https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

LEX v1.4: A New Large-Eddy Simulation Model in JAX with GPU Acceleration and Automatic Differentiation

Xingyu Zhu[1], Yongquan Qu[2,3], and Xiaoming Shi[1,4]

[1]Division of Environment and Sustainability, Hong Kong University of Science and Technology, Hong Kong, China
[2]NSF Center for Learning the Earth with Artificial Intelligence and Physics (LEAP), Columbia University, New York, NY, USA.
[3]Department of Earth and Environmental Engineering, Columbia University, New York, NY, USA.
[4]Center for Ocean Research in Hong Kong and Macau, Hong Kong University of Science and Technology, Hong Kong, China

Correspondence: Xiaoming Shi (shixm@ust.hk)

Abstract. Large-eddy simulations (LESs) are essential tools for studies on atmospheric turbulence and clouds and play critical roles in the development of turbulence and convection parameterizations. Current numerical weather models have approached kilometer-scale resolution as supercomputing facilities advance. However, this resolution range is in the so-called gray zone, where subgrid-scale (SGS) turbulence actively interacts with resolved motion and significantly influences the large-scale characteristics of simulated weather systems. Thus, developing SGS turbulence models for the gray zone requires new LES models, which must run sufficiently fast when simulating large domains and enable new approaches to develop SGS models. Here we used the Python library JAX to develop a new LES model. It is based on the generalized pseudo-incompressible equations formulated by Durran (2008). The new LES model is capable of adequate parallelism and can run at a fast speed with GPU acceleration. For a classic warm bubble case, the traditional Smagorinsky model fails to reproduce the correct structure evolution of the warm bubble, though it can modestly correct the rising speed in gray-zone resolution simulations. Utilizing the capability of JAX for automatic differentiation, we trained a deep learning-based SGS turbulence model for the same case. The trained deep learning SGS model, based on simple three-dimensional convolutional neural networks (CNNs), enables this physics-deep learning hybrid model to accurately simulate the expansion of the thermal bubble and the development of rotors surrounding the center of the bubble at a gray-zone resolution. The gray-zone simulation results are comparable to that of the benchmark LES resolution.

1 Introduction

Large-eddy simulation (LES) has been widely used in the atmospheric science community as a benchmark for the development of subgrid-scale (SGS) turbulence parameterizations in numerical weather prediction (NWP) and climate models (Teixeira and Cheinet, 2004; Sullivan and Patton, 2011; Verrelle et al., 2017; Wu et al., 2020; Jadhav and Chandy, 2021). LESs have also helped the community to achieve a better understanding of cloud feedback, which interacts with boundary layer turbulence and contributes to climate sensitivity (Bretherton, 2015; Blossey et al., 2016; Tan et al., 2017; Shen et al., 2022). The capability

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

of LESs to simulate small-scale turbulence motion and its interaction with other processes, such as clouds and radiation, with high fidelity makes them a unique and valuable tool in atmospheric science.

Although supercomputing platforms continue to advance, LES on a large domain for operational NWP is still not reachable. Current-generation regional NWP models and some regional climate simulations are often run at kilometer-scale resolution (Prein et al., 2017; Schär et al., 2020). Global simulations with kilometer-scale resolution have also been recently demonstrated for a four-month-long integration (Wedi et al., 2020).

The new challenge in kilometer-scale resolution resides in the gray zone for turbulence and convection. Gray zone (or terra incognita) is defined when the filter length scale has the same order as the dominant turbulence length scale (Wyngaard, 2004). In the gray zone, turbulence and conventions can only be partially resolved and thus SGS motions interact actively with resolved motions in all three spatial dimensions and are not in statistical equilibrium, which contrasts with the assumptions used in the conventional planetary boundary layer (PBL) turbulence and cumulus convection schemes, such as horizontal homogeneity and quasi-equilibrium. As a result, the conventional parameterization schemes cannot be directly applied with such grid spacings (Chow et al., 2019; Honnert et al., 2020). Meanwhile, LES-type turbulence schemes cannot be applied to the gray zone either because they often assume isotropic turbulence and downscale energy transfer. In contrast, gray-zone turbulence is anisotropic and allows energy backscatter (Shi et al., 2019).

Therefore, LESs are becoming an increasingly valuable tool for further advancement of SGS turbulence representation in gray-zone simulations and new LES codes which can run faster than before are needed to enable simulations covering large domains to capture the potential influence of SGS turbulence on the organization of convection and clouds (Shi and Fan, 2021).

GPU codes are known to run much faster than conventional Fortran or C codes on CPUs (Demeshko et al., 2013; Price et al., 2014; Schalkwijk et al., 2015; van Heerwaarden et al., 2017). Recent years have seen many research efforts focused on GPU model development. Donahue et al. (2024) rewrote a GPU architecture for the Simple Cloud-Resolving Energy Exascale Earth System Atmosphere Model (SCREAM) with C(++) and found an averaged $6\times$ acceleration compared to the CPU codes. Sridhar et al. (2022) developed ClimateMachine with Julia and provided an architecture-portable framework for heterogeneous CPU/GPU computing for atmospheric modeling. An ocean dynamical core that can be operated on GPUs was also implemented with Julia in the newly developed ocean model, Oceananigans, and made significant achievements in model efficiency (Silvestri et al., 2024, 2025). For LES, Sauer and Muñoz-Esparza (2020) developed FastEddy, a CUDA C(++) based model, and achieved a $6\times$ acceleration on one GPU over state-of-the-art LES using 64 CPUs.

Meanwhile, except for GPU acceleration, differentiability of LES codes is crucial for advancing next-generation deep learning (DL) based SGS parameterizations, but till now few GPU-based LES have mentioned differentiability. Differentiable LES exposes every step of the dynamical core as differentiable operations, enabling end-to-end gradient propagation through the simulation. This capability supports a more powerful training paradigm for DL parameterizations: the neural SGS module is optimized via differentiable roll-outs, adjusting its parameters based on how errors accumulate through the evolving flow dynamics. Recent years have seen a surge in the application of such coupled frameworks to physical parameterization problems (Kochkov et al., 2021; Qu and Shi, 2023; Watt-Meyer et al., 2024; Qu et al., 2024). Models trained in this way demonstrate superior forecast stability compared to both traditional schemes and offline-trained neural parameterizations. By integrating

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

the physics-based core directly into the training loop, these hybrid approaches yield more reliable, interpretable, and accurate weather and climate predictions than purely data-driven DL models. Commonly, hybrid models rely on high-fidelity numerical simulation data for training, but recent study also shows that they are available to include observational knowledge into the training process, further indicating the great potential of hybrid models to be applied for realistic simulations. For example, NeuralGCM (Kochkov et al., 2024) matches or outperforms state-of-the-art DL forecasting models across both short and long lead times, while also reducing computational cost relative to conventional general circulation models. It further improves the model performance and gives more accurate forecasts for precipitation by jointly using ERA-5 and satellite observational data (Yuval et al., 2024). These results underscore the promising application of differentiable LES for next-generation SGS parameterizations.

In this paper, a new fast and differentiable LES code that runs on GPUs is implemented with a newly developed Python library, JAX (Bradbury et al., 2018). Different from Fortran or C, numerical models written in Python codes are easier to be coupled with DL models for training. Existing studies have also shown that JAX-GPU codes enable faster accelerations and less computational costs when the problem sizes become quite large (Häfner et al., 2021; Zhou et al., 2024), making it a useful tool to realize LES on large domains. The new LES code is named LEX. LEX has the following distinct advantages: (1) it is numerically stable with its acoustic-wave-filtered governing equations and advanced integration schemes, (2) it computes quite fast by using XLA (accelerated linear algebra), a domain-specific compiler that accelerates code via many techniques and enables the compiled codes to run on TPUs or GPUs, (3) it is platform-agnostic, where the same code can be compiled and run on CPUs, GPUs, or TPUs, (4) it is auto-differentiable so that it enables DL-base parameterization to be trained with a coupled online training strategy (Rasp, 2020) in a physics-DL hybrid structure (von Rueden et al., 2020).

The structure of this paper is mainly organized as follows. Section 2 introduces the setup of the LEX model and the thermal testing case, and also the training flow of the hybrid ML-based SGS model. Validation results for the LEX model are shown in Section 3. Testing results for the ML model are presented in Section 4. In Section 5, the computational costs are compared to investigate to what extent the LEX model is faster than the conventional LES model and also the feasibility of the DL-based SGS model. Section 6 contains the summary and discussions.

2 Method and Experiment Design

2.1 LEX

2.1.1 Governing Equations

To develop LEX, the acoustic-wave-filtered equations for compressible stratified flow developed by Durran (2008) is adopted as the governing equations, where a pseudo-density ρ^* is defined to eliminate sound waves and enforce the mass conservation equation such at

$$\frac{1}{\rho^*}\frac{D\rho^*}{Dt} + \nabla \cdot \mathbf{u} = 0, \tag{1}$$

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

and the pseudo-density is defined as

$$\rho^* = \frac{\tilde{\rho}(x,y,z,t)\tilde{\theta}_\rho(x,y,z,t)}{\theta_\rho}, \tag{2}$$

where ˜ denotes a spatially varying reference state. The potential temperature is used in the definition of (Durran, 2008) for the dry air. Here the density potential temperature θ_ρ is used as a replacement to include the effect of water variables for a moist situation. It is defined and approximated as

$$\theta_\rho = \theta\left(\frac{1+q_v/\varepsilon}{1+q_v+q_l+q_i}\right) \approx \theta\left[1+\left(\frac{1}{\epsilon}-1\right)q_v - q_l - q_i\right], \tag{3}$$

where q_v, q_l, q_i are mixing ratios of water vapor, liquid water, and cloud ice. In the reference state, q_l and q_i can be assumed to be zero, thus $\tilde{\theta}_\rho$ is the reference state virtual potential temperature. With this definition, and further with some approximation, the mass (pseudo-density) conservation equation becomes

$$\frac{\partial\tilde{\rho}\tilde{\theta}_\rho}{\partial t} + \nabla\cdot(\tilde{\rho}\tilde{\theta}_\rho\mathbf{u}) = \frac{\tilde{\rho}H_m}{c_p\tilde{\pi}}, \tag{4}$$

where H_m is the heating rate per unit mass, $\mathbf{u}$ is velocity, and π is the Exner function. Perturbations with respect to the reference state is defined such that $\theta' = \theta - \tilde{\theta}$ and $\pi' = \pi - \tilde{\pi}$. Durran (2008) further separated $\tilde{\pi}$ into a large horizontally uniform component $\tilde{\pi}_v(z,t)$ and a remainder $\tilde{\pi}_h(x,y,z,t)$ for computational accuracy and notational convenience. Then the momentum and thermodynamics equations are the following,

$$\frac{D\mathbf{u}_h}{Dt} + f\mathbf{k}\times\mathbf{u}_h + c_p\theta_\rho\nabla_h(\tilde{\pi}_h+\pi') = 0 \tag{5}$$

$$\frac{Dw}{Dt} + c_p\theta_\rho\frac{\partial\pi'}{\partial z} = B \tag{6}$$

$$\frac{D\theta}{Dt} = \frac{H_m}{c_p\tilde{\pi}}, \tag{7}$$

where $\mathbf{u}_h$ is the horizontal velocity vector, ∇_h is the horizontal gradient operator, and f is the Coriolis parameter. B is the linearized buoyancy,

$$B = g\left[\frac{\theta'}{\tilde{\theta}} + \left(\frac{1}{\epsilon}-1\right)(q_v-\tilde{q}_v) - q_l - q_i\right], \tag{8}$$

in which, $\tilde{q}_v$ is the reference state mixing ratio of water vapor. The reference state satisfies the equation of state and the hydrostatic balance equation:

$$\tilde{\pi} = \left(\frac{R}{p_s}\tilde{\rho}\tilde{\theta}_\rho\right)^{R/c_v} \tag{9}$$

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

$$c_p\tilde{\theta}_\rho\frac{\partial\tilde{\pi}}{\partial z}=-g. \tag{10}$$

The last unknown variable needed for integration is the pressure perturbation π', which needs to be solved diagnostically to enforce Equation (4). The diagnostic relationship is obtained by multiplying the momentum equation by $\tilde{\rho}\tilde{\theta}_\rho$, taking the divergence of the result and subtracting $\partial/\partial t$ of Equation (4). The resulting diagnostic equation is provided by Durran (2008) as his Equation (5.2):

$$\begin{aligned}c_p\nabla\cdot(\tilde{\rho}\tilde{\theta}_\rho\theta_\rho\nabla\pi')=&-\nabla\cdot(\tilde{\rho}\tilde{\theta}_\rho\mathbf{u}\cdot\nabla)\mathbf{u}-f\nabla_h\cdot(\mathbf{k}\times\tilde{\rho}\tilde{\theta}_\rho\mathbf{u}_h)+\frac{\partial\,\tilde{\rho}\tilde{\theta}_\rho B}{\partial z}\\&-c_p\nabla_h\cdot(\tilde{\rho}\tilde{\theta}_\rho\theta_\rho\nabla_h\tilde{\pi}_h)-\frac{\partial}{\partial t}\left(\frac{\tilde{\rho}H_m}{c_p\tilde{\pi}}\right)+\nabla\cdot\left(\frac{\partial\,\tilde{\rho}\tilde{\theta}_\rho}{\partial t}\mathbf{u}\right)+\frac{\partial^2\tilde{\rho}\tilde{\theta}_\rho}{\partial t^2}.\end{aligned} \tag{11}$$

Assuming the tendency in the reference state is small, the last few terms involving time derivative can be ignored in the equation above, then the diagnostic relation for π' is

$$\begin{aligned}c_p\nabla\cdot(\tilde{\rho}\tilde{\theta}_\rho\theta_\rho\nabla\pi')=&-\nabla\cdot(\tilde{\rho}\tilde{\theta}_\rho\mathbf{u}\cdot\nabla)\mathbf{u}-f\nabla_h\cdot(\mathbf{k}\times\tilde{\rho}\tilde{\theta}_\rho\mathbf{u}_h)+\frac{\partial\,\tilde{\rho}\tilde{\theta}_\rho B}{\partial z}\\&-c_p\nabla_h\cdot(\tilde{\rho}\tilde{\theta}_\rho\theta_\rho\nabla_h\tilde{\pi}_h)=\mathcal{R}.\end{aligned} \tag{12}$$

The model has no microphysics scheme yet, so water vapor is included just like a tracer, though it affects buoyancy.

2.1.2 Numerical Techniques

For time integration, the four-stage third-order Strong-stability-preserving Runge-Kutta (SSPRK3) scheme (Durran, 2010) is used to ensure better numerical stability. The weighted essentially non-oscillatory (WENO) schemes (Shu, 1998) are employed to solve the advection tendencies for the momentum equations, with a fifth-order scheme for the horizontal direction and a third-order scheme for the vertical direction. Three layers of ghost points are used in each side of x and y directions to employ the fifth-order WENO scheme for the horizontal fluxes. The discretization adopts the staggered Arakawa C-grid. The pressure equation (12) is solved with the biconjugate gradient stabilized method (BiCGSTAB).

2.1.3 Testing Simulation Configurations

The three-dimensional numerical simulation of a rising thermal (Wicker and Skamarock, 1998; Bryan and Fritsch, 2002) is applied to validate the accuracy of LEX. The employed grid spacing is 100 m in both x, y, and z directions. The entire domain is 24 km by 24 km horizontally and 12 km vertically. The initial reference state has a constant potential temperature of 300 K, and features motionless air, hydrostatic equilibrium, and lapse rates corresponding to neutral stability. Water vapor is included for moist cases with a constant relative humidity of 10% everywhere in the initial condition. The thermal is set at the central part of the domain at the bottom, with the initial potential temperature perturbation being as:

$$\theta'=\begin{cases}\theta_c\cos^2(\frac{\pi L_b}{2}) & \text{if } L_b\le 1,\\ 0 & \text{otherwise,}\end{cases}$$

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

where θ_c is used to adjust the maximum value of the potential temperature perturbation at the centre of the thermal to simulate different thermal rising speeds, and L_b is the radial normalized distance between any point in the domain and the centre of the thermal, which is defined as:

$$L_b = \sqrt{(\frac{x - x_c}{x_r})^2 + (\frac{y - y_c}{y_r})^2 + (\frac{z - z_c}{z_r})^2}, \tag{13}$$

where $\boldsymbol{X}_c$ is the coordinates of the thermal centre, with $x_c = y_c = 12\,\text{km}$, $z_c = 2\,\text{km}$, and $\boldsymbol{X}_r$ is the initial radius of the thermal, with $x_r = y_r = z_r = 2\,\text{km}$.

The initial potential temperature perturbation will induce an upward buoyancy force and initiate the vertical acceleration of the bubble in the very beginning. The buoyancy will then cause the thermal to rise and evolve. During rising, the upper part of the thermal will elongate. Two rotors will also be developed on each side of the bubble in this process. The structure of the thermal maintains strictly symmetric as it evolves.

2.2 Deep Learning SGS Model

2.2.1 SGS Correction

Written with JAX, DL models can be coupled with the LES model for training. This paper tested the hybrid model's capability to use a DL-based SGS parameterization.

The SGS process refers to the unresolved part of the numerical simulations due to the relatively coarse grid size. Taking the potential temperature equation (7) for example, it can be written as the following on a numerical grid:

$$\frac{D\bar{\theta}}{Dt} = \frac{H_m}{c_p\tilde{\pi}} + \tau, \tag{14}$$

where $\bar{\theta}$ is the LES grid filtered potential temperature and τ is the SGS tendency.

To improve the stability of the numerical integration, in this paper, the DL model is used to represent an SGS correction term instead of the tendency term (Um et al., 2021; Kochkov et al., 2021; Qu and Shi, 2023), which is defined as:

$$T_\theta = \int_{t_0}^{t_0+\Delta t} \tau dt, \tag{15}$$

where T is the SGS correction term and the integration is for one time step of the dynamical core. T can be obtained from the DL model, which is:

$$T_\theta = \mathcal{M}(\hat{\bar{\theta}}'_{t_0+\Delta t}, \hat{\bar{q}}'_{v,t_0+\Delta t}, \hat{\bar{u}}_{t_0+\Delta t}, \hat{\bar{v}}_{t_0+\Delta t}, \hat{\bar{w}}_{t_0+\Delta t}, \hat{\bar{\pi}}'_{t_0+\Delta t}). \tag{16}$$

$\mathcal{M}$ is the DL model, $t_0 + \Delta t$ denotes that those are the variables after one time step integration of the dynamical core, and $\hat{}$ denote that those have not been corrected by the DL model. Thus for each DL model correcting step, the forecast status of the potential temperature (θ) is updated as:

$$\bar{\theta}_{t_0+\Delta t} = \hat{\bar{\theta}}_{t_0+\Delta t} + T_\theta, \tag{17}$$

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

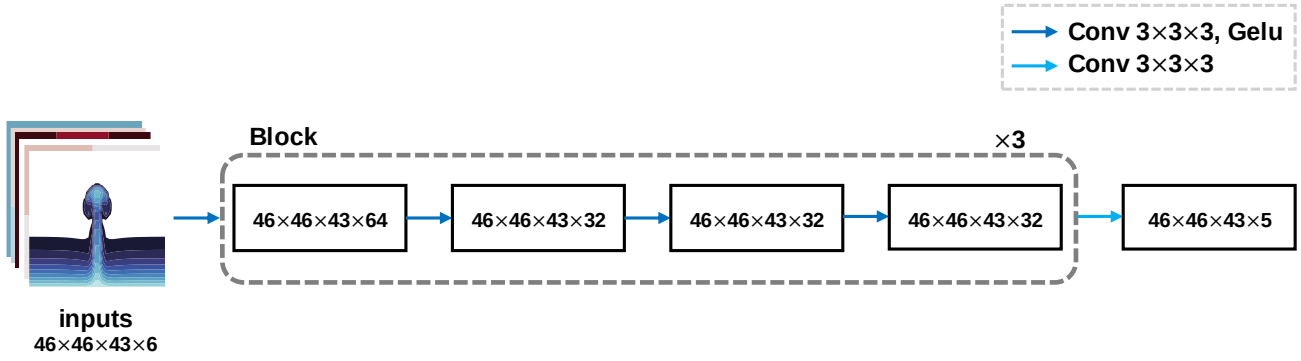

Figure 1. Model Architecture for the three-dimensional convolutional neural network, where $a \times b \times c \times d$ means width×length×height×channel. The inputs include the density potential temperature perturbation (θ'), pressure perturbation (π'), mixing ratio of water vapor perturbation (q_v'), horizontal and vertical velocity (u, v, w), and the outputs are SGS corrections for the density potential temperature (θ), mixing ratio of water vapor (q_v), horizontal and vertical velocity (u, v, w).

and similarly, such SGS correction terms are applied to the mixing ratio of water vapor (q_v), the horizontal and vertical velocity ($\mathbf{u} = u, v, w$):

$$\bar{\mathbf{u}}_{t_0+\Delta t} = \hat{\bar{\mathbf{U}}}_{t_0+\Delta t} + T_{\mathbf{U}}, \tag{18}$$

$$\bar{q}_{v,t_0+\Delta t} = \hat{\bar{q}}_{v,t_0+\Delta t} + T_{q_v}. \tag{19}$$

2.2.2 Data, Model Structure and Training Configurations

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

The training dataset is based on high-resolution 'truth' simulations (HighRes), which have a grid spacing of 100 m in both horizontal and vertical directions. Five distinct warm bubble cases are included in the training dataset. Each is initialized with different potential temperature perturbations prescribed with θ_c at 1.0, 1.5, 2.0, 2.5, and 3.0 K. A time step of 5 s is used for 30-minute simulations. Then by coarse-graining the HighRes data, the training dataset is generated, with a grid spacing of 600 m in the horizontal and 300 m in the vertical. 90% of the generated dataset is used for training, and 10% is used for testing.

To validate the trained model, two additional cases with $\theta_c = 2.6$ K and 5.0 K are chosen to generate initial conditions for validation simulations These two initial conditions are chosen to be within the training dataset potential temperature perturbation range as well as outside that range to evaluate the capability and generalizability of the trained DL SGS model.

A three-dimensional convolutional neural network (3-D CNN) with a $3 \times 3 \times 3$ kernel and the 'GELU' activation function (Hendrycks and Gimpel, 2023) is designed as the structure of the DL SGS, shown in Figure 1. A dry case and a moist case are trained in this paper, respectively. For the moist case, the density potential temperature perturbation (θ'), pressure perturbation(π'), mixing ratio of water vapor perturbation(q_v'), horizontal and vertical velocity (u, v, w) are used as inputs for the 3-D CNN, with ghost points reserved to preserve the physical information at the boundaries. The outputs are SGS corrections for the density potential temperature (θ_ρ), mixing ratio of water vapor (q_v), horizontal and vertical velocity (u, v, w). The dry case is trained with a similar procedure, but it excludes water vapor in both the input and output variables. It should be noticed here that the outputs with ghost points are then stripped of their boundary extensions and repadded with new ghost points to maintain numerical stability. Moreover, all the physical quantities are min-max normalized to a unified range of $[0, 1]$ before being input to the CNN model to avoid unit-induced disparities in data distribution.

The overall training flow can be summarized as follows. At the beginning of each training step, a numerical integration step is performed for the dynamical core from given initial states, which are coarse-grained from the high-resolution benchmark simulations. Then the integration results are used as inputs to the DL model to yield the SGS correction terms, which are applied to the direct integration results to get the new physical states. Such new physics states serve as the initial states for the next numerical integration step. This loop is iterated for N look-ahead steps. The mean squared error (MSE) at each step is accumulated to be the total loss of the current training step, with which we use the Adam (Kingma and Ba, 2017) optimizer to adjust the DL model parameters. In this study $N = 5$.

The training process is quite sensitive to the initialization of the DL model parameters. Thus it begins with two look-ahead steps as a pre-training strategy, and gradually adds up to five look-ahead steps to keep the model's numerical stability. Also, to mitigate the influence of potential rounding errors, double-precision (float64) is employed throughout the training process.

Moreover, after each correcting step, the corrected values of the mixing ratio of water vapor below zero are artificially reset to zero. This is because the DL model can not distinguish the physical meaning of the input variables and will generate unreasonable outputs, such as negative values for water vapor, which is against the laws pf physics and will strongly impact the integrated results of the numerical simulation in the next time step during the training loop, and accordingly make the DL model hard to be optimized.

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

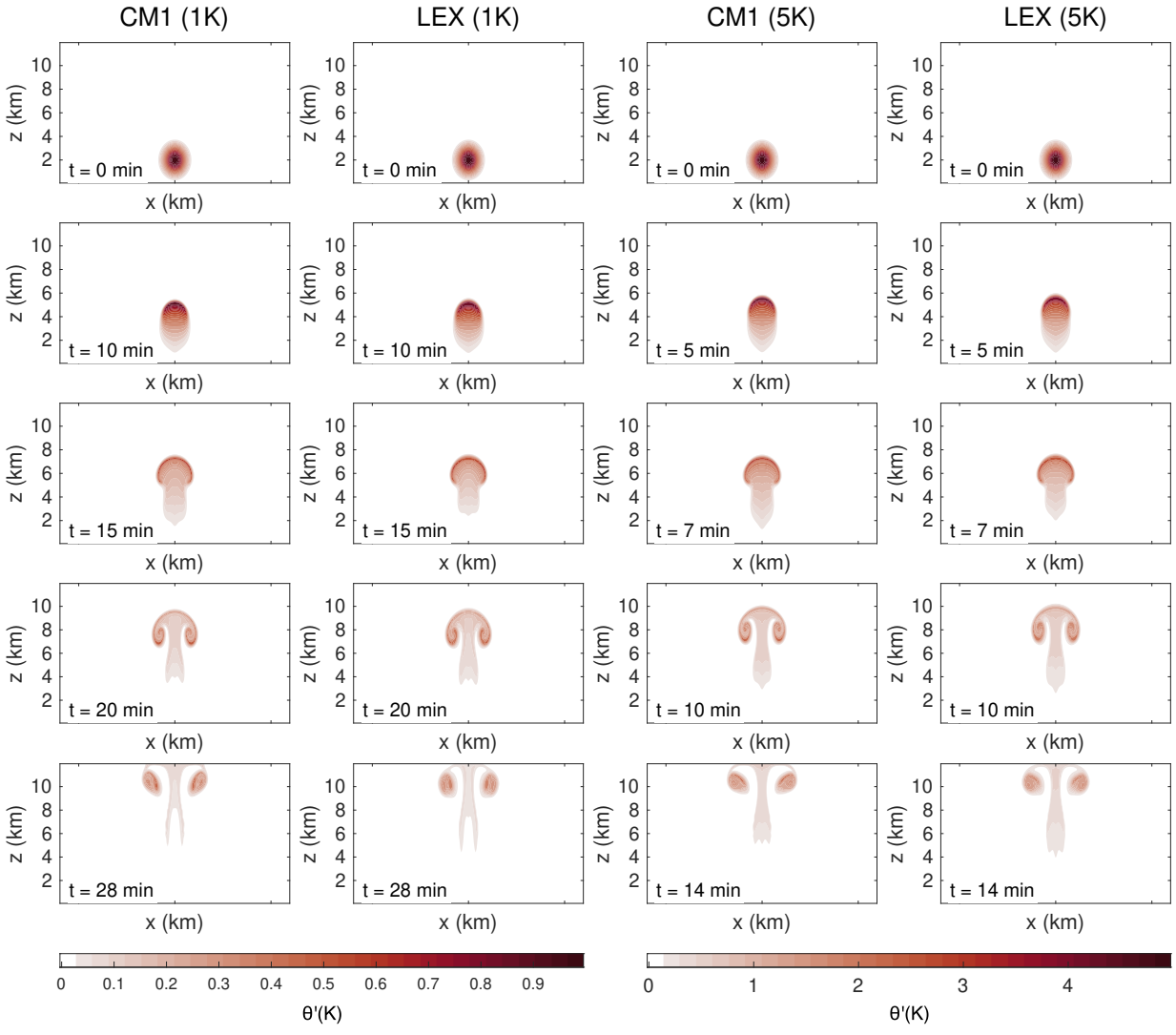

Figure 2. Snapshots of simulated potential temperature perturbations (θ') in CM1 and LEX under different initial central perturbations: 1 K at t = 0, 10, 15, 20, and 28 min (the first and second columns) and 5 K at t = 0, 5, 7, 10, and 14 min (the third and fourth columns).

3 LEX Validation

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

The accuracy of the LEX model is validated against high-fidelity simulation results obtained from the fully compressible Cloud Model 1 (CM1) (Bryan and Fritsch, 2002). Two moist cases with different initial central potential temperature perturbations are tested. The first and second columns of Figure 2 show the snapshots of the simulated potential temperature perturbations in CM1 and LEX with $\theta_c = 1K$ at $t = 0, 10, 15$ and 20 min. The third and fourth columns of Figure 2 show the simulated results with $\theta_c = 5K$ at $t = 0, 5, 7, 10$ and 14 min. Comparing the two pairs, it is evident that the simulation results of the LEX model are identical with those of CM1, regardless of the initial potential temperature perturbations, indicating the reliability and accuracy of the LEX model code in JAX. Figure S1 and Figure S2 further confirm the robustness of the LEX model by presenting the simulated results for the mixing ratio of water vapor and pressure perturbations with $\theta_c = 1K$ and $\theta_c = 5K$. However, because LEX calculates pressure based on the pseudo-compressible approximation, subtle differences appear after the thermal reaches the upper boundary of the domain in pressure simulations.

Furthermore, the figure shows that different initial potential temperature perturbations make the thermal rise at different speeds. A higher temperature brings the thermal a faster rising speed. The bubble with $\theta_c = 5\,\mathrm{K}$ takes around half of the simulated time to rise to a similar height as the bubble with $\theta_c = 1\,\mathrm{K}$. The acceleration of the rising speed due to a warmer initial potential temperature perturbation makes an evident difference for the simulated physical states of the numerical model in each integration time step. Therefore, though the evolution patterns are quite similar in the five training cases, they still provide rich and complex variation in the training dataset, sufficient for the training and testing of the DL SGS model below. The additional validation case initiated with $\theta_c = 5\,\mathrm{K}$ is far outside of the training dataset range, further proving the generalizability of the trained DL model.

4 Preliminary Testing for Deep Learning-based Parameterizations

4.1 Conventional SGS Model

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

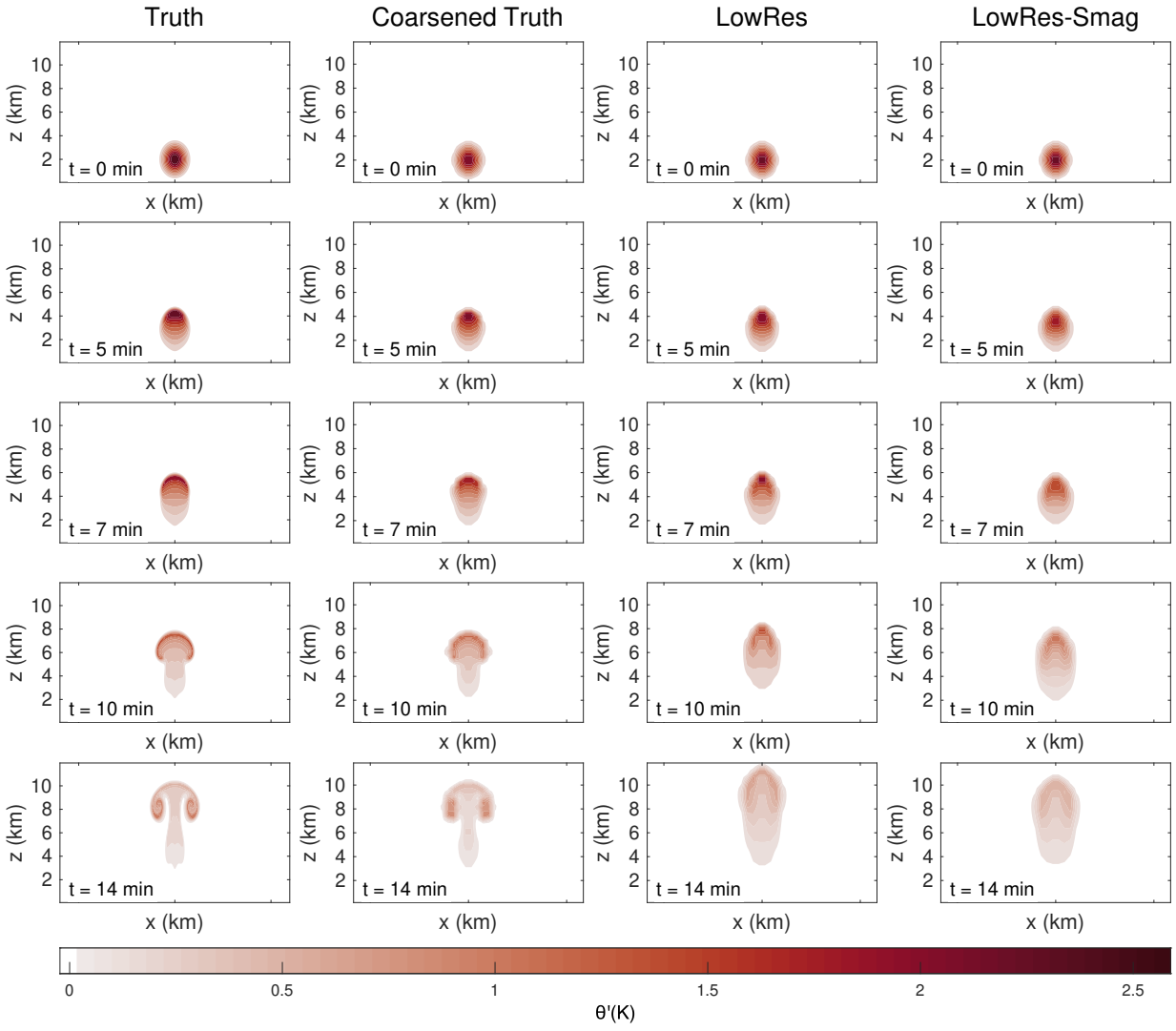

Figure 3. Snapshots of simulated potential temperature perturbations (θ') at t = 0, 5, 7, 10, and 14 min, with $\theta_c = 2.6K$, where the first column is the 'Truth' simulation with a high resolution of $100 \times 100 \times 100m$, the second column is coarsen-grained from the Truth simulation with a coarse resolution of $600 \times 600 \times 300m$, the third column is the numerical simulation results with the coarse grids, and the forth column is the LowRes simulation with the Smagorinsky scheme to deal with the SGS turbulence.

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

This section first tests the reliability of the classic Smagorinsky scheme in the gray zone. The following testing simulations are run with LEX for comparisons: (1) a 'Truth' simulation with a high resolution of $100 \times 100 \times 100$ m as the referenced ground truth; (2) the 'Coarsened Truth', which is coarse-grained from the 'Truth' simulations, to serve as the baseline on the coarse grids; (3) a 'LowRes' simulation which is run on the coarse grids with the resolution of $600 \times 600 \times 300$ m; and (4) the 'LowRes-Smag' simulation in which the conventional Smagorinsky scheme (Smagorinsky, 1963; Shi et al., 2018) is used to solve the SGS turbulence on the coarse grids.

Figure 3 and Figure S3 clearly illustrate that the LowRes simulation tends to have a faster rising speed than the baseline simulation and it fails to resolve the correct symmetric rotor structure at the warm bubble edges due to the relatively large grid spacing. Using the Smagorinsky scheme to solve the SGS motions can slightly help correct the rising speed of the LowRes simulation. The rising speed of the thermal is lowered and is adjusted to be similar to the referenced truth state with the Smagorinsky scheme. However, the expected symmetric rotor structure cannot be simulated properly, and the intensity of the large-scale motion is also wrongly estimated, under-estimation in this case. This shows that the conventional parameterization schemes still have limitations in approximating the appropriate physical dynamics and providing reliable numerical predictions in the gray zone. Therefore, improved parameterization schemes need to be developed to handle the SGS motions more precisely in such grid spacings.

4.2 Deep Learning-based SGS Model

4.2.1 Dry Case

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

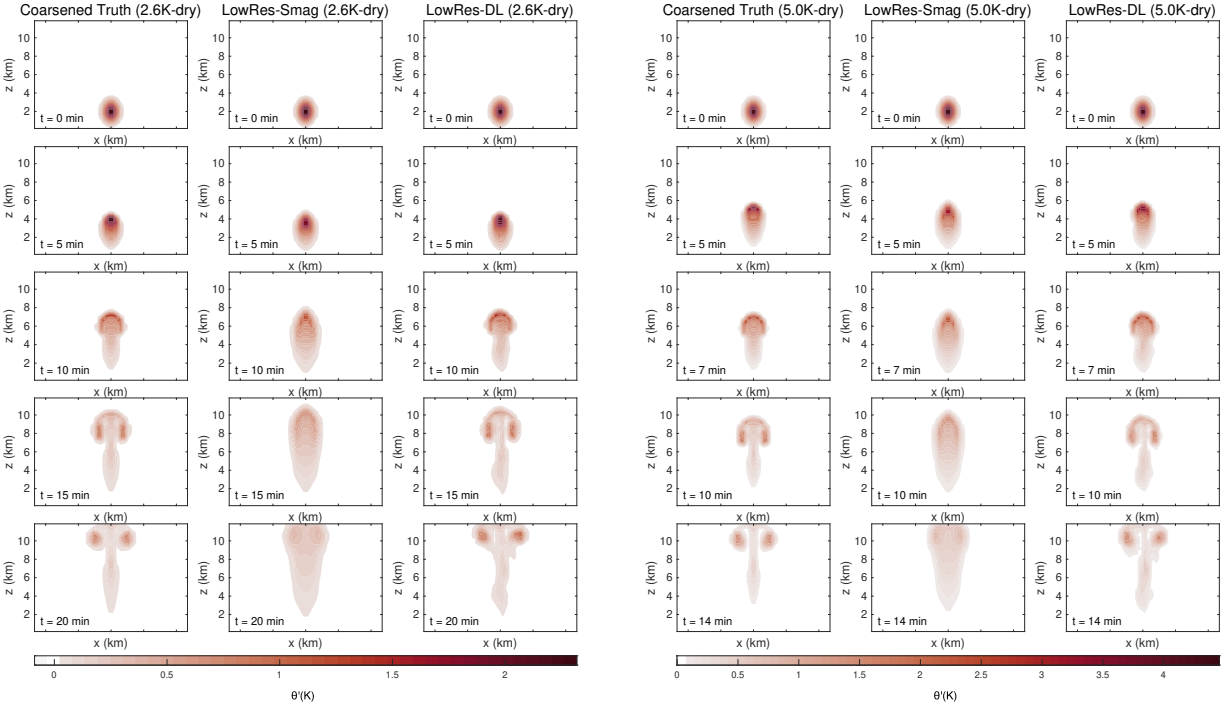

Figure 4. Snapshots of simulated potential temperature perturbations (θ') at t = 0, 5, 10, 15, and 20 min for dry cases, with $\theta_c = 2.6K$ (the first to the third columns), and $\theta_c = 5.0K$ (the fourth to the sixth columns), where the first and fourth columns are the 'Coarsened Truth' simulations with a coarse resolution of $600 \times 600 \times 300m$, the second and fifth columns are the 'LowRes-Smag' simulations with the Smagorinsky scheme to deal with the SGS turbulence, and the third and sixth columns are 'LowRes-DL' simulations with the trained CNN model to serve as the turbulence parameterization scheme.

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

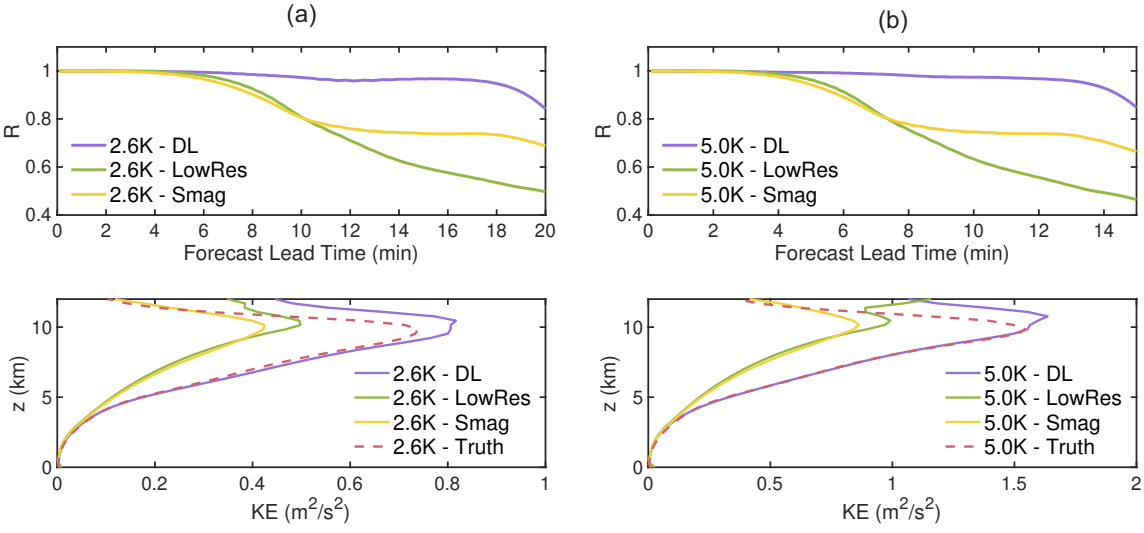

Figure 5. The correlation coefficient (R) and energy profile for dry cases of the Coarsened Truth, LowRes, LowRes-Smag and LowRes-DL simulations, with $\theta_c = 2.6K$ (a), and $\theta_c = 5.0K$ (b).

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

The training for the DL-based SGS model is conducted first with a dry case. The online testing results are shown in Figure 4. The LowRes-DL is the result of the coarse simulation with the trained CNN model's correction for SGS tendencies. Compared to the conventional Smagorinsky scheme, the trained DL model is able to simulate the right rising speed, and can further develop the proper symmetric structure of the thermal, showing its superiority for turbulence predictions in the gray zone. Moreover, results are similar no matter whether the initial potential temperature is within the range of the training dataset or out of that, indicating the good generalization capability of this CNN model.

The quantitative assessments of the DL model's forecast performance are also conducted with the correlation coefficient (R) and the kinetic energy (KE) profile. As is seen in Figure 5, the numerical simulation with the DL-based SGS model can maintain a high level of correlation with the baseline (Coarsened Truth) during the applicable testing period. What's more, compared to the LowRes and LowRes-Smag prediction results, LowRes-DL can better forecast the small-scale turbulence motions, which is proved by the highly aligned maximum peak height of the kinetic energy and the corresponding magnitude with those of the baseline.

4.2.2 Moist Case

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

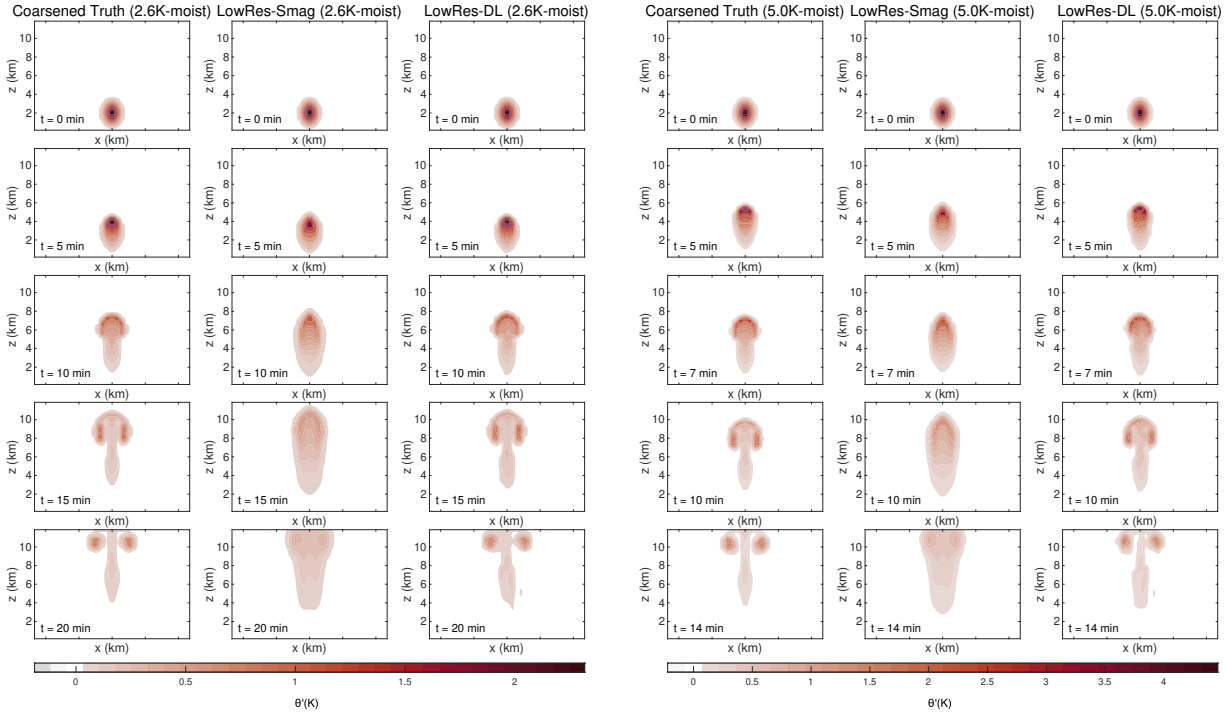

Figure 6. Snapshots of simulated potential temperature perturbations (θ') at t = 0, 5, 10, 15, and 20 min for moist cases, with $\theta_c = 2.6K$ (the first to the third columns), and $\theta_c = 5.0K$ (the fourth to the sixth columns), where the first and fourth columns are the 'Coarsened Truth' simulations with a coarse resolution of $600 \times 600 \times 300m$, the second and fifth columns are the 'LowRes-Smag' simulations with the Smagorinsky scheme to deal with the SGS turbulence, and the third and sixth columns are 'LowRes-DL' simulations with the trained CNN model to serve as the turbulence parameterization scheme.

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

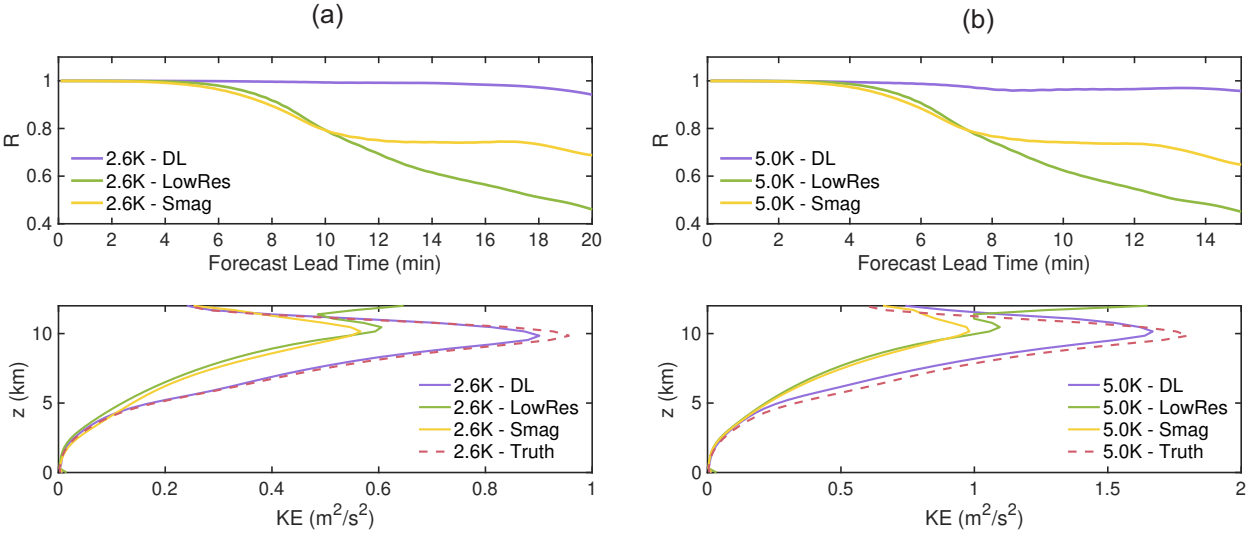

Figure 7. The correlation coefficient (R) and kinetic energy profile for moist cases of the Coarsened Truth, LowRes, LowRes-Smag and LowRes-DL simulations, with $\theta_c = 2.6\,\mathrm{K}$ (a), and $\theta_c = 5.0\,\mathrm{K}$ (b).

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

Table 1. Computational performance comparison of CM1 and LEX. The CM1 was run using 64 cores of an AMD Ryzen Threadripper 3990X. The LEX was run on NVIDIA RTX A6000 GPU. No SGS models were used in CM1 or LEX simulations.

Model	Resolution (m)	Time Step (s)	Hardware (CPU/GPU)	Integration Time (min)	IO/Setup Time (s)	Computing Time (s)
CM1	100×100×100	2	CPU (64 cores)	20	40.00	789.00
LEX	100×100×100	2	GPU	20	171.97	548.01
	100×100×100	12	GPU	20	171.97	84.75

The CNN-based SGS model trained with the moist case presents similar results with those of the dry case. As illustrated in Figures 6 and 7 and Supplementary Figure S4, when effects of the water vapor are included in the training process, the trained CNN model can still perform with satisfying simulation capabilities for SGS corrections, producing proper thermal rising speed and rotor-structure. Also, both the temporal and spatial evaluations show the high consistency between the LEX simulation with the trained CNN model and the high-resolution baseline simulation in the moist condition.

However, though the predicted thermal structure is overall identical with the baseline simulation, it is not strictly symmetric. This is because the correction terms generated from the DL model add small amplitude errors to the original thermal structure. The thermal's structure is asymmetric after some DL model correcting steps and then is hard to restore strict symmetry. Moreover, the training of the CNN model in the moist case also tends to be more sensitive to the model initialization and more difficult to converge, compared to the training process with the dry air. A possible reason might be the lack of microphysics in LEX which results in the thermal unable to condense at the upper domain. The reasons are still not clear and need to be investigated in the future work.

In this section, the experiments of the dry and moist cases proved the capability of the LEX model to be used for training a DL SGS model in a physics-DL hybrid framework. The newly developed DL model can well represent SGS motions in the gray zone, being a more accurate and reliable parameterization tool than conventional schemes.

5 Computing Time Comparisons

5.1 LEX Compared to CM1

The computational costs are compared in this section. As mentioned in Section 1, LEX has better numerical stability and is expected to show faster computing speed with JAX acceleration techniques. Using the conventional CM1 model as the benchmark model, Table 1 shows that employing the same time step of two seconds to run a 20-minute simulation, the total computing time for LEX is 789 s using 64 cores, while the LEX run takes 548 s on one GPU. If we assume the CM1 simulation using one core would take 64 times the computing time, then the ratio of computation time for CM1 on one core to LEX on one GPU card is 92:1. Because the 20-minute simulation is a relatively short integration period, leading to the LEX setup and

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

Table 2. Computational speed comparison of DL-based SGS model and conventional Smagorinsky Scheme

Model	Resolution (m)	Hardware	Precision	Parameterization Scheme	Integration Time (min)	Time Step (s)	Computing Time (s)
LEX	600×600×300	GPU	Single	N/A	20	5	3.89
LEX	600×600×300	GPU	Double	N/A	20	5	7.86
LEX	600×600×300	CPU	Single	Smagorinsky	20	5	96.97
LEX	600×600×300	GPU	Single	Smagorinsky	20	5	4.57
LEX	600×600×300	GPU	Double	Smagorinsky	20	5	9.97
LEX+DL	600×600×300	GPU	Double	DL	20	5	36.87
LEX+DL	600×600×300	GPU	Single+Double	DL	20	5	10.08

just-in-time compilation time accounting for a significant fraction of the total running time. However, if we run the LEX for a substantially longer time, the compilation and setup time probably can be ignored.

Furthermore, at the resolution of $100 \times 100 \times 100$ m, the longest time step for CM1 to maintain numerical stability is two seconds, but for LEX, it can be up to twelve seconds, thanks to its acoustic-wave-filtering equations and the strong stability integration scheme SSPRK3. As a result, LEX's running time can be further reduced by a factor of 1/6. That means the ratio of CM1 on one core to LEX on one GPU card is almost 600. This demonstrate the great application potential of LEX to run for long simulations.

The effectiveness of GPU acceleration is also shown in Table 2. Calculating with the same resolution and a five-second time step for a 20-minute integration time, LEX with the Smagorinsky scheme runs around 21 times faster on the GPU than on the CPU, excluding the just-in-time compilation time.

5.2 DL-based SGS Model Compared to Conventional Smagorinsky Scheme

LEX can be trained with a DL-based SGS model and succeed in numerical predictions in the gray zone, but whether such physics-DL hybrid models can be applied in real weather forecasts also relies on their computational costs. The parameterizations for SGS processes are only one part of the entire numerical weather predictions, thus, they are expected to run at a fast speed. Since the DL model is trained with the double-precision float64, its computing time is first evaluated with the same precision to run the hybrid model. However, Table 2 shows that when running with float64, the LEX-DL model with a five-second time step takes around eight times of the computing time of the LEX-Smag model using float32 with a same time step, not satisfying performance. One reason for this is float64 needs more computational resources than float32, and the other is float64 convolutions are not supported by XLA now, which further increases its computational costs. Therefore, to enhance the computing efficiency of the DL-based SGS model, a mixed precision mode is applied, where the LEX model keeps the double-precision of float64, but the inference of the DL model uses a single-precision of float32 instead. Moreover,

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

we posteriorly find the mixed precision mode is also applicable for the DL model training process once the model parameters have been sufficiently pretrained, further reducing the training computational expenses.

The single precision cannot be applied to the entire LEX-DL model, which will cause great numerical instabilities and make the integration hard to continue. However, when it is employed only for the DL inference part, experiments show that almost no difference will be made to the simulation results (Figure S5, S6). This need of double precision for the dynamical core might be because the DL model will cause some tiny noise towards the stable thermal structure. Different precisions of the noise make little difference, but the impact of the initial error it causes is quite sensitive to the round-off errors generated in the latter model integration time step. With a lower calculation precision, the initial error will be larger after one numerical integration time step due to the larger round-off error, compared to the double precision. This enhanced error will be input to the DL model and then cause the further enhanced error for the next numerical integration time step. This process repeats cyclically and will make the errors accumulate quickly to such an extent that the integration of the numerical model collapses.

As a result, only applying the single precision to the DL model inference and using the double precision to integrate the LEX model can perfectly maintain the numerical stability and model forecast capacity, as well as significantly reduce the computational cost at the same time. As is shown in Table 2, using the same time step of five seconds, the LEX-DL model with a mixed precision mode can achieve 73% computing time reduction than that with the double precision, which only needs $10.08s$ to complete the integration task after the compilation. A further comparison is also conducted and it is found that though the fastest speed the hybrid model can achieve now is two times that of the LEX-Smag model with the single precision, it is almost the same with the computing time of the LEX-Smag model with a double precision for a 20-minute simulation, which means the DL model's inference time has little difference with the computing time of the classic Smagorinsky, and the main cause of the time cost comes from the higher calculation precision. This can be further proven from the testing experiments for LEX without any parameterization schemes. The computing time difference between LEX with a single and double precision is around four seconds, which is almost the same with that of the LEX-Smag with a single precision and LEX-DL with a mixed precision. Thus, if a fancier DL model were applied to make the LEX-DL model simulate with a single precision, the LEX-DL model would be hopeful to have a comparable computing speed with the LEX-Smag with a single precision.

6 Conclusions

As the model resolutions are entering the kilometer-scale range, parameterizations for SGS motions in the gray zone remain key obstacles in today's numerical weather forecasts, because turbulence and convection can only be partially resolved and conventional parameterization schemes are no longer applicable in the gray zone. LES models are always valuable and important tools for studying small-scale turbulence motions in the field of atmospheric science. They are used to compare the different SGS parameterization schemes and help develop improved SGS models for different flows (Remmler and Hickel, 2013; Khani and Waite, 2015). However, LESs that are available for large domains are still lacking to date. In this background, the new LES model written with JAX, LEX, is developed in this paper. By validating its simulation results with those of the

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

traditional CM1 model using different initial conditions for a simple thermal case, LEX is proven to be a reliable and robust LES model.

Moreover, LEX can be applied for simulations on large domains with its fast computation speed. With GPU acceleration, the acceleration tools from JAX, and good numerical stability that allows larger time steps, running LEX on one GPU is as fast as running CM1 on 600 CPU cores. One disadvantage of LEX is that the just-in-time compilation takes much time. Therefore, LEX is better used for long-period simulations. As the integration time increases, the advantage of LEX model's fast computational speed will become increasingly apparent, compared to the other traditional LES models.

The newly developed LEX model is also auto-differentiable. To report its differentiability, based on LEX, a DL-based SGS model is further trained for SGS parameterization in the gray zone for the thermal bubble case. A simple 3-D CNN is applied to produce correction terms for the prognostic variables. The coupled online training of the physics-DL hybrid model integrates the dynamics in the loops every epoch. The trained model exhibits excellent capability to correct the dynamical core integration and simulate the symmetric rotor structure of the rising thermal in the gray zone. The traditional Smagorinsky scheme is also tested and exhibits poorer performance, with the thermal perturbation wrongly estimated and failing to produce the rotor structure.

The DL model is not only more reliable in representing the SGS turbulence in the gray zone, but its inference time can also be comparable to that of the conventional parameterization scheme. However, the preliminary results show that a mixed mode of precision is still necessary to apply a LEX-DL model, resulting in the total simulation time increasing – a 20-minute simulation with a time step of five seconds needs ten seconds to be completed. This probably can be optimized in the future with a more suitable DL model architecture. Overall, hybrid models are promising tools to be applied for SGS representations for real atmospheric forecasts.

In this study, coupling LEX to the DL model's training process still presents some difficulties. For example, both for the dry and moist case, the use of double precision is necessary for the pretraining process to maintain numerical stability and avoid gradient explosion, which requires large amounts of computational time and resources. The dry case can demonstrate stable training dynamics and achieve asymptotic convergence with double precision. But for the moist case, the optimization shows oscillatory convergence behavior. What's more, the trained model will still generate tiny noise and cannot give a perfect symmetric structure in the simulations. This paper is intended to show the preliminary results for the DL-based SGS model to prove the auto-differentiation capacity of LEX. In our future work, we will try to improve the DL model and solve these specific problems.

Overall, LEX can now be utilized with accuracy and fast-computing speed. It is auto-differentiable so that corresponding DL-based SGS models can be trained to provide high-fidelity parameterizations for SGS motions in the gray zone. The development of LEX is expected to help deepen knowledge of the small-scale turbulence processes and enable the future development of more reliable parameterization schemes in the gray zone.

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

Code availability. The current version of the LEX model is publicly available on Github at https://github.com/MetLab-HKUST/LEX under the MIT license. LEX codes, and scripts for producing figures are archived on Zenodo under https://doi.org/10.5281/zenodo.15486687 (Zhu et al., 2025a). Related data used in this study can be accessed from Zenodo under https://doi.org/10.5281/zenodo.15730773 (Zhu et al., 2025b). CM1 version cm1r21.1 is used in this study, which can be accessed from https://github.com/NCAR/CM1.

Author contributions. SXM proposed this study and developed LEX. ZXY developed the dataset, conducted the preliminary training and evaluations. ZXY, QYQ, and SXM contributed to the paper writing together.

Competing interests. The authors have no competing interests to declare.

Acknowledgements. The work is substantially supported by a grant from the Research Grants Council (RGC) of the Hong Kong Special Administrative Region, China (Project Reference Number: AoE/P-601/23-N). Additionally, XZ and XS also received support from the RGC Grant HKUST-16301322 and the Center for Ocean Research in Hong Kong and Macau (CORE), a joint research center between the Laoshan Laboratory and the Hong Kong University of Science and Technology (HKUST). YQ acknowledges funding from NSF through the Learning the Earth with Artificial intelligence and Physics (LEAP) Science and Technology Center (STC) (Award #2019625).

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

References

Blossey, P. N., Bretherton, C. S., Cheng, A., Endo, S., Heus, T., Lock, A. P., and van der Dussen, J. J.: CGILS Phase 2 LES intercomparison of response of subtropical marine low cloud regimes to CO2 quadrupling and a CMIP3 composite forcing change, J. Adv. Model. Earth Sy., 8, 1714–1726, https://doi.org/10.1002/2016MS000765, 2016.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q.: JAX: composable transformations of Python+NumPy programs, http://github.com/jax-ml/jax, 2018.

Bretherton, C. S.: Insights into low-latitude cloud feedbacks from high-resolution models, Philos. T. Roy. Soc. A, 373, https://doi.org/10.1098/rsta.2014.0415, 2015.

Bryan, G. H. and Fritsch, J. M.: A benchmark simulation for moist nonhydrostatic numerical models, Mon. Weather Rev., 130, 2917 – 2928, https://doi.org/10.1175/1520-0493(2002)130<2917:ABSFMN>2.0.CO;2, 2002.

Chow, F. K., Schär, C., Ban, N., Lundquist, K. A., Schlemmer, L., and Shi, X.: Crossing multiple gray zones in the transition from mesoscale to microscale simulation over complex terrain, Atmosphere-Basel, 10, https://doi.org/10.3390/atmos10050274, 2019.

Demeshko, I., Maruyama, N., Tomita, H., and Matsuoka, S.: Multi-GPU implementation of the NICAM atmospheric model, in: Euro-Par 2012: Parallel Processing Workshops, pp. 175–184, Springer, https://doi.org/10.1007/978-3-642-36949-0_20, 2013.

Donahue, A. S., Caldwell, P. M., Bertagna, L., Beydoun, H., Bogenschutz, P. A., Bradley, A. M., Clevenger, T. C., Foucar, J., Golaz, C., Guba, O., Hannah, W., Hillman, B. R., Johnson, J. N., Keen, N., Lin, W., Singh, B., Sreepathi, S., Taylor, M. A., Tian, J., Terai, C. R., Ullrich, P. A., Yuan, X., and Zhang, Y.: To exascale and beyond—the simple cloud-resolving E3SM atmosphere model (SCREAM), a performance portable global atmosphere model for cloud-resolving scales, J. Adv. Model. Earth Sy., 16, e2024MS004 314, https://doi.org/10.1029/2024MS004314, 2024.

Durran, D. R.: A physically motivated approach for filtering acoustic waves from the equations governing compressible stratified flow, J. Fluid Mech., 601, 365–379, https://doi.org/10.1017/S0022112008000608, 2008.

Durran, D. R.: Ordinary Differential Equations, in: Numerical Methods for Fluid Dynamics: With Applications to Geophysics, pp. 35–87, Springer, https://doi.org/10.1007/978-1-4419-6412-0_2, 2010.

Hendrycks, D. and Gimpel, K.: Gaussian error linear units (GELUs), https://doi.org/10.48550/arXiv.1606.08415, 2023.

Honnert, R., Efstathiou, G. A., Beare, R. J., Ito, J., Lock, A., Neggers, R., Plant, R. S., Shin, H. H., Tomassini, L., and Zhou, B.: The atmospheric boundary layer and the "Gray Zone" of turbulence: A critical review, J. Geophys. Res.-Atmos., 125, e2019JD030 317, https://doi.org/10.1029/2019JD030317, 2020.

Häfner, D., Nuterman, R., and Jochum, M.: Fast, cheap, and turbulent—global ocean modeling with GPU acceleration in Python, J. Adv. Model. Earth Sy., 13, e2021MS002 717, https://doi.org/10.1029/2021MS002717, 2021.

Jadhav, K. and Chandy, A. J.: Assessment of SGS models for large eddy simulation (LES) of a stratified taylor–green vortex, Flow Turbul. Combust., 106, 37–60, https://doi.org/10.1007/s10494-020-00175-5, 2021.

Khani, S. and Waite, M. L.: Large eddy simulations of stratified turbulence: the dynamic Smagorinsky model, J. Fluid Mech., 773, 327–344, https://doi.org/10.1017/jfm.2015.249, 2015.

Kingma, D. P. and Ba, J.: Adam: a method for stochastic optimization, https://doi.org/10.48550/arXiv.1412.6980, 2017.

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., and Hoyer, S.: Machine learning–accelerated computational fluid dynamics, P. Natl. Acad. Sci. U.S.A., 118, e2101784 118, https://doi.org/10.1073/pnas.2101784118, 2021.

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., Klöwer, M., Lottes, J., Rasp, S., Düben, P., Hatfield, S., Battaglia, P., Sanchez-Gonzalez, A., Willson, M., Brenner, M. P., and Hoyer, S.: Neural general circulation models for weather and climate, Nature, 632, 1060–1066, https://doi.org/10.1038/s41586-024-07744-y, 2024.

Prein, A. F., Rasmussen, R., and Stephens, G.: Challenges and advances in convection-permitting climate modeling, B. Am. Meteorol. Soc., 98, 1027 – 1030, https://doi.org/10.1175/BAMS-D-16-0263.1, 2017.

Price, E., Mielikainen, J., Huang, M., Huang, B., Huang, H.-L. A., and Lee, T.: GPU-accelerated longwave radiation scheme of the rapid radiative transfer model for general circulation models (RRTMG), IEEE J. Sel. Top. Appl., 7, 3660–3667, https://doi.org/10.1109/JSTARS.2014.2315771, 2014.

Qu, Y. and Shi, X.: Can a machine learning–enabled numerical model help extend effective forecast range through consistently trained subgrid-scale models?, Artif. Intell. Earth Syst., 2, e220 050, https://doi.org/10.1175/AIES-D-22-0050.1, 2023.

Qu, Y., Bhouri, M. A., and Gentine, P.: Joint parameter and parameterization inference with uncertainty quantification through differentiable programming, https://doi.org/10.48550/arXiv.2403.02215, 2024.

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), Geosci. Model Dev., 13, 2185–2196, https://doi.org/10.5194/gmd-13-2185-2020, 2020.

Remmler, S. and Hickel, S.: Spectral structure of stratified turbulence: direct numerical simulations and predictions by large eddy simulation, Theor. Comp. Fluid Dyn., 27, 319–336, https://doi.org/10.1007/s00162-012-0259-9, 2013.

Sauer, J. A. and Muñoz-Esparza, D.: The FastEddy® Resident-GPU accelerated large-eddy simulation framework: model formulation, dynamical-core validation and performance benchmarks, J. Adv. Model. Earth Sy., 12, e2020MS002 100, https://doi.org/10.1029/2020MS002100, 2020.

Schalkwijk, J., Jonker, H., Siebesma, A., and van Meijgaard, E.: Weather forecasting using GPU-based large-eddy simulations, B. Am. Meteorol. Soc., 96, 715–723, https://doi.org/10.1175/BAMS-D-14-00114.1, 2015.

Schär, C., Fuhrer, O., Arteaga, A., Ban, N., Charpilloz, C., Girolamo, S. D., Hentgen, L., Hoefler, T., Lapillonne, X., Leutwyler, D., Osterried, K., Panosetti, D., Rüdisühli, S., Schlemmer, L., Schulthess, T. C., Sprenger, M., Ubbiali, S., and Wernli, H.: Kilometer-scale climate models: prospects and challenges, B. Am. Meteorol. Soc., 101, E567 – E587, https://doi.org/10.1175/BAMS-D-18-0167.1, 2020.

Shen, Z., Sridhar, A., Tan, Z., Jaruga, A., and Schneider, T.: A library of large-eddy simulations forced by global climate models, J. Adv. Model. Earth Sy., 14, e2021MS002 631, https://doi.org/10.1029/2021MS002631, 2022.

Shi, X. and Fan, Y.: Modulation of the bifurcation in radiative-convective equilibrium by gray-zone cloud and turbulence parameterizations, J. Adv. Model. Earth Sy., 13, e2021MS002 632, https://doi.org/10.1029/2021MS002632, 2021.

Shi, X., Chow, F. K., Street, R. L., and Bryan, G. H.: An evaluation of LES turbulence models for scalar mixing in the stratocumulus-capped boundary layer, J. Atmos. Sci., 75, 1499 – 1507, https://doi.org/10.1175/JAS-D-17-0392.1, 2018.

Shi, X., Enriquez, R. M., Street, R. L., Bryan, G. H., and Chow, F. K.: An implicit algebraic turbulence closure scheme for atmospheric boundary layer simulation, J. Atmos. Sci., 76, 3367 – 3386, https://doi.org/10.1175/JAS-D-18-0375.1, 2019.

Shu, C.-W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: Advanced Numerical Approximation of Nonlinear Hyperbolic Equations. Lecture Notes in Mathematics, pp. 325–432, Springer, https://doi.org/10.1007/BFb0096355, 1998.

Silvestri, S., Wagner, G. L., Hill, C., Ardakani, M. R., Blaschke, J., Campin, J.-M., Churavy, V., Constantinou, N. C., Edelman, A., Marshall, J., Ramadhan, A., Souza, A., and Ferrari, R.: Oceananigans.jl: a Julia library that achieves breakthrough resolution, memory and energy efficiency in global ocean simulations, https://doi.org/10.48550/arXiv.2309.06662, 2024.

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

Silvestri, S., Wagner, G. L., Constantinou, N. C., Hill, C. N., Campin, J.-M., Souza, A. N., Bishnu, S., Churavy, V., Marshall, J., and Ferrari, R.: A GPU-based ocean dynamical core for routine mesoscale-resolving climate simulations, J. Adv. Model. Earth Sy., 17, e2024MS004 465, https://doi.org/10.1029/2024MS004465, 2025.

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

Sridhar, A., Tissaoui, Y., Marras, S., Shen, Z., Kawczynski, C., Byrne, S., Pamnany, K., Waruszewski, M., Gibson, T. H., Kozdon, J. E., Churavy, V., Wilcox, L. C., Giraldo, F. X., and Schneider, T.: Large-eddy simulations with ClimateMachine v0.2.0: a new open-source code for atmospheric simulations on GPUs and CPUs, Geosci. Model Dev., 15, 6259–6284, https://doi.org/10.5194/gmd-15-6259-2022, 2022.

Sullivan, P. P. and Patton, E. G.: The effect of mesh resolution on convective boundary layer statistics and structures generated by large eddy simulation, J. Atmos. Sci., 68, 2395–2415, https://doi.org/10.1175/JAS-D-10-05010.1, 2011.

Tan, Z., Schneider, T., Teixeira, J., and Pressel, K. G.: Large-eddy simulation of subtropical cloud-topped boundary layers: 2. Cloud response to climate change, J. Adv. Model. Earth Sy., 9, 19–38, https://doi.org/10.1002/2016MS000655, 2017.

Teixeira, J. and Cheinet, S.: A simple mixing length formulation for the eddy-diffusivity parameterization of dry convection, Bound.-Lay. Meteorol., 110, 435–453, https://doi.org/10.1023/B:BOUN.0000007230.96303.0d, 2004.

Um, K., Brand, R., Yun, Fei, Holl, P., and Thuerey, N.: Solver-in-the-Loop: learning from differentiable physics to interact with iterative PDE-solvers, https://doi.org/10.48550/arXiv.2007.00016, 2021.

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, Geosci. Model Dev., 10, 3145–3165, https://doi.org/10.5194/gmd-10-3145-2017, 2017.

Verrelle, A., Ricard, D., and Lac, C.: Evaluation and improvement of turbulence parameterization inside deep convective clouds at kilometer-scale resolution, Mon. Weather Rev., 145, 3947–3967, https://doi.org/10.1175/MWR-D-16-0404.1, 2017.

von Rueden, L., Mayer, S., Sifa, R., Bauckhage, C., and Garcke, J.: Combining machine learning and simulation to a hybrid modelling approach: current and future directions, in: Advances in Intelligent Data Analysis XVIII, pp. 548–560, Springer, https://doi.org/10.1007/978-3-030-44584-3_43, 2020.

Watt-Meyer, O., Brenowitz, N. D., Clark, S. K., Henn, B., Kwa, A., McGibbon, J., Perkins, W. A., Harris, L., and Bretherton, C. S.: Neural network parameterization of subgrid-scale physics from a realistic geography global storm-resolving simulation, J. Adv. Model. Earth Sy., 16, e2023MS003 668, https://doi.org/10.1029/2023MS003668, 2024.

Wedi, N., Polichtchouk, I., Dueben, P., Anantharaj, V., Bauer, P., Boussetta, S., Browne, P., Deconinck, W., Gaudin, W., Hadade, I., Hatfield, S., Iffrig, O., Lopez, P., Maciel, P., Mueller, A., Saarinen, S., Sandu, I., Quintino, T., and Vitart, F.: A baseline for global weather and climate simulations at 1 km resolution, J. Adv. Model. Earth Sy., 12, https://doi.org/10.1029/2020MS002192, 2020.

Wicker, L. J. and Skamarock, W. C.: A time-splitting scheme for the elastic equations incorporating second-order Runge–Kutta time differencing, Mon. Weather Rev., 126, 1992 – 1999, https://doi.org/10.1175/1520-0493(1998)126<1992:ATSSFT>2.0.CO;2, 1998.

Wu, E., Yang, H., Kleissl, J., Suselj, K., Kurowski, M. J., and Teixeira, J.: On the parameterization of convective downdrafts for marine stratocumulus clouds, Mon. Weather Rev., 148, 1931–1950, https://doi.org/10.1175/MWR-D-19-0292.1, 2020.

Wyngaard, J. C.: Toward numerical modeling in the "Terra Incognita", J. Atmos. Sci., 61, 1816 – 1826, https://doi.org/10.1175/1520-0469(2004)061<1816:TNMITT>2.0.CO;2, 2004.

https://doi.org/10.5194/egusphere-2025-2568
Preprint. Discussion started: 30 June 2025

Yuval, J., Langmore, I., Kochkov, D., and Hoyer, S.: Neural general circulation models optimized to predict satellite-based precipitation observations, https://doi.org/10.48550/arXiv.2412.11973, 2024.

Zhou, A., Hawkins, L., and Gentine, P.: Proof-of-concept: using ChatGPT to translate and modernize an earth system model from Fortran to Python/JAX, https://doi.org/10.48550/arXiv.2405.00018, 2024.

Zhu, X., Qu, Y., and Shi, X.: LEX v1.4, https://doi.org/10.5281/zenodo.15486687, 2025a.

Zhu, X., Qu, Y., and Shi, X.: LEX v1.4 Data, https://doi.org/10.5281/zenodo.15730773, 2025b.