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

egusphere-2025-2568

General Comments

The present work proposes a deep learning (DL)-based subgrid-scale (SGS) model for large-eddy simulations (LES) of gray zone turbulence in atmospheric flows. The authors implement a JAX-based code framework called LEX which allows end-to-end training of DL-based explicit correction terms to the grid-filtered governing equations. The DL SGS model is trained on coarse-grained data for a rising thermal. A posteriori tests show that the DL SGS model offers improved approximation quality compared to the classical Smagorinsky model.

The overall approach of the paper follows the current trend of hybrid machine learning (ML) and computational fluid dynamics (CFD). The technical novelty of the proposed method is therefore limited. However, applying hybrid ML-CFD to gray zone LES modeling is novel, and the presented results of applying DL-based SGS models to such flows are encouraging. In addition, the LEX framework appears to be a good starting point for future developments of DL-based surrogates for numerical weather prediction.

While the presentation of the manuscript is clear, the paper lacks thorough quantitative analysis and scientific precision in parts. Some claims and statements of the authors are too vague, too imprecise or lack sufficient foundation.

Therefore, the paper can only be reconsidered for publication after a major revision, in which the below listed comments are convincingly addressed.

Specific Comments

Major comments

1. There are statements throughout the manuscript which are scientifically imprecise or lack supporting evidence. For example:

   - lines 5 & 6" "Thus, developing SGS turbulence models for the gray zone requires new LES models, which ... enable new approaches to develop SGS models". I do not understand what the authors mean by this? To my understanding, an LES model already contains a SGS model. Therefore, how can a new LES model enable the development of an SGS model?
   - Line 8: "The new LES model is capable of adequate parallelism ...". How is this claim supported? To my understanding, the LEX model is only run on a single GPU, and parallel simulations are not discussed at all.
   - 21 & 22: "The capability of LESs to simulate small-scale turbulence motion...". This sentence feels scientifically imprecise. LES is supposed to resolve large-scale motions while only modeling the effect of small-scale turbulence on aforementioned large eddies.

- Line 40: "GPU codes are known to run much faster than conventional Fortran or C codes on CPUs". In my opinion, this statement is too generic and scientifically imprecise. While GPUs leverage massive thread-level parallelism, achieving actual code speed up is highly dependent on specific applications.
- Lines 68 & 69: "Existing studies have also shown that JAX-GPU codes enable ... less computational costs when the problem sizes become quite large". What do the authors mean by this? Surely, the computational cost can not decrease with increasing problem size? What does "quite large" mean?
- Line 298: "... float64 convolutions are not supported by XLA now,...". In my opinion, this is not true. float64 support is backened (i.e. hardware) specific. The NVIDIA A6000 GPU does in fact not natively support float64. However, NVIDIA A100 or H100 GPUs provide float64 support on the hardware side. Please correct this statement.

2. What is the motivation to choose the conventional Smagorinsky model as a baseline for comparison? It is well known that the dynamic Smagorinsky model outperforms the classical model in many scenarios. This would be a much stronger baseline for benchmarking the DL-based SGS model.

3. I have the following comments and questions regarding model training:
   - What is the time step size of the coarse-grained simulation? The time step size of the high-resolution simulation is 5s. While the spatial coarse-graining factors are explicitly mentioned, the authors do not mention whether temporal coarse-graining is also applied.
   - What is the rationale for choosing a 6x CG in the horizontal direction and a 3x CG in vertical direction?
   - Please specify the loss functional explicitly. Specifically, the mean-squared error of which quantities is used?
   - Please provide more information regarding hyperparameters of the model training. How many optimization steps are used during training? What is the final training loss level? What is the stopping criterion? What is the learning rate? Is there a learning rate scheduler?
   - The authors mention, that the training for the dry case can "achieve asymptotic convergence" while it "shows oscillatory convergence behavior" for the moist case. Please add loss plots for both scenarios to the manuscript (e.g., to the appendix).
   - Are the authors using custom implementations to propagate the AD gradients through the BiCGSTAB solve?

4. I have the following comments and questions regarding the chosen parameterization and DL model:
   - The standard WENO3- and WENO5-JS schemes are known to be overly dissipative. What is the motivation to choose this parameterization?
   - The DL-SGS model output is applied to θ, u, v, w and q_v. Does the same hold true for the Smagorinsky model? Is the mixing ratio of water vaport q_v a transported quantity or is it post-processed?

5. Validation of the LEX solver with CM1 results are purely qualitative. Please add quantitative comparisons if possible. The authors mention that "results of the LEX are identical with those of CM1". This is an overstatement in my opinion, as Fig. 2 shows visible discrepancies between the two simulations. For example, the lower parts of the thermal are clearly different at later times and the structure of the rotors show differences. The authors should tone down this claim or provide quantitative evidence for it.

6. I would encourage the authors to verify the implementation of the AD gradients with finite-difference analogs. A simple test case can be chosen in which only a single parameter of the DL-based model is optimized with AD and with FD. The error between the two should converge as the step size in the FD approximation approaches zero.

7. The authors mention the trained DL model can "develop the proper symmetric structure of the thermal". This statement is not true. Figs. 4 and 6 clearly show that the DL model breaks symmetry. The authors themselves acknowledge this fact later in Section 4.2.2.

8. It is mentioned that the mixing ratio of water vapor has to be clipped after application of the DL model (Section 2.2.2). I am interested how often this occurs for the trained model over the course of a simulation.

9. I agree with the authors that the DL-based SGS model outperforms the conventional Smagorinsky model for the thermal test case. To my understanding, the DL model is applied after a full integration step while the Smagorinsky model is applied per stage (i.e., thrice per integration step). Can the authors elaborate on this? It would be very interesting to visualize the output of the DL model to try to understand its improved SGS modeling capabilities. Have the authors done such analyses? Is the model output interpretable? What conclusions can be drawn from it?

10. I have the following comments and questions regarding the computing time comparison:

    - The performance comparison in Section 5 is somewhat misleading. The authors claim that they achieve a 92:1 speed up when comparing the LEX code run on an A6000 GPU with the CM1 code run on a single CPU core. I think the authors are aware that such a comparison is not meaningful at all. Can the authors comment on this?
    - In Section 5.2, the wall-clock time of the DL-based SGS model is compared with the Smagorinsky model. Given the short simulation time, the wall-clock time measurements are strongly influenced by the duration of the just-in-time compilation. I would encourage the authors to simply evaluate the Smagorinsky model and the DL-based SGS model on their own to provide more meaningful WCT measurments or to exclude the duration of the jit-compilation from the performance measurements.

11. The authors should consider citing JAX-Fluids [1, 2] and [3]. JAX-Fluids is a JAX-based fully-differentiable CFD solver for compressible single- and two-phase flows, which is closely connected with the present research. Specifically, JAX-Fluids implements functionality for LES and has been used for end-to-end training of implicit LES models [3].

Minor comments

1. What is the reason for v1.4 in the title of the manuscript? Maybe I have missed it, but it is not mentioned in the remainder of the paper. Is the present work building upon a previous release of the LEX solver?

2. In section 2.1.1, some variables are not defined, including $\epsilon, c_p, c_v, w, g, p_s, R$. While I assume that many of these quantities are well known (presumably, c_p is the heat capacity at constant pressure), it would improve clarity to specify their definition once.

3. Please define the correlation coefficient R and the kinetic energy KE in Section 4.2.

Technical Corrections

1. Please proofread and type-check the manuscript carefully. A couple of typos:

    (a) *In the gray zone, turbulence and convection ...* in line 30.

(b) *the acoustic-wave-filtered equations ... are adopted* in line 84.

(c) *for validation simulations.* in lines 178 & 179.

(d) I think the abbreviation *LESs* is not commonly used.

References

[1] D. A. Bezgin, A. B. Buhendwa, N. A. Adams, Jax-fluids: A fully-differentiable high-order computational fluid dynamics solver for compressible two-phase flows, Computer Physics Communications 282 (2023) 108527. `doi:10.1016/j.cpc.2022.108527`.
URL `https://linkinghub.elsevier.com/retrieve/pii/S0010465522002466`

[2] D. A. Bezgin, A. B. Buhendwa, N. A. Adams, Jax-fluids 2.0: Towards hpc for differentiable cfd of compressible two-phase flows, Computer Physics Communications 308 (2025) 109433. `doi:10.1016/j.cpc.2024.109433`.
URL `https://linkinghub.elsevier.com/retrieve/pii/S0010465524003564`

[3] D. A. Bezgin, A. B. Buhendwa, S. J. Schmidt, N. A. Adams, Ml-iles: End-to-end optimization of data-driven high-order godunov-type finite-volume schemes for compressible homogeneous isotropic turbulence, Journal of Computational Physics 522 (2 2025). `doi:10.1016/j.jcp.2024.113560`.