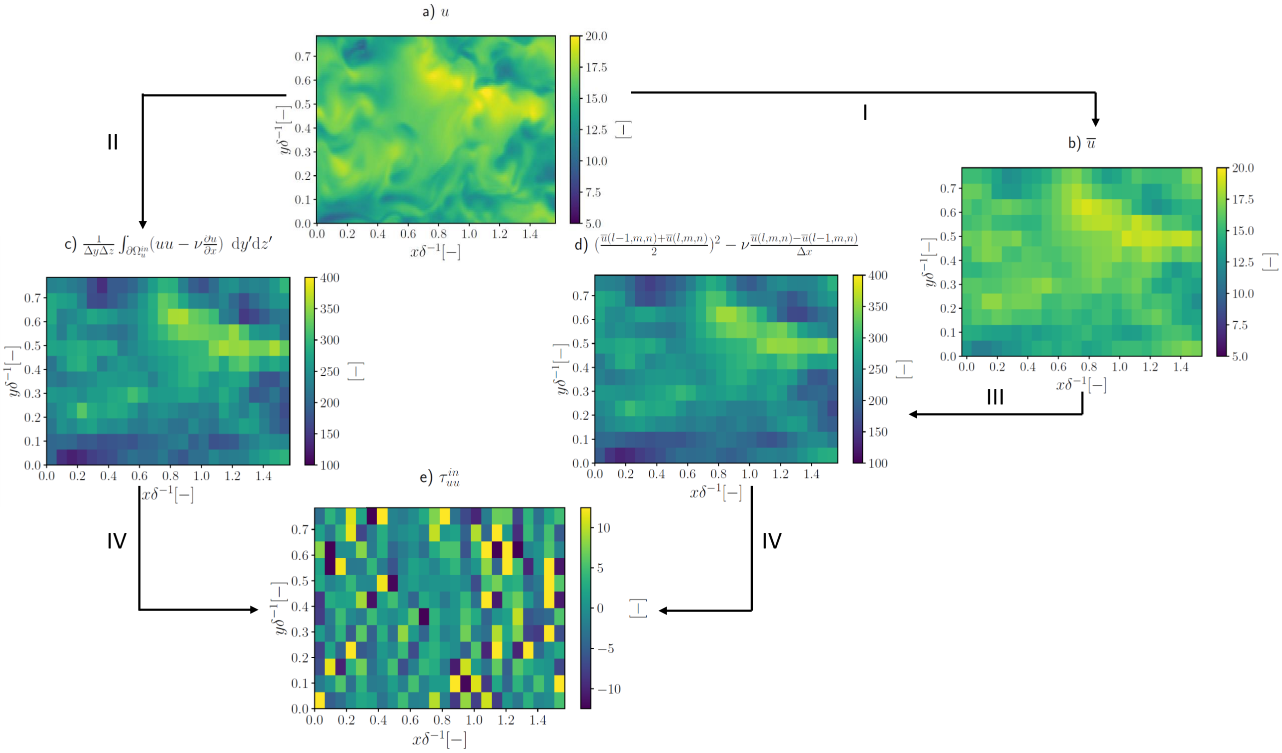


Figure 2. Four-step filtering procedure applied to generate the training data used in this study. As an example, the panels show for each step in the calculation of τ_{uu}^{in} the horizontal cross-sections at a height of 0.109δ for part of the domain. The headers above the panels indicate the plotted quantities. All the values have been made dimensionless with u_τ^{-1}, ${u_\tau}^{-2}$, and δ^{-1}. Here, δ [m] refers to the channel-half width and u_τ [m s^{-1}] to the friction velocity. More details about the indicated steps can be found in Sect. 4.

We will describe in the following the filtering procedure we applied to generate the training data for our ANNs. Using the framework outlined in Sect. 2 as a basis, from the 31 stored DNS flow fields (Sect. 3) we calculated consistent pairs of i) low-resolution flow velocity fields $\overline{u_j}$ (that served as input for the ANN; panel (b) Fig. 2), and ii) correction terms τ_{ij}^{in}, τ_{ij}^{out} (that served as the ground truth for the ANN predictions; panel (e) Fig. 2). We used these pairs to train the ANNs in a supervised manner, allowing the ANN to learn predicting the correction terms based only on the low-resolution flow fields (Sect. 6). This filtering procedure involved four different steps in total (Fig. 2), which are described in more detail below:

Step I: Downscale u_j ($768 \times 384 \times 256$) to $\overline{u_j}$ using Eq. 2. As a typical LES resolution, we decided to focus only on one equidistant lower-resolution grid of $96 \times 48 \times 64$ (x × y × z) cells. The a priori and a posteriori results of this one resolution already showed important strengths and challenges of the proposed approach, especially because it is relatively coarse close to the walls.

Step II: Calculate from u_j the *exact* advection and viscous stress terms (first term right-hand side of definitions given in Eq. 5). To this end, we integrated $u_i u_j$ and $\nu \frac{\partial u_j}{\partial x_i}$ from the fine grid over the grid faces of the control volumes $\Omega_{j(\mathrm{l,m,n})}$ (Fig. 3). As a consequence of the employed staggered finite-volume Arakawa C-grid, the location of the control volume (and thus the corresponding grid face) differed for each component of u_j.

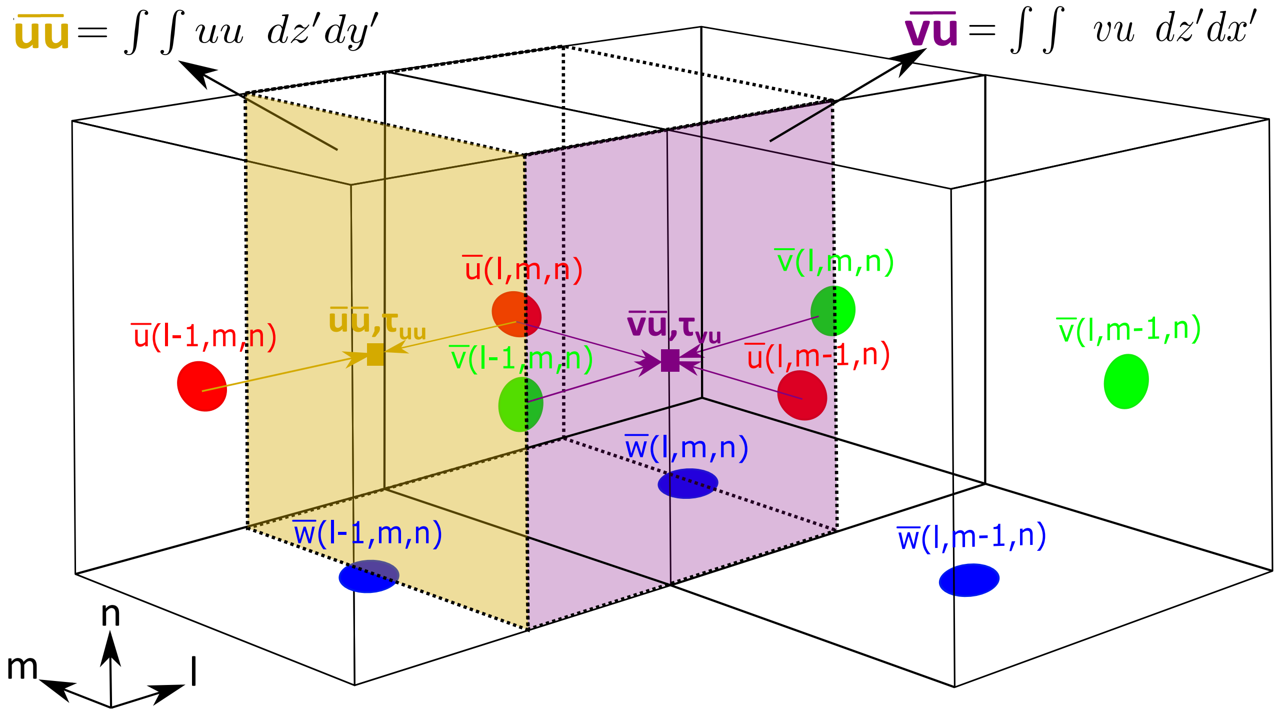

$\overline{\mathbf{uu}} = \int\int uu\ \ dz'dy'$
$\overline{\mathbf{vu}} = \int\int\ \ vu\ \ dz'dx'$
$\overline{\mathrm{u}}$(l,m,n)
$\overline{\mathrm{v}}$(l,m,n)
$\overline{\mathrm{v}}$(l,m-1,n)
$\overline{\mathrm{u}}$(l-1,m,n)
$\overline{\mathrm{uu}}$,τ_{uu}
$\overline{\mathrm{vu}}$,τ_{vu}
$\overline{\mathrm{v}}$(l-1,m,n)
$\overline{\mathrm{u}}$(l,m-1,n)
$\overline{\mathrm{w}}$(l,m,n)
$\overline{\mathrm{w}}$(l-1,m,n)
$\overline{\mathrm{w}}$(l,m-1,n)
n
m
l

Figure 3. Three-dimensional visualization of filtering procedure step II and III for a grid cell with indices $(\mathrm{l,m,n})$ for the staggered, Arakawa C-grid employed. As an example, a calculation of surface-integrated $\overline{uu}, \overline{uv}$ and $\overline{u}\,\overline{u}, \overline{u}\,\overline{v}$ are shown (for the sake of clarity, the calculation of the corresponding viscous fluxes is not shown). The dotted black line indicates the control volume for u, the solid black lines the individual grid cells, the shaded planes the areas over which the integrals corresponding to $\overline{uu}, \overline{vu}$ were evaluated, and the colored arrows the interpolations taken to calculate $\overline{u}\,\overline{v}$ and $\overline{u}\,\overline{u}$.

Step III: Using only the coarse-grained flow fields $\overline{u_j}$, reconstruct the advection and viscous stress terms with linear second-order interpolation methods (Fig. 3; second and third term right-hand side of definitions in Eq. 5). This ensured that we reproduced important interpolation errors made in an actual LES with staggered grids. These errors appeared to remove a substantial fraction of the turbulent energy remaining after the filtering operation, reflecting their detrimental impact on the smallest resolved scales (Fig. 4).

Step IV: Calculate the correction terms τ_{ij}^{in}, τ_{ij}^{out} via Eq. 5, which at this stage simply boils down to subtracting the result of step III from the result of step II.

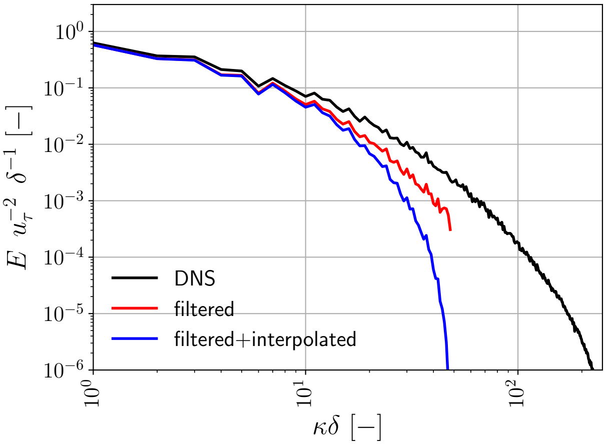


Figure 4. Example streamwise power spectra of u, taken at a height of 0.109δ (i.e. log-layer). Here, δ $[\mathrm{m}]$ refers to the channel-half width. The power spectral density E on the vertical axis has been normalized by δ^{-1} and ${u_\tau}^{-2}$, where u_τ $[\mathrm{m\ s^{-1}}]$ is the friction velocity. Here, the black line corresponds to the power spectrum of the DNS field (panel (a) Fig. 2), the red line to the power spectrum remaining after the finite-volume LES filter (Eq. 2) has been applied (step I filtering procedure; panel (b) Fig. 2), and the blue line the spectrum remaining after both the finite-volume filter and the interpolations required on the coarse LES grid to calculate τ_{uu}^{in} (step III filtering procedure, panel (d) Fig. 2).

5 ANN architecture

We used feed-forward, fully-connected ANNs with a single hidden layer and without skip connections, to predict the correction terms τ_{ij}^{in} and τ_{ij}^{out} with the resolved flow fields u_j as input. These are simple ANNs that facilitate computationally fast evaluations and easy implementation. We did not use deeper, more sophisticated ANNs to limit the computational cost involved in making predictions with the ANN as much as possible. This computational cost is critical for the affordability of an ANN SGS model in an actual LES simulation (Sect. 8).

To introduce non-linearity in the ANN, we used as an activation function the leaky rectified linear unit (ReLu) function (Maas et al., 2013) with the constant α set to the common value 0.2. This non-linear activation function, together with the linear matrix-vector multiplications and bias parameter addition, defines the entire functional form of the ANN.

Similar to conventional LES SGS models, the ANN should preferably act on a small subdomain of the full grid to facilitate integration in our simulation code MicroHH, which uses domain decomposition for distributed memory computing. We consequently predicted with the ANN only the $\tau_{ij}^{in/out}$-values associated with one grid cell $(\mathrm{l},\mathrm{m},\mathrm{n})$ at a time. As input to the ANN, we used the locally resolved flow fields $\overline{u_j}$ in a $5\times5\times5$ stencil surrounding the grid cell for which we predict $\tau_{ij}^{in/out}$. Similar

https://doi.org/10.5194/gmd-2020-289
Preprint. Discussion started: 10 November 2020

to Cheng et al. (2019) and Yang et al. (2019), we opted not to make our inputs Galilean/rotational invariant as the walls already provide an intrinsic coordinate system and velocity reference.

To select appropriate $5 \times 5 \times 5$ inputs stencils close to the boundaries of the domain, we made use of the horizontal periodic boundary conditions and the vertical no-slip conditions. We encoded the no-slip conditions in the input stencils by mirroring $\overline{u_j}$ over the walls, such that $\overline{u_j}$ linearly interpolated to the wall was $0 \ \mathrm{m \ s^{-1}}$. This may have helped the ANN to distinguish the near-wall region from the bulk of the flow, potentially removing the need for separate subgrid and wall models.

Using the $5 \times 5 \times 5$ stencils in combination with the employed staggered Arakawa C-grid, an asymmetric bias is introduced in the ANN input and output variables if no special care is taken. We overcame this issue by combining three separate single-layer ANNs, where each one corresponded to one of the three control volumes considered (Sect. 4). Here, each received a stencil with slightly adjusted dimensions, and predicted only the correction terms ($\tau_{ij}^{in}, \tau_{ij}^{out}$) corresponding to the considered control volume (resulting in 6 outputs per ANN; Fig. 5). This ensured symmetry in the inputs and outputs of the ANN (Fig. 6 panel (a)), and did not increase the computational effort involved in evaluating the ANN after training.

In fact, this allowed us to reduce the number of ANN evaluations in the a posteriori simulation (Sect. 8) by almost a factor 2. Except for close to the walls, evaluating the ANN with a checkerboard-like pattern was sufficient to obtain all the needed correction terms (Fig. 6 panel (b)). Close to the walls, we did require (sometimes partial) ANN evaluations at every grid cell to calculate all needed correction terms: the checkerboard-like pattern does not provide all the correction terms at the edges of the domain. In the horizontal directions, we could make use of the periodic boundary conditions at the edges of the domain.

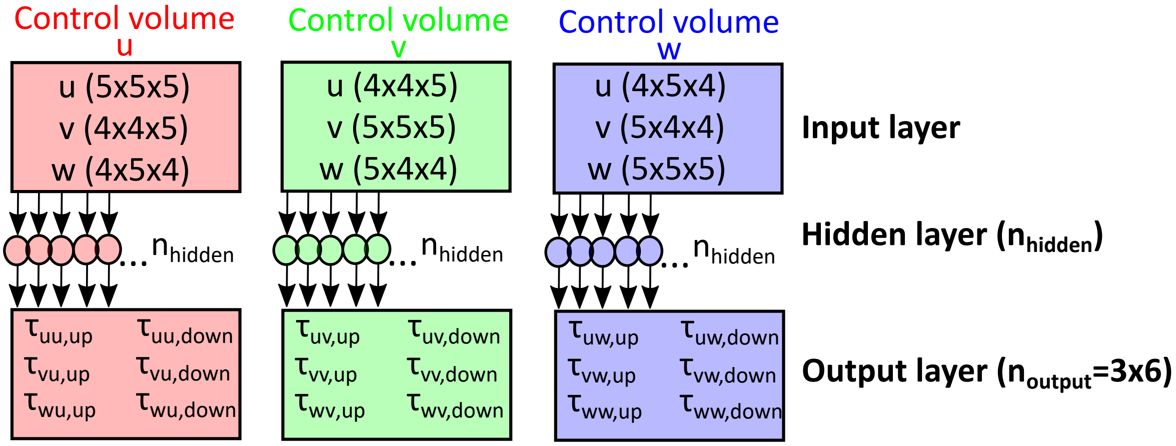


Figure 5. Architecture of ANN-framework used in this study. We combined three separate ANNs that each correspond to one of the three considered control volumes. For more information, please refer to Sect. 5.

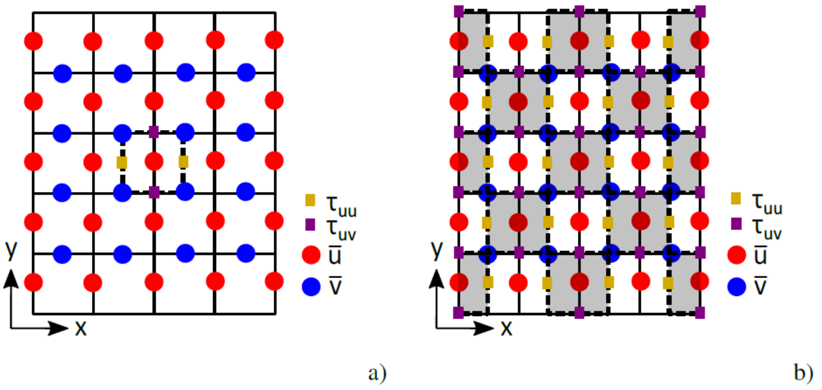


Figure 6. Panel a): Example two-dimensional input stencil of $\overline{u}$,$\overline{v}$ that the ANN corresponding to the control volume of u receives, together with four of its outputs (i.e. τ_{uu}^{in}, τ_{uu}^{out}, τ_{uv}^{in}, τ_{uv}^{out}). Panel b): Two-dimensional visualization of the way we evaluated the ANN during a posteriori simulations. By evaluating the ANN in checkerboard-like pattern (i.e. only evaluating the grey-shaded grid cells) and making use of the periodic boundary conditions, we could calculate all needed correction terms except those close to the walls.

6 ANN training

We trained the employed ANNs (Fig. 5) using the training data (consisting of corresponding local $5 \times 5 \times 5$ $\overline{u_j}$ fields and correction terms $\tau_{ij}^{in/out}$; Sect. 4) we generated from 31 stored flow fields (Sect. 3). With the LES resolution we employed $(64 \times 48 \times 96)$, we could extract 294.912 unique samples from each flow field. Of the 31 stored flow fields, we used 25 for training (i.e. $7.372.800$ samples), 3 for validation during training and tuning of the the hyperparameters (i.e. 884.736 samples), and also 3 for the a priori test (Sect. 7).

During training, the main objective is usually to minimize the so-called cost or loss function, which can be chosen freely. We defined it as the mean squared error (MSE) between the 18 DNS-derived $\tau_{ij,DNS}^{in/out}$-components (Sect. 4), and the 18 ANN-predicted $\tau_{ij,ANN}^{in/out}$-components (Sect. 5), combining the results from all three separate ANNs (Sect. 5). By minimizing this MSE, the ANN should 'learn' to represent the unresolved physics and instantaneous discretization errors incorporated in $\tau_{ij,DNS}^{in/out}$. We did not add so-called regularization terms to the loss function, which are typically added when the ANN shows sign of over-fitting (e.g. a much higher loss on the validation set than on the training set). Without regularization, we already observed good convergence of both the training and test error without signs of over-fitting for all the ANNs we tested (Fig.7).

We did implement preferential sampling near the walls: during training we selected the five horizontal layers closest to the bottom and top wall more often than the other horizontal layers (starting from the bottom/top wall towards the center of the channel, respectively with a factor 10, 8, 6, 4, and 2). Our training data set is unbalanced without this preferential sampling, as most samples would then originate from the bulk of the flow where the physics are different than close to the walls. The preferential sampling restores the balance in the training data set between the physics near the wall and the bulk of the flow, potentially allowing the ANN to represent both properly without the need for a separate wall model.

To train our ANNs, we used TensorFlow (v 1.12.0), an open-source machine learning framework (Abadi et al., 2016). We relied on the backpropagation algorithm (Rumelhart et al., 1986) incorporated within TensorFlow to minimize the loss function. In short, this algorithm calculates for a group of training samples (i.e. the so-called training batch) the gradient of the loss function with respect to all the trainable parameters of the ANN (i.e. the weights and bias parameters). Subsequently, the chosen training optimizer uses these gradients to adjust the parameters accordingly. The magnitude of these adjustments are governed by the chosen learning rate η. We chose the popular ADAM optimizer (Kingma and Ba, 2014) with a relatively low value for η (0.0001) and a relatively large batch size of 1000. As our training data contains a high amount of noise inherent to turbulence, these parameter choices were in our case needed to stabilize the training results. Furthermore, in line with common practice, we normalized all the inputs and 'true' outputs with their means and standard deviations. This improved the convergence during training and accelerated learning.

In Table 2 we give an overview of all the hyperparameters and settings we used. The chosen initialization methods for the weights and bias parameter are standard for the architecture and activation function we selected.

Table 2. Fixed hyperparameters and settings used in the ANNs we trained. Here, # means 'number of'.

# training iterations (epochs)	500.000 ($\approx$ 38 epochs taking into account the preferential sampling)
# hidden layers	1
Batch size	1000
Loss function	mean squared error, no regularization
Activation function	Leaky ReLu with $\alpha = 0.2$ (Maas et al., 2013)
Optimizer	ADAM with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-08$ (Kingma and Ba, 2014)
Learning rate η	0.0001
Normalization	z-score ($\frac{\mathrm{value-mean}}{\mathrm{standard\ deviation}}$)
Weight/kernel initializer	He uniform variance scaling initializer (He et al., 2015)
Bias initializer	zeros initializer

We performed a more extensive sensitivity analysis with the number of neurons in the hidden layer, $\mathrm{n_{hidden}}$, as it is for our architecture a good measure of the model complexity. In general we found that increasing $\mathrm{n_{hidden}}$, and thus increasing the model complexity, improved the reduction of the loss function without showing signs of over-fitting (Fig. 7). However, the improvement in training loss reduction clearly reduced with increasing model complexity, while a higher model complexity increases the computational cost of the ANN SGS model. In the next sections we will therefore focus on the results we obtained with $\mathrm{n_{hidden}} = 64$, as a reasonable compromise between accuracy and total computational cost.

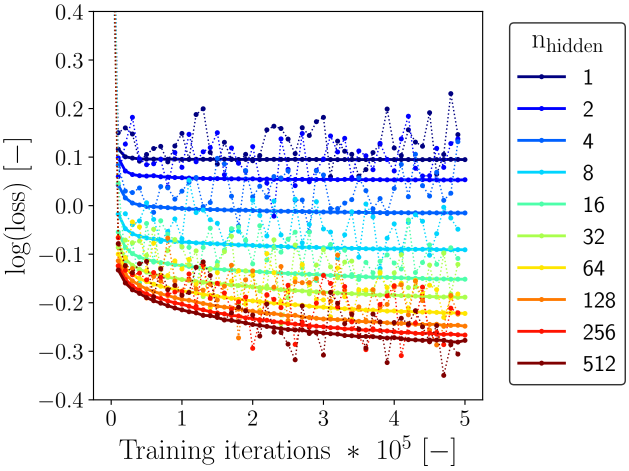


Figure 7. Evolution of the loss corresponding to the considered training batches (dotted lines) and the 3 validation flow fields (solid lines) for a changing number of neurons in the single hidden layer as a function of training iteration. To improve readability and keep the total computational effort involved in the training feasible, we show here both losses only for every 10.000 iterations instead of every single iteration.

7 ANN a priori test

To test the performance of the ANN SGS models, we first performed an a priori test by directly comparing the ANN predictions and DNS-derived correction terms for 3 previously unseen flow fields of u, v, w. In this comparison, we included the subgrid fluxes $\tau_{ij,Smag}$ predicted with the popular Smagorinsky SGS model (with $c_s = 0.1$). In line with usual practice, we augmented the Smagorinsky SGS model with an ad-hoc Van Driest (Van Driest, 1956) wall-damping function (with $A = 26$) to (partly) compensate for its known over-dissipative behaviour (e.g. Pope, 2001; Sagaut, 2006).

Furthermore, to facilitate easier comparison with the Smagorinsky model, for the ANN and DNS results we combined the two separate correction terms $\tau_{ij}^{in}, \tau_{ij}^{out}$. In the remainder of the paper we will denote the resulting combined fields as $\tau_{ij,ANN}$ and $\tau_{ij,DNS}$ respectively, which each contain the same 9 components as $\tau_{ij,Smag}$. We did this in accordance with the way how we evaluated the ANNs within our CFD-code MicroHH during the a posteriori test (Sect. 5).

We focused in the a priori test particularly on the τ_{wu}-component in the log-layer. In channel flow, it is mainly the vertical gradient of the τ_{wu}-component that has to balance the imposed horizontal pressure gradient (e.g. Pope, 2001). τ_{wu} is therefore critical for the quality of the achieved steady state solution and its statistics. The log-layer, in turn, is especially interesting because of its universal character. In the log-layer, the horizontally averaged profiles of the mean velocity and Reynolds stress tensor components becomes partly independent of the Reynolds number when properly scaled with wall units (e.g. Pope, 2001).

Furthermore, to get some more insight into the behaviour of our ANNs, we calculated for every input variable in the $5 \times 5 \times 5$ stencils the so-called permutation feature importance (e.g. Fisher et al., 2019; Molnar, 2019; Breiman, 2001) associated with

https://doi.org/10.5194/gmd-2020-289
Preprint. Discussion started: 10 November 2020

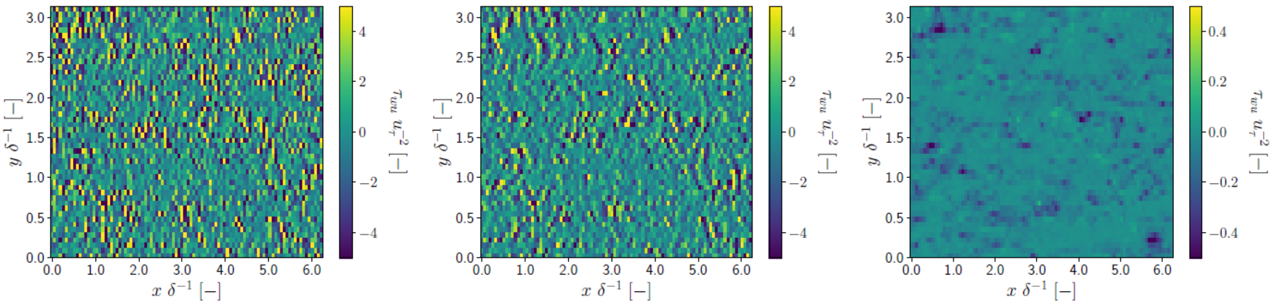
predicting τ_{wu}^{in} and τ_{wu}^{out} in the log-layer (Sect. 7.3). The permutation feature importance indicates how important a certain input variable is for the prediction quality of the ANN: the higher it is, the more important that variable is for the ANN.

Specifically, the permutation feature importance measures by which factor the prediction error (in our case measured as the root-mean square error between the DNS-values and ANN predictions) increases when the information contained in that input variable is destroyed, while the information in the other input variables is retained. We destroyed the information in each input variable by randomly shuffling it in the corresponding horizontal plane. Besides that, we averaged the calculated permutation feature importances over all the 3 testing flow fields and over 10 different random shufflings, to stabilize the results. We intentionally chose not to shuffle the input variables along different heights. Because of the strong mean vertical gradient in u, this would possibly introduce an unrealistic bias into the calculated permutation feature importances. We do emphasize that the permutation feature importances are likely affected by the correlations existing in our input data. The permutation feature importances we report therefore need to interpreted with caution.

7.1 Performance

The ANN predictions $\tau_{wu,ANN}$ (with $\mathrm{n_{hidden}} = 64$) in the log-layer generally show excellent agreement with $\tau_{wu,DNS}$ (Fig. 8 and 9). Especially the consistency we found in horizontal cross-sections (Fig. 8) is striking given the noisy spatial patterns of $\tau_{wu,DNS}$, which $\tau_{wu,ANN}$ reproduces quite accurately both qualitatively and quantitatively. In agreement with this, $\tau_{wu,ANN}$ also reproduces quite well the distribution and spectrum of $\tau_{wu,DNS}$ (Fig. 9). The notable high spectral content at high wave modes, is mainly caused by the coarsening we applied in the training data generation. The coarsening tends to remove the local spatial structure present at high resolutions, resulting in strong variability at the smallest resolved spatial scales. We note that the found correspondence between $\tau_{wu,ANN}$ and $\tau_{wu,DNS}$ in the log-layer is in agreement with the result of Cheng et al. (2019) and Gamahara and Hattori (2017), despite that their training data generation procedure was slightly different from ours (Sect. 2 and 4).

From the tails of the distribution and the high wave modes of the spectra however (Fig. 9), it is apparent that $\tau_{wu,ANN}$ does still slightly underestimate the extremes at small spatial scales characteristic of $\tau_{wu,DNS}$. Probably, these extremes were hard to predict accurately because of their high stochastic nature and inherent rare occurrence. Yang et al. (2019) identified this issue in the context of an ANN-based LES wall model, and found that this issue persisted even when the errors were weighted inversely proportional to their PDF (i.e. giving extreme values larger weights in the loss function).



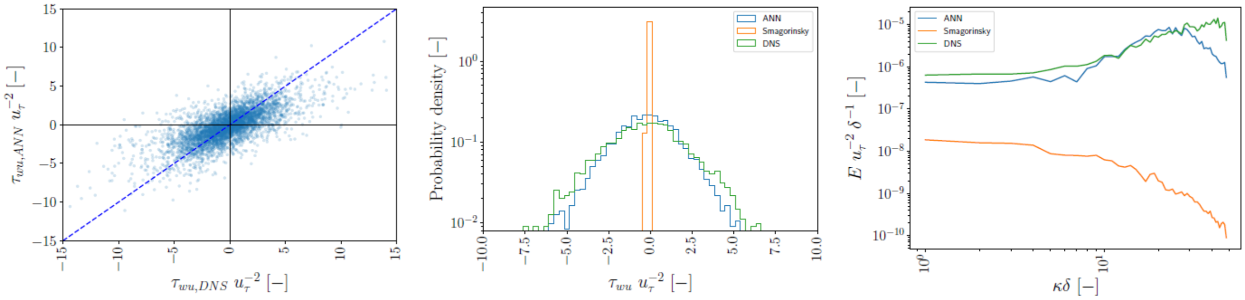
a) DNS

b) ANN ($n_{\text{hidden}} = 64$)

c) Smagorinsky

Figure 8. Horizontal cross-sections of τ_{wu} in the log-layer ($0.09375\frac{z}{\delta}$ $(55.3125z^+)$) for a representative flow field not used to train and validate the ANNs. All values are normalized by the friction velocity u_τ and half-channel width δ.

.



a)

b)

c)

Figure 9. Performance of $\tau_{wu,ANN}$ (with $n_{\text{hidden}} = 64$) in the log-layer ($0.09375\frac{z}{\delta}$ $(55.3125z^+)$) for a representative flow field not used to train and validate the ANNs. Panel (a) shows the corresponding scatter-plot between $\tau_{wu,ANN}$ and $\tau_{wu,DNS}$, where the dotted blue line indicates the 1:1 line. Panel (b) shows the probability density functions, and panel (c) the streamwise spectra averaged in the spanwise direction. $\tau_{wu,ANN}$ and $\tau_{wu,DNS}$ have been normalized by the friction velocity u_τ^{-2}. The power spectral density E on the vertical axis in panel (c) has been normalized by δ^{-1} and ${u_\tau}^{-2}$. As a reference, in panel (b) and (c) $\tau_{wu,smag}$ is shown as well.

Extending our focus from the log-layer to the whole vertical domain, we see that, especially near the walls, $\tau_{wu,ANN}$ still had some difficulty to accurately represent the horizontally averaged vertical profile of $\tau_{wu,DNS}$ (Fig. 10). We note that, in our training data, for $\tau_{wu,DNS}$ the horizontally averaged flux was small compared to its point-wise fluctuations. As a result, the loss associated with $\tau_{wu,DNS}$ was probably more sensitive to the point-wise fluctuations than the average flux.

https://doi.org/10.5194/gmd-2020-289
Preprint. Discussion started: 10 November 2020

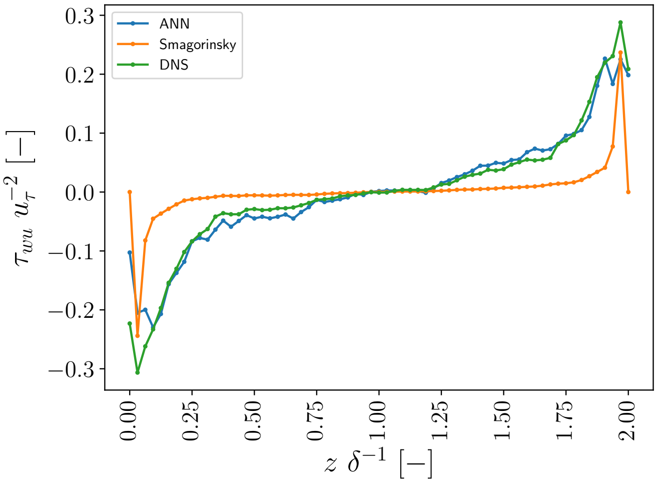


Figure 10. Horizontally averaged vertical profiles of $\tau_{wu,DNS}$, $\tau_{wu,ANN}$, and $\tau_{wu,smag}$ at one representative time step not used to train and validate the ANNs. All values are normalized by the friction velocity u_τ^{-2} and half-channel width δ^{-1}.

Extending our focus further towards all components, we found that in general the ANN correlated well with the DNS subgrid fluxes (Table 3; mostly ρ $0.6-1.0$). Looking more closely at the found correlations, we did find that the correlations differed depending on the channel height. Closer to the walls, the correlations generally slightly decreased compared to the middle of the channel. We minimized this difference by implementing the preferential sampling discussed before (Sect. 6). Furthermore, some components (i.e. the isotropic components and the components associated with the u control volume) were clearly better predicted than others: this was likely related to differences in their magnitude that persisted even after the applied normalization (i.e. the same normalization was applied over the entire domain, meaning that some components with strong vertical gradients still contained more extreme values than components without a clear vertical gradient), and differences in their stochastic variability and consequent signal-to-noise ratio.

One clear outlier is τ_{wu} at the first vertical level (with $\rho = 0.358$), which appeared to be most difficult to predict. This component was located at the bottom wall because of the staggered grid orientation, and consequently only the viscous flux contributed. As a consequence, the target DNS values and input patterns were different than for other vertical levels and components, making it hard for the ANN to give accurate predictions. Still, the magnitude of the ANN predictions matched reasonably well the DNS values (not shown).

Table 3. Pearson correlation coefficients between ANN predictions ($\mathrm{n_{hidden}} = 64$) and DNS values for 3 test flow fields, for every τ_{ij}-component and multiple heights (where the indicated height is of the corresponding staggered grid box center and thus does not necessarily match the height of the component). Here, the j-index refers to the considered control volume (Sect. 2). Note that the $\tau_{uw,vw}$-components are left out at the first vertical level, as these are due to the staggered grid located exactly at the bottom wall. At the bottom-wall we imposed a no-slip boundary-condition, meaning that these components are by definition 0.

height	τ_{uu}	τ_{vu}	τ_{wu}	τ_{uv}	τ_{vv}	τ_{wv}	τ_{uw}	τ_{vw}	τ_{ww}
all	0.840	0.824	0.769	0.730	0.783	0.718	0.692	0.652	0.778
0.015625δ ($9.21875z^{+}$)	0.928	0.878	0.358	0.845	0.756	0.668	–	–	0.540
0.046875δ ($27.65625z^{+}$)	0.790	0.838	0.793	0.732	0.716	0.704	0.663	0.605	0.678
0.078125δ ($46.09375z^{+}$)	0.771	0.818	0.728	0.704	0.736	0.712	0.596	0.632	0.694
0.109375δ ($64.53125z^{+}$)	0.757	0.808	0.717	0.689	0.744	0.700	0.608	0.607	0.687
0.265625δ ($156.71875z^{+}$)	0.767	0.806	0.739	0.700	0.751	0.732	0.652	0.646	0.743
0.765625δ	0.842	0.838	0.811	0.769	0.753	0.745	0.762	0.712	0.793
1.234375δ	0.846	0.847	0.818	0.782	0.752	0.708	0.771	0.676	0.796
1.734375δ	0.781	0.815	0.761	0.721	0.756	0.747	0.681	0.700	0.738
1.890625δ	0.762	0.813	0.715	0.701	0.729	0.719	0.619	0.655	0.691
1.921875δ	0.769	0.824	0.713	0.722	0.731	0.703	0.609	0.642	0.678
1.953125δ	0.798	0.843	0.728	0.746	0.734	0.704	0.609	0.640	0.706
1.984375δ	0.933	0.884	0.806	0.830	0.775	0.712	0.674	0.622	0.580

7.2 Comparison Smagorinsky

Focusing on the horizontally averaged vertical profiles, we found that $\tau_{wu,smag}$ does not compare well with $\tau_{wu,DNS}$ (Fig. 10). Except close to the walls, the Smagorinsky SGS model strongly underestimates the horizontally averaged τ_{wu}. We emphasize that the correspondence close to the walls was only achieved because of the implemented ad-hoc Van Driest wall damping function (Van Driest, 1956).

Considering the individual grid points, the a priori performance of the Smagorinsky SGS model is in sharp contrast with the a priori ANN performance: $\tau_{ij,Smag}$ shows barely any agreement with the DNS values both qualitatively and quantitatively (Fig. 8 and 9). The poor point-wise a-priori performance of Smagorinsky is not surprising and well-known in literature (e.g. Clark et al., 1979; McMillan and Ferziger, 1979; Liu et al., 1994). In our case though, the point-wise a-priori performance of Smagorinsky is worse than usually documented: the found correlations with DNS in our study (mostly $\rho = 0.0$ for individual heights and all components, not shown) are lower than reported before (where $\rho =\sim 0...0.4$; Cheng et al. (2019); Clark et al. (1979); McMillan and Ferziger (1979); Liu et al. (1994)). Furthermore, $\tau_{ij,Smag}$ is off by more than one order of magnitude and is too smooth (Fig. 8 and 9): in comparison to $\tau_{ij,DNS}$, the PDF is narrower (Fig. 9 panel b), and the spectral content in $\tau_{ij,Smag}$ is smaller and skewed towards low wave modes (Fig. 9 panel c).

This exacerbated point-wise a priori performance of the Smagorinsky SGS model is caused by our alternative definition for τ_{ij}, which, in contrast to the commonly defined τ_{ij}, compensates for all the instantaneous discretization effects introduced by the staggered finite-volumes in both the advection and viscous flux terms (Sect. 2). As these discretization effects remove a large part of the variance present in the LES (Figure 4), our τ_{ij} inherently contains a large amount of variance that is not represented by Smagorinsky.

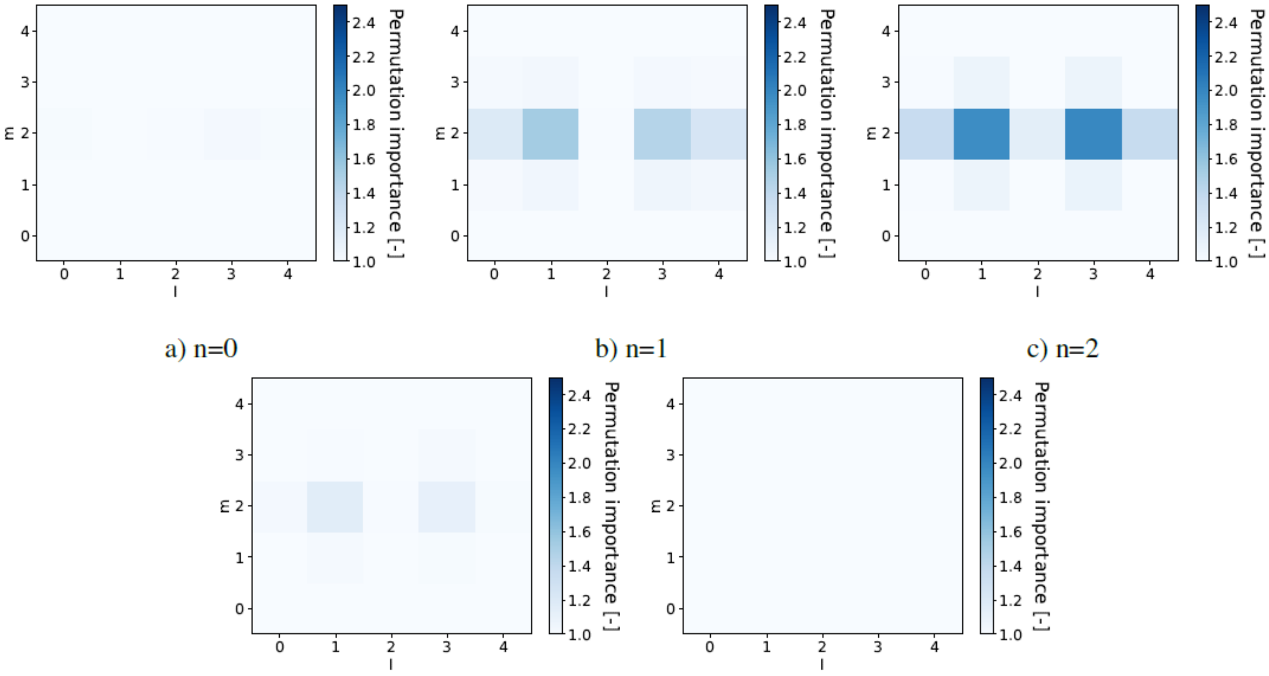
All in all, our ANN SGS model is clearly better able to represent τ_{ij} in an a priori test than the Smagorinsky SGS model. This shows the promise ANN SGS models like ours could have to construct more accurate SGS models, if their a priori potential can be successfully leveraged in a posteriori simulations.

7.3 Permutation feature importance

For τ_{wu}^{in} and τ_{wu}^{out} in the log-layer, the calculated permutation feature importances (shown as an example for u in Fig. 11 and 12) suggest that the input variables most important to the ANN (with $n_{hidden} = 64$) are generally located close to the considered correction term (with the notable exception of the middle column in the u-velocity input stencil).

Comparing the calculated feature importances corresponding to τ_{wu}^{in} and τ_{wu}^{out}, we see a striking shift in the vertical. For the u-input stencil the u-velocity input stencil (Fig. 11-12), it is evident that the vertical patterns corresponding to τ_{wu}^{in} and τ_{wu}^{out} are nearly mirrored versions of each other. Besides that, the most important input variables also seem to be mainly oriented along the mean flow direction (l). In contrast, the edges in the vertical (n) and span-wise (m) direction mostly have low feature importances.

All in all, the observed patterns give interesting indications about which input variables have the most predictive value for the considered ANN, and could possibly be used to adjust the input stencil accordingly. Specifically, they suggest in our case that the input stencils can be made smaller in the vertical and span-wise direction without sacrificing their predictive value, and that an extension along the stream-wise direction may help to increase the predictive value of the input stencils.



a) n=0

b) n=1

c) n=2

d) n=3

e) n=4

Figure 11. Permutation feature importance of all u-velocities in the local input stencil (with indices (l,m,n)) for predicting τ_{wu}^{in} in the log-layer ($0.09375\frac{z}{\delta}$ $(55.3125z^{+})$) with $n_{hidden} = 64$, averaged over 3 flow fields reserved for a priori testing and 10 random shufflings. The five panels a-e each show one of the five horizontal planes (indicated by their vertical index n) present in the input stencils. τ_{wu}^{in} is located in the center of the shown horizontal plane, halfway between $n = 1$ and $n = 2$.

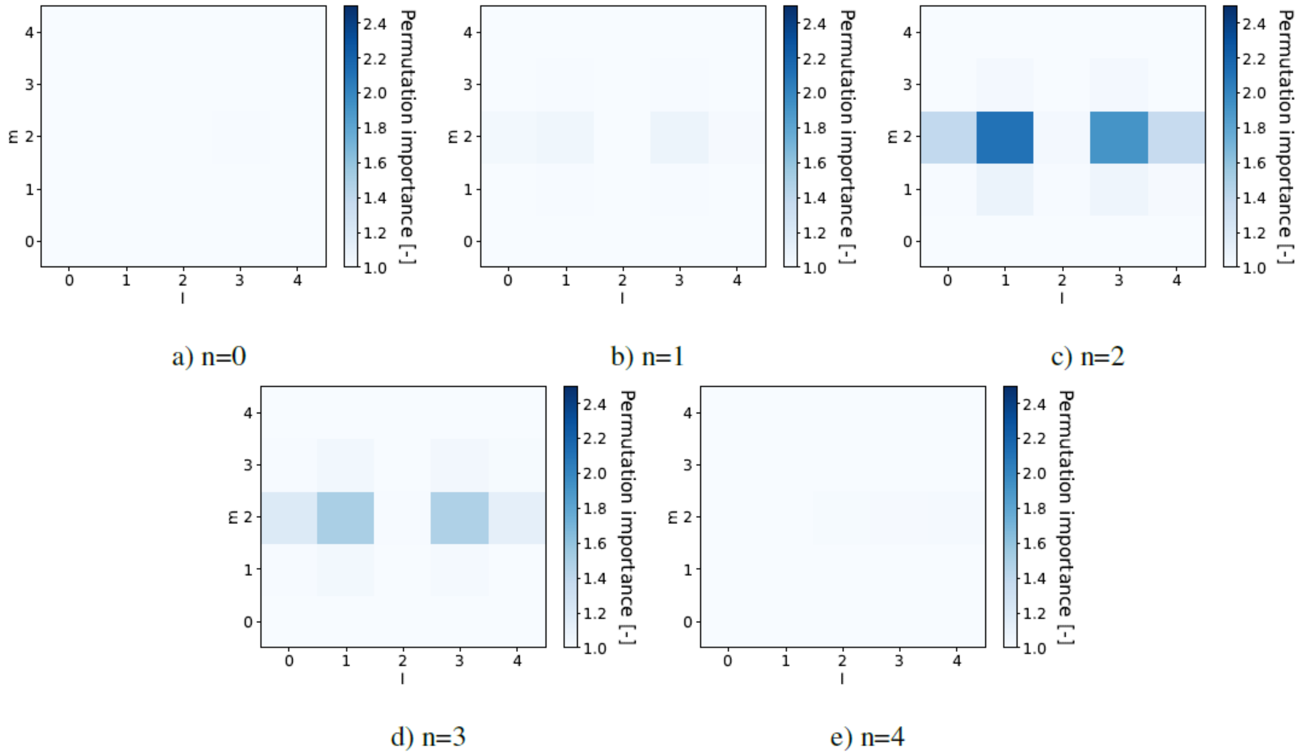


Figure 12. Permutation feature importance of all u-velocities in the local input stencil (with indices (l, m, n)) for predicting τ_{wu}^{out} in the log-layer ($0.09375\frac{z}{\delta}$ ($55.3125z^{+}$)) with $n_{hidden} = 64$, averaged over 3 flow fields reserved for a priori testing and 10 random shufflings. The five panels a-e each show one of the five horizontal planes (indicated by their vertical index n) present in the input stencils. τ_{wu}^{out} is located in the center of the shown horizontal plane, halfway between $n = 2$ and $n = 3$.

8 ANN a posteriori test

To test the a posteriori performance of our ANN LES SGS model(with $n_{hidden} = 64$) in an actual LES simulation, we directly incorporated it into our CFD code MicroHH (v2.0) (van Heerwaarden et al., 2017). We designed our ANN LES SGS model such that the integration into our CFD code was relatively straightforward (Sect. 2). Furthermore, we improved the computational performance of the ANN SGS model by implementing BLAS routines from the Intel(R) Math Kernel Library (version: 2019 update 5 for Linux), which has been optimized for the Intel CPUs we used (i.e. E5-2695 v2 (Ivy Bridge) and E5-2690 v3 (Haswell)). Still, the computational effort involved in the ANN SGS model was large: an equivalent LES simulation with the Smagorinsky SGS model was for our set-up about a factor 15 faster, showing that more optimizations are needed for practical applications.

With the ANN SGS model incorporated in our CFD code, we ran a LES with, similar as in the training and a priori test, an equidistant grid of 96 × 48 × 64 cells for the turbulent channel flow test case described in Sect. 3. Here, we used second-order

https://doi.org/10.5194/gmd-2020-289
Preprint. Discussion started: 10 November 2020

linear interpolation to calculate all the velocity tendencies, which is consistent with our training data generation procedure (Sect. 2 and 4). Furthermore, we initialized the LES simulation from one of the 3 flow fields reserved for the a priori testing. We did this to ensure that any possible errors in the initialization phase of the LES (i.e. before steady state is achieved) did not impact the solution. Still, our LES ran freely from the prescribed initialized steady-state fields, meaning that all the model and discretization errors made in calculating the channel flow steady state dynamics were included.

8.1 Performance

Our ANN LES SGS model produced numerically unstable a posteriori results without resorting to artificially introducing additional variance (for instance via eddy-viscosity models) or imposing strong ad-hoc numerical constraints. This was also reported by Beck et al. (2019), who showed that their SGS model only became numerically stable when combined with eddy-viscosity models. Several earlier studies that attempted to replace LES SGS or wall models with ANNs (Maulik et al., 2019; Wang et al., 2018; Yang et al., 2019; Zhou et al., 2019; Xie et al., 2019), did show stable a posteriori results, but in some cases only after imposing strong numerical constraints that dampened backscatter (Maulik et al., 2019; Zhou et al., 2019).

Crucially, for our set-up the instantaneous discretization errors associated with the finite-volume staggered LES grid were large, removing a large part of the variance present at high wave modes in the DNS (Fig. 4). Since we designed our ANN SGS model to fully compensate for these instantaneous discretization errors, our SGS model tended to introduce a large amount of variance at the highest resolved wave modes. This is probably less prominent for the ANN SGS models developed in the aforementioned studies (with the exception of Beck et al. (2019)) and for traditional SGS models like Smagorinsky, which do not fully compensate for all the instantaneous discretization errors made with staggered finite-volumes. In the a priori test, we consistently found that the Smagorinsky SGS model, as opposed to our ANN SGS model, strongly underestimated the small-scale variability of $\tau_{ij,DNS}$ (Sect. 7.1 and 7.2).

The introduction of additional variance at the highest wave modes by our ANN SGS model, is on its own not necessarily a problem if the energy transfer from the resolved to the unresolved scales is sufficient. In other words, our ANN SGS model needs to provide sufficient additional dissipation. However, we found that our ANN SGS model does not satisfy this requirement: for all three wind velocity components we observed a gradual pile-up of spectral energy at the smallest wave modes of the LES, which eventually resulted in numeric instability (shown as an example for the u-component in Fig. 13).

https://doi.org/10.5194/gmd-2020-289
Preprint. Discussion started: 10 November 2020

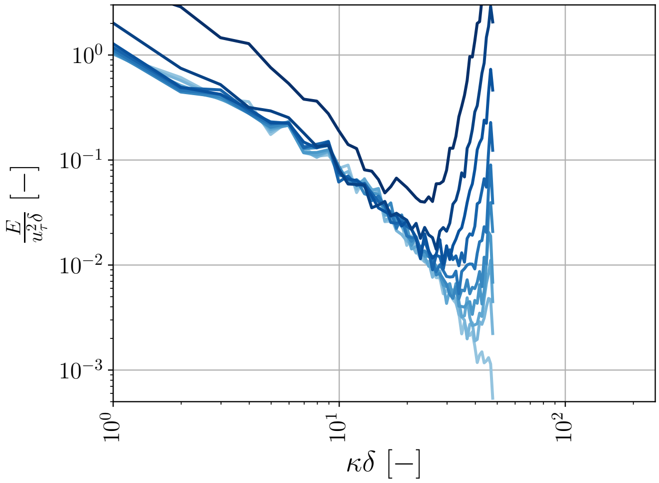


Figure 13. Time evolution of stream-wise spectra averaged in the span-wise direction, where the colour brightness indicates the different time steps. Here, the light blue colours refer to the first time steps, and the dark blue colours to the final time steps. The time steps range from t=0s to t=27s, with intervals of 3s.

We hypothesize that two issues prevented the ANN SGS model from producing the required dissipation: 1) error accumulation, and 2) aliasing errors.

In the first place, similar to Beck et al. (2019), we hypothesize that high-frequency errors in the ANN predictions accumulated over time due to strong positive feed-backs between our ANNs and the LES simulation. We stress that ANN SGS models like ours can never be perfect, and consequently will always introduce errors in an a posteriori simulation that affect, together with the full LES dynamics, its own inputs in the next time step. It strongly depends on the characteristics of the SGS model, whether this can result in positive feedback loops that cause divergence from the physical solution and subsequent numeric instability. In this regard, eddy-viscosity models like Smagorinsky have an important stabilizing property in steady-state channel flow: as soon as the energy content starts deviating from the physical solution, the subgrid dissipation is automatically adjusted (via a change in the gradients serving as input) to compensate for it.

Such a stabilizing property, however, was clearly lacking in our ANN SGS model. This is not surprising: we designed our ANN SGS model to compensate for many instantaneous discretization effects, which typically dampen the error accumulation at high-frequencies. It is well-known that, due to the chaotic nature of turbulence, small errors introduced by the predicted transports have a tendency to grow over time (e.g. Liu et al., 1994). On top of that, it has been shown before by Nadiga and Livescu (2007) that 'perfect' SGS models (that exactly compensate for the unresolved physics, modelling errors, and instantaneous discretization errrors), are inherently unstable in implicit-filtering LES due to the presence of multiple different attractors. These issues were likely exacerbated by the growing need for the ANN to extrapolate beyond its training state once the simulation started deviating from the physical solution. This extrapolation likely increased the ANN prediction errors, which would in turn accelerate the divergence from the physical solution.

In the second place, we hypothesize that, during the a posteriori test, aliasing errors became prominent due to the introduced variance at high wave modes. Such aliasing errors are known to introduce instability when not dampened by discretization errors and/or dealiasing techniques (e.g. Kravchenko and Moin, 1997; Chow and Moin, 2003). The quadratic velocity products in the non-linear advection term, can in principle introduce wave modes that are not supported by the finite LES grid. The additional variance could have prevented them from being dampened by the discretization errors, causing them to appear as spurious resolved wave modes in the finite LES solution. This would increase the amount of dissipation required in the LES simulation. These aliasing errors were not accounted for during the ANN training, as it only relied on *instantaneous* coarse-grained flow fields that did not contain additional variance.

9 Conclusions and recommendations

In this study, we evaluated and developed a data-driven large-eddy simulation (LES) subgrid-scale (SGS) model based on artificial neural networks (ANNs) that aims to represent the unresolved physics and instantaneous discretization errors in wall-bounded turbulent flows, without requiring simplifying assumptions, a separate wall model, and/or ad-hoc empirical corrections to achieve numerically stable results in wall-bounded flows. We focused specifically on the widely-used LES approach where a staggered finite-volume grid acts as an implicit filter, where the discretization errors can strongly interact with the resolved physics.

We designed our ANN SGS model such that, similar to conventional eddy-viscosity SGS models like Smagorinsky, it could be applied locally in the grid domain: the employed ANNs used as input only local $5 \times 5 \times 5$ stencils of the resolved wind velocity components ($\overline{u}$,$\overline{v}$,$\overline{w}$). Interestingly, an additional analysis we performed with so-called permutation feature importances, suggested that our ANNs mostly focused on a small part of the stencils oriented along the mean flow direction. Hence, the input stencils we used could perhaps be further optimized by selecting smaller stencils that extend along the mean flow direction.

Using as a test case turbulent channel flow (with $Re_\tau = 590$), we trained the ANNs with individual 3D flow fields obtained from direct numerical simulation (DNS). By applying an explicit finite-volume filter (i.e. a discrete 3D top-hat filter) on the high-resolution DNS fields and mimicking the instantaneous discretization errors made in actual LES, we generated millions of ANN input-output pairs that allowed us to train the ANNs in a supervised manner.

Subsequently, we performed both an *a priori* and *a posteriori* test. As an a priori test, we directly compared the ANN predictions to the DNS derived values for flow fields unseen during training. We found, in general, excellent agreement for all heights in the channel: the spatial patterns in the DNS values were well captured, and the correlation coefficients between the ANN predictions and DNS values were high (mostly between 0.6 and 1.0). The ANNs were thus well able to represent the unresolved physics and instantaneous discretization errors in the entire flow, based only on the resolved flow fields. This shows their potential to serve as highly accurate SGS models both in the near-wall region and bulk flow. We do note that the ANNs we employed still had a few shortcomings that can possibly be improved upon: the extreme SGS fluxes were slightly underestimated, and the predicted horizontally averaged vertical profile of τ_{wu} deviated close to the walls.

To test our ANN SGS model a posteriori, we incorporated our trained ANN SGS model directly into a LES of the selected turbulent channel flow test case. Contrary to the a priori test, the ANN SGS model did not produce satisfactory results. Since our ANN SGS model, in contrast to traditional SGS models like Smagorinsky, compensated for many instantaneous discretization effects by introducing additional variance, the need for additional dissipation increased. The ANN SGS model appeared not to provide this dissipation sufficiently, causing an artificial build-up of TKE at high wave-modes that eventually destabilized the solution. We hypothesized that our ANN SGS model did not produce sufficient dissipation because of 1) error accumulation, and 2) aliasing errors.

We therefore conclude that our ANN SGS model cannot, in its current form, achieve computationally stable results. One way forward could be to adjust the training procedure such that it reflects better the a posteriori simulation. A potential elegant way to achieve this may be the *online* learning procedure proposed by Rasp (2020), where our ANN SGS model would be directly trained in an online a posteriori LES simulation. This would require a simultaneously running DNS that is continuously kept in sync with the LES by constant forcing, such that the ANN can learn directly from the DNS how its predictions should be adjusted for the full LES dynamics and its own errors.

A possibly less involved solution may be to manually mimic the online training in the offline setting used in this study, avoiding the need to dynamically couple a DNS and LES. This may be done by manually adjusting the fully resolved DNS flow fields at each time step during an offline DNS simulation similar to ours, in such a way that that the offline DNS simulation more closely resembles an DNS simulation incorporated in an online training procedure. Alternatively, errors expected to be introduced by an ANN SGS model in an a posteriori LES simulation, could be added to the filtered flow fields $\overline{u}$, $\overline{v}$, $\overline{w}$ during training. This may help to reduce the sensitivity of the ANN to its own errors, alleviating the need for extrapolation once the a posteriori LES simulation starts diverging.

Another way forward may be to include more physical constraints and insights in the ANN design and training. It could for instance be interesting to include energy conservation terms in the loss function used during offline training, as it may allow the ANNs to better conserve energy after training.

Besides that, another main approach to achieve computationally stable results would be to relax some of the requirements we specified, allowing the usage of more commonly applied methods in fluid mechanics to increase dissipation. One option could be to impose strong numerical constraints that dampen backscatter, as done for instance by Maulik et al. (2019) and Zhou et al. (2019). Another option could be to implement an additional eddy-viscosity model that can provide the required additional dissipation, in line with existing so-called mixed models (e.g. Sagaut, 2006). This was successfully done by Beck et al. (2019), who achieved accurate and stable a posteriori results with their LES ANN SGS model only when they combined it with eddy-viscosity models.

Furthermore, another option could be to perform LES with *explicit* filtering, which would strongly dampen the discretization errors and provide additional subgrid dissipation (e.g. Chow and Moin, 2003; Sagaut, 2006). This does have the disadvantage that the total computational effort increases. An additional (potentially expensive) filtering operation would have to be applied, and, to retain the same number of degrees of freedom in the resolved flow fields as without explicit filtering, the grid resolution would have to be increased (e.g. Chow and Moin, 2003; Sagaut, 2006).

All in all, our developed ANN LES SGS model has, based on its excellent a priori performance, potential to improve the representation of the unresolved physics and discretization errors in turbulent flows. However, the developed ANN LES SGS model is in its current form still prone to numeric instability in a posteriori simulations. Hence, several open challenges remain before the potential of ANN LES SGS models like ours can be successfully leveraged in practical applications.

Code availability. All the code used to generate the data and figures in this paper are hosted at a GitHub repository, located at https://github.com/robinstoffer/microhh2/tree/NN. To facilitate a posteriori LES simulations with the ANN SGS model developed in this paper, the parameters of the ANN selected in this paper are included as txt-files. The authors will assign a DOI to the repository once the paper has been accepted for publication.

Author contributions. RS generated the training data, wrote the ANN SGS model, and tested its performance in an a priori and a posteriori test. CvH supported RS in developing and writing the ANN SGS model. RS and CvH developed the ideas that led to the study. CvH, CvL, DP, VC, and MV all provided extensive feedback on the training data generation, ANN design, and hard-ware specific tuning. MJ helped in interpreting the a posteriori results and coming up with suggestions for future research. RS wrote the manuscript. CvH, OH, CvL, MJ, and MV provided valuable feedback on the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The authors gratefully acknowledge the SURF Open Innovation Lab and WIMEK graduate school (WUR) for funding this study.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: TensorFlow: A system for large-scale machine learning, in: 12th (USENIX) Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283, 2016.

Bardina, J., Ferziger, J., and Reynolds, W.: Improved subgrid-scale models for large-eddy simulation, in: 13th fluid and plasmadynamics conference, p. 1357, https://doi.org/10.2514/6.1980-1357, 1980.

Beck, A., Flad, D., and Munz, C.: Deep neural networks for data-driven LES closure models, Journal of Computational Physics, 398, 108 910, https://doi.org/10.1016/j.jcp.2019.108910, 2019.

Bou-Zeid, E., Meneveau, C., and Parlange, M.: A scale-dependent Lagrangian dynamic model for large eddy simulation of complex turbulent flows, Physics of fluids, 17, 025 105, https://doi.org/10.1063/1.1839152, 2005.

Breiman, L.: Random forests, Machine learning, 45, 5–32, https://doi.org/10.1023/A:1010933404324, 2001.

Brunton, S. L., Noack, B. R., and Koumoutsakos, P.: Machine learning for fluid mechanics, Annual Review of Fluid Mechanics, 52, 477–508, https://doi.org/10.1146/annurev-fluid-010719-060214, 2020.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., and Zdeborová, L.: Machine learning and the physical sciences, Reviews of Modern Physics, 91, 045 002, https://doi.org/10.1103/RevModPhys.91.045002, 2019.

Cheng, Y., Giometto, M., Kauffmann, P., Lin, L., Cao, C., Zupnick, C., Li, H., Li, Q., Abernathey, R., and Gentine, P.: Deep learning for subgrid-scale turbulence modeling in large-eddy simulations of the atmospheric boundary layer, arXiv preprint arXiv:1910.12125, 2019.

Chow, F. K. and Moin, P.: A further study of numerical errors in large-eddy simulations, Journal of Computational Physics, 184, 366–380, https://doi.org/10.1016/S0021-9991(02)00020-7, 2003.

Clark, R. A., Ferziger, J. H., and Reynolds, W. C.: Evaluation of subgrid-scale models using an accurately simulated turbulent flow, Journal of fluid mechanics, 91, 1–16, https://doi.org/10.1017/S002211207900001X, 1979.

Denaro, F. M.: What does Finite Volume-based implicit filtering really resolve in Large-Eddy Simulations?, Journal of Computational Physics, 230, 3849–3883, https://doi.org/10.1016/j.jcp.2011.02.011, 2011.

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

Fisher, A., Rudin, C., and Dominici, F.: All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously, Journal of Machine Learning Research, 20, 1–81, http://jmlr.org/papers/v20/18-760.html, 2019.

Gamahara, M. and Hattori, Y.: Searching for turbulence models by artificial neural network, Physical Review Fluids, 2, 054 604, https://doi.org/10.1103/PhysRevFluids.2.054604, 2017.

Ghosal, S.: An analysis of numerical errors in large-eddy simulations of turbulence, Journal of Computational Physics, 125, 187–206, https://doi.org/10.1006/jcph.1996.0088, 1996.

Giacomini, B. and Giometto, M. G.: On the suitability of general-purpose finite-volume-based solvers for the simulation of atmospheric-boundary-layer flow, Geoscientific Model Development Discussions, pp. 1–28, https://doi.org/10.5194/gmd-2020-84, 2020.

He, K., Zhang, X., Ren, S., and Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, in: The IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034, https://doi.org/10.1109/ICCV.2015.123, 2015.

Hornik, K., Stinchcombe, M., and White, H.: Multilayer feedforward networks are universal approximators., Neural networks, 2, 359–366, https://doi.org/10.1016/0893-6080(89)90020-8, 1989.

Jimenez, J. and Moser, R. D.: Large-eddy simulations: where are we and what can we expect?, AIAA journal, 38, 605–612, https://doi.org/10.2514/2.1031, 2000.

Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

Kravchenko, A. G. and Moin, P.: On the effect of numerical errors in large eddy simulations of turbulent flows, Journal of computational physics, 131, 310–322, https://doi.org/10.1006/jcph.1996.5597, 1997.

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

Langford, J. A. and Moser, R. D.: Optimal LES formulations for isotropic turbulence, Journal of fluid mechanics, 398, 321–346, https://doi.org/10.1017/S0022112099006369, 1999.

Langford, J. A. and Moser, R. D.: Breakdown of continuity in large-eddy simulation, Physics of Fluids, 13, 1524–1527, https://doi.org/10.1063/1.1358876, 2001.

Lilly, D. K.: The representation of small-scale turbulence in numerical simulation experiments, in: Proceedings of the IBM Scientific Computing Symposium on Environmental Sciences, pp. 195–210, https://doi.org/10.5065/D62R3PMM3, 1967.

Ling, J., Jones, R., and Templeton, J.: Machine learning strategies for systems with invariance properties, Journal of Computational Physics, 318, 22–35, https://doi.org/10.1016/j.jcp.2016.05.003, 2016a.

Ling, J., Kurzawski, A., and Templeton, J.: Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, Journal of Fluid Mechanics, 807, 155–166, https://doi.org/10.1017/jfm.2016.615, 2016b.

Liu, S., Meneveau, C., and Katz, J.: On the properties of similarity subgrid-scale models as deduced from measurements in a turbulent jet, Journal of Fluid Mechanics, 275, 83–119, https://doi.org/10.1017/S0022112094002296, 1994.

Maas, A. L., Hannun, A. Y., and Ng, A. Y.: Rectifier nonlinearities improve neural network acoustic models, in: ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013.

Maulik, R., San, O., Rasheed, A., and Vedula, P.: Subgrid modelling for two-dimensional turbulence using neural networks, Journal of Fluid Mechanics, 858, 122–144, https://doi.org/10.1017/jfm.2018.770, 2019.

McMillan, O. J. and Ferziger, J. H.: Direct testing of subgrid-scale models, AIAA Journal, 17, 1340–1346, https://doi.org/10.2514/3.61313, 1979.

Milano, M. and Koumoutsakos, P.: Neural network modeling for near wall turbulent flow, Journal of Computational Physics, 182, 1–26, https://doi.org/10.1006/jcph.2002.7146, 2002.

Molnar, C.: Interpretable Machine Learning, lulu.com, https://christophm.github.io/interpretable-ml-book/, 2019.

Moser, R. D., Kim, J., and Mansour, N. N.: Direct numerical simulation of turbulent channel flow up to Re τ= 590, Physics of fluids, 11, 943–945, https://doi.org/10.1063/1.869966, 1999.

Nadiga, B. T. and Livescu, D.: Instability of the perfect subgrid model in implicit-filtering large eddy simulation of geostrophic turbulence, Physical Review E, 75, 046 303, https://doi.org/10.1103/PhysRevE.75.046303, 2007.

Pope, S. B.: Turbulent flows, Cambridge University Press, Cambridge, United Kingdom, 2001.

Rasp, S.: 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, 2185–2185, https://doi.org/10.5194/gmd-13-2185-2020, 2020.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J.: Learning representations by back-propagating errors, nature, 323, 533–536, https://doi.org/10.1038/323533a0, 1986.

Sagaut, P.: Large eddy simulation for incompressible flows: an introduction, Springer Science & Business Media, Berlin, Germany, 2006.

Sarghini, F., De Felice, G., and Santini, S.: Neural networks based subgrid scale modeling in large eddy simulations, Computers & fluids, 32, 97–108, https://doi.org/10.1016/S0045-7930(01)00098-6, 2003.

Schmitt, F. G.: About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity, Comptes Rendus Mécanique, 335, 617–627, https://doi.org/10.1016/j.crme.2007.08.004, 2007.

Singh, A. P., Duraisamy, K., and Zhang, Z. J.: Augmentation of turbulence models using field inversion and machine learning, in: 55th AIAA Aerospace Sciences Meeting, p. 0993, https://doi.org/10.2514/6.2017-0993, 2017.

Smagorinsky, J.: General circulation experiments with the primitive equations: I. The basic experiment, Monthly weather review, 91, 99–164, https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2, 1963.

Van Driest, E. R.: On turbulent flow near a wall, Journal of the aeronautical sciences, 23, 1007–1011, https://doi.org/10.2514/8.3713, 1956.

van Heerwaarden, C. C., van Stratum, B. J. H., Heus, T., Gibbs, J. A., Fedorovich, E., and Mellado, J. P.: MicroHH 1.0: a computational fluid dynamics code for direct numerical simulation and large-eddy simulation of atmospheric boundary layer flows, Geoscientific Model Development, 10, 3145–3165, https://doi.org/10.5194/gmd-10-3145-2017, 2017.

Völker, S., Moser, R. D., and Venugopal, P.: Optimal large eddy simulation of turbulent channel flow based on direct numerical simulation statistical data, Physics of Fluids, 14, 3675–3691, https://doi.org/10.1063/1.1503803, 2002.

Vollant, A., Balarac, G., and Corre, C.: Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures, Journal of Turbulence, 18, 854–878, https://doi.org/10.1080/14685248.2017.1334907, 2017.

Wang, J., Wu, J., and Xiao, H.: Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, Physical Review Fluids, 2, 034 603, https://doi.org/10.1103/PhysRevFluids.2.034603, 2017.

Wang, Z., Luo, K., Li, D., Tan, J., and Fan, J.: Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation, Physics of Fluids, 30, 125 101, https://doi.org/10.1063/1.5054835, 2018.

Wu, J., Xiao, H., and Paterson, E.: Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework, Physical Review Fluids, 3, 074 602, https://doi.org/10.1103/PhysRevFluids.3.074602, 2018.

Xie, C., Wang, J., Li, K., and Ma, C.: Artificial neural network approach to large-eddy simulation of compressible isotropic turbulence, Physical Review E, 99, 053 113, https://doi.org/10.1103/PhysRevE.99.053113, 2019.

Yang, X., Zafar, S., Wang, J., and Xiao, H.: Predictive large-eddy-simulation wall modeling via physics-informed neural networks, Physical Review Fluids, 4, 034 602, https://doi.org/10.1103/PhysRevFluids.4.034602, 2019.

Zandonade, P. S., Langford, J. A., and Moser, R. D.: Finite-volume optimal large-eddy simulation of isotropic turbulence, Physics of fluids, 16, 2255–2271, https://doi.org/10.1063/1.1736672, 2004.

Zhou, Z., He, G., Wang, S., and Jin, G.: Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network, Computers & Fluids, 195, 104 319, https://doi.org/10.1016/j.compfluid.2019.104319, 2019.