2023 INCITE Proposal Submission
Proposal

Title: Extreme-Scale Data Assimilation for Predictive Flow Simulations
Principal Investigator: Jonathan MacArt
Organization: University of Notre Dame
Date/Time Generated: 6/16/2022 8:01:26 PM

Section 1: PI and Co-PI Information

Question #1

Principal Investigator: *The PI is responsible for the project and managing any resources awarded to the project. If your project has multiple investigators, list the PI in this section and add any Co-PIs in the following section.*

Principal Investigator

First Name

Jonathan

Last Name

MacArt

Organization

University of Notre Dame

Email

jmacart@nd.edu

Work Phone

574-631-6676

Address Line 1

369 Fitzpatrick Hall of Engineering

Address Line 2

(No answer given.)

City

Notre Dame

State

IN

Zip Code

46556

Question #2

Co-PI (s)

First Name

Justin

Last Name

Sirignano

Organization

University of Oxford

Email

Justin.Sirignano@maths.ox.ac.uk

Question #3

Institutional Contact: *For the PI's institution on the proposal, identify the agent who has the authority to review, negotiate, and sign the user agreement on behalf of that institution. The person who can commit an organization may be someone in the contracts or procurement department, legal, or if a university, the department head or Sponsored Research Office or Grants Department.*

Institutional Contact

Institutional Contact Name

Lori McDonald

Institutional Contact Phone

574-631-1107

Institutional Contact Email

lmcdona2@nd.edu

Section 2: Project Information

Question #1

Select the category that best describes your project.

Research Category

Engineering: Fluids and Turbulence

Question #2

Please provide a project summary in two sentences that can be used to describe the impact of your project to the public (50 words maximum)

Project Summary

The project enables extreme-scale data assimilation for simulations of external aerodynamics, leveraging both high-fidelity numerical data and sparse experimental data. It advances the state-of-the-art for predictive accuracy, computational cost, and model-based design of real-world aerodynamic applications, which will enable faster and higher-risk design cycles.

Section 3: Early Career Track

Question #1

Early Career

Starting in the INCITE 2022 year, INCITE is committing 10% of allocatable time to an Early Career Track in INCITE. The goal of the early career track is to encourage the next generation of high-performance computing researchers. Researchers within 10 years from earning their PhD (after December 31st 2012) may choose to apply. Projects will go through the regular INCITE Computational Readiness and Peer Review process, but the INCITE Management Committee will consider meritorious projects in the Early Career Track separately.

***Who Can Apply**: Researchers less than 10 years out from their PhD that need LCF-level capabilities to advance their overall research plan and who have not been a previous INCITE PI.*

How to Apply:

In the regular application process, there will be a check-box to self-identify as early career.

- *The required CV should make eligibility clear.*
- *If awarded, how will this allocation fit into your overall research plan for the next 5 years?*

Projects will go through the regular INCITE review process. The INCITE Program is targeting at least 10% of allocatable time. When selecting the INCITE Career Track, PIs are not restricted to just competing in that track.

- *What is the Early Career Track?*
  - *The INCITE Program created the Early Career Track to encourage researchers establishing their research careers. INCITE will award at least 10% of allocatable time to meritorious projects.*
- *Will this increase my chances of receiving an award?*
  - *Potentially, this could increase chances of an award. Projects must still be deemed scientifically meritorious through the review process INCITE uses each year.*
- *What do I need to do to be considered on the Early Career Track?*
  - *In the application process, select 'Yes' at 'If you are within 10 years of your PhD, would you like to be considered in the Early Career Track?' You will need to write a paragraph about how the INCITE proposal fits into your 5-year research and career goals.*
- *What review criteria will be used for the Early Career Track?*
  - *The same criteria for computational readiness and scientific merit will be applied to projects in the Early Career Track as will be applied to projects in the traditional track. The different will be manifest in awards decisions by the INCITE management committee.*

Early Career Track

If you are within 10 years of your PhD, would you like to be considered in the Early Career Track? Choosing this does not reduce your chances of receiving an award.

Yes

If ‘yes’, what year was your PhD? If 'no' enter N/A

2018

If ‘yes’, how will this allocation fit into your overall research plan for the next 5 years? If 'no' enter N/A.

This allocation will provide (a) an extremely valuable turbulent-flow database that will enable my planned research trajectory in AI/ML for fluid mechanics and (b) access to leadership-class distributed ML training resources (Summit, Frontier) to optimize and demonstrate my methods and codes at scale. The project will exercise my planned leadership-class, solver-in-the-loop ML methods for turbulent fluid mechanics, which is the cornerstone of my five-year research plan. Results from the INCITE allocation will inform and enable ongoing, related work in turbulent reacting flows, aviation biofuels, and hypersonic propulsion.

Section 4: INCITE Allocation Request & Other Project Funding/Computing Resources

Question #1

OLCF Summit (IBM / AC922) Resource Request - 2023

Node Hours

1083200

Storage (TB)

502

Off-Line Storage (TB)

497

Question #2

OLCF Frontier (Cray Shasta) Resource Request – 2023

Node Hours

10000

Storage (TB)

10

Off-Line Storage (TB)

0

Question #3

OLCF Frontier (Cray Shasta) Resource Request – 2024

Question #4

OLCF Frontier (Cray Shasta) Resource Request – 2025

Question #5

ALCF Theta (Cray XC40) Resource Request - 2023

Question #6

ALCF Polaris Resource Request - 2023

Question #7

ALCF Polaris Resource Request - 2024

Question #8

ALCF Polaris Resource Request - 2025

Question #9

ALCF Aurora (Intel Xe) Resource Request – 2023

Question #10

ALCF Aurora (Intel Xe) Resource Request – 2024

Question #11

ALCF Aurora (Intel Xe) Resource Request – 2025

Question #12

List any funding this project receives from other funding agencies.

Funding Sources

Question #13

List any other high-performance computing allocations being received in support of this project.

Other High Performance Computing Resource Allocations

Section 5: Project Narrative and Supplemental Materials

Question #1

Using the templates provided here, please follow the INCITE Proposal Preparation Instructions to prepare your proposal. Elements needed include (1) Project Executive Summary, (2) Project Narrative, (3) Personnel Justification and Management Plan, (4) Milestone Table, (5) Publications Resulting from prior INCITE Awards (if appropriate), and (6) Biographical Sketches for the PI and all co-PI's. Concatenate all materials into a single PDF file. Prior to submission, it is strongly recommended that proposers review their proposals to ensure they comply with the proposal preparation instructions.

Concatenate all materials below into a single PDF file.

1. **Project Executive Summary (One Page Max)**

2. **Project Narrative (15 Pages Max)**
3. **Personnel Justification and Management Plan (1 Page Max)**
4. **Milestone Table**
5. **Publications resulting from prior INCITE Awards (if appropriate)**
6. **Biographical Sketches for the PI and all co-PI's.**

FINAL-INCITE-22-Summit-Proposal.pdf
The attachment is on the following page.

PROJECT EXECUTIVE SUMMARY

Machine learning (ML) for optimization and inference tasks has been rapidly adopted across virtually all science and engineering disciplines. While current ML methods can easily scale to leadership-class systems by increasing the number of simultaneous tasks, simply training over more data does not guarantee that the learned models are physically representative in the case of physical systems. Instead, an emerging trend in *physics-constrained ML* seeks to optimize models consistently with the governing partial differential equations (PDEs), thus ensuring the physical realizability of the learned models in predictive simulations.

We have recently developed a theoretically optimal method for physics-constrained ML based on PDE-constrained optimization and adjoint PDEs. The proposed research will further develop these methods to enable **extreme-scale, solver-in-the-loop data assimilation (DA)**. We will test the learned models in turbulent flow problems relevant to aircraft drag reduction, high-efficiency wind turbine blades, and turbomachinery for energy conversion. Extending and proving our DA methods on leadership-class systems will enable simultaneous optimization over hundreds of configurations and flow parameters, which is necessary for highly generalizable models (that is, accurate and stable at conditions different from the training data) in high-dimensional parameter spaces such as turbulent flows. Additionally, leadership-class simulations will provide trusted data at high-enough Reynolds numbers to develop genuinely useful engineering models. Finally, the methodological and software improvements will enable extreme-scale DA in other fields such as atmospheric science, medicine, alternative fuels design, hypersonics, and materials science.

In addition, the proposed research will develop a **multi-fidelity DL approach** based on a hierarchy of training data fidelities. Such an approach is desperately needed in physics-constrained ML, as high-fidelity data is extremely hard to obtain—in fluid mechanics, only moderate Reynolds numbers are accessible by fully resolved simulations, and flight-relevant Reynolds numbers will remain out of reach for several generations of exascale machines. As the fidelity of the training data is progressively reduced, more data will be required to avoid overfitting, which in turn will require increasingly distributed ML optimization. It is expected that the hierarchical multi-fidelity ML approach will significantly improve predictive accuracy for a much wider range of configurations than training solely on high-fidelity data at a limited range of conditions.

Accurate prediction of turbulent flow separation, including wake formation, drag, and aerodynamic stall, is essential to the design of many engineering applications. The high Reynolds numbers at operating conditions necessitate reduced-order PDEs such as the large-eddy simulation (LES) equations or the Reynolds-averaged Navier–Stokes (RANS) equations. However, these reduced-order PDEs require models for unclosed terms, and coarse mesh resolutions further degrade predictive accuracy. Our approach optimizes over the reduced-order equations (LES or RANS) using a form of PDE-constrained optimization with targets obtained from both experimental and computational data. In our published research, the resulting predictive accuracy is comparable to fully resolved Direct Numerical Simulation (DNS) but has computational costs comparable to very-coarse-grained LES. The data assimilation procedure itself is system-agnostic and has been successful for a variety of laminar, turbulent, hypersonic, and reacting flows. Our ultimate goal is to address real-world problems such as modeling an airplane wing at flight Reynolds numbers.

In preparation for an INCITE allocation, we have optimized our PDE-constrained ML platform, *PyFlowCL*, for system-scale data assimilation on *Summit*. *PyFlowCL* is a high-performance, MPI-distributed platform for embedded DL in the Navier–Stokes equations. It scales with at least 80 % parallel efficiency for predictions and at least 97 % parallel efficiency for optimization on up to 16,384 *Summit* GPUs (2700 nodes). *PyFlowCL* leverages deep integration with the *PyTorch* library for high-performance computing, GPU acceleration, ML optimization, and differentiable programming. As a flow solver, it is unique in its foundation upon *PyTorch* and its concomitant ML capabilities and flexibility across system architectures. As a distributed DA platform, it is unique in its solver-in-the-loop optimization capabilities and its automatic, efficient construction of the necessary adjoint PDEs using differentiable programming. Its system architecture flexibility will enable a straightforward transition from *Summit* to *Frontier*.

PROJECT NARRATIVE

1 SIGNIFICANCE OF RESEARCH

Accurate prediction of unsteady turbulent flows is crucial for the design of engineering devices of national interest. However, even on leadership-class supercomputers such as *Summit* and *Frontier*, fully resolved simulations of the Navier–Stokes equations at typical flight-vehicle, turbomachinery, wind turbine blade Reynolds numbers are intractable. Engineering design thus design relies on approximations to the fully resolved equations, including the coarse-grained Large-Eddy Simulation (LES) equations and the Reynolds-averaged Navier–Stokes (RANS) equations.

LES and RANS enable computationally tractable simulations by substantially reducing the spatial resolution, temporal resolution, and problem dimensionality. However, this comes at a cost: unclosed terms, representing the neglected turbulence scales, are introduced into the equations—modeling these terms has been a grand challenge for over 50 years. The shortcomings of RANS models for free-shear and separated flows (e.g., jets, mixing layers, wakes, and stalling airfoils) are well known [1–3]; RANS models "tuned" for one configuration do not generalize to different geometries. For LES, the Smagorinsky model [4] and dynamic variants [5–7] are usually successful for free-shear flows, but significant challenges remain in resolving or modeling near-wall flows [1, 8], and flow-separation predictions can be qualitatively incorrect for high angles of attack [9, 10]. Currently, the application of LES to industrial devices is severely limited by prohibitive mesh sizes required for accuracy in near-wall and wake regions [11]. Accurate modeling of these regions is especially critical for aerodynamic applications such as drag reduction and stall prediction. Thus major challenges remain, and current models cannot be trusted without experimental validation.

Deep learning (DL) methods have been shown mathematically and in practice to accurately model high-dimensional, nonlinear functions. This makes DL a promising choice to address nonlinear closure challenges in fluid mechanics. The derivatives of the RANS and LES quantities form a high-dimensional space from which closure models may be inferred. Recent research, including by the PIs, has highlighted the immense potential of DL for RANS and LES closures [12–18].

The proposed INCITE allocation will enable ***extreme-scale data assimilation (DA)*** for LES of turbulent flows and will substantially advance the state-of-the-art for predictive accuracy, computational cost, and model-based design of real-world aerodynamic applications. We have developed DL-based DA methods [15–17, 19] that will be extended and generalized over the course of the proposed allocation. These methods leverage both high-fidelity numerical data and sparse experimental data to enable coarse-grained simulations with high predictive accuracy and extremely low cost. Our goal is to enable predictive LES for challenging flow configurations, which will result in faster and higher-risk design cycles for industrial applications. The proposed allocation encompasses:

- Production of extreme-scale Direct Numerical Simulation (DNS) data for DL training and evaluation.
- Highly distributed DA while solving the LES equations, which will require thousands of interconnected, GPU-accelerated nodes.
- Evaluation of the learned models across a wide range of aerodynamics configurations and interpretation of the learned physics.
- Development and evaluation of multi-fidelity optimization methods to accelerate model development using fine-grained, coarse-grained, and experimental data.

A key innovation of our approach is its optimization over the entire PDEs; that is, it ensures that the modeled physics satisfy the PDEs. When a DL closure model (e.g., a neural network) is used, the solution of the PDEs becomes a function of the model parameters, which are unknown before optimization. DA methods must therefore address the strongly coupled relationship between the DL closure and the PDE solution [14, 16]—i.e., the neural network affects the PDE solution, and the derivatives of the PDE solution

are inputs to the neural network. In *virtually all current DA methods*, the closures are estimated offline and *completely decoupled from the governing equations*. In contrast, our DA approach uses adjoint PDE methods to optimize over the entire DL–PDE solution, leading to significant improvements in predictive accuracy. The importance of this difference is explained in more detail in Section 1.3. Thus, the planned approach is a form of **physics-constrained data assimilation**.

The planned research will additionally address a fundamental question in turbulence modeling: can powerful, nonlinear parametric models (such as those available from DL) uncover previously unknown relationships between filtered velocities and the unclosed terms? If so, then this knowledge should be harnessed to develop new classes of models. Traditional models are typically low-dimensional, which does not permit accurate physical representations. The success of DL in fields such as image recognition, computer vision, and natural language processing is due to its ability to learn and represent these types of high-dimensional nonlinear relationships. The proposed INCITE allocation will enable addressing the ability of PDE-constrained DL models to capture such high-dimensional relationships.

1.1 Potential Scientific Impact

PDEs are the fundamental building blocks of models in many scientific, engineering, and applied mathematics fields. The scientific community is also widely interested in leveraging large datasets and machine learning techniques. The proposed allocation will enable the development and evaluation of methods with broad applicability across a wide range of disciplines. An INCITE allocation will enable further development of the PIs' distributed-DA software platform and testing at the exascale. In turn, this will provide foundations and best practices for researchers in extreme-scale DA in other fields.

The specific focus area of our proposal is highly relevant to national defense and commercial interests. Accurately predicting turbulent flows with tractable computational cost will enable rapid advances in many commercial and national-defense applications, including aircraft and rocket propulsion, scramjet engines, power generation, wind-turbine design, high-energy weapons, and UAVs.

Currently, RANS is most common in industry and government for high-Reynolds-number applications [11]. LES at larger filter sizes, combined with DL closures and GPU acceleration, has the potential to achieve significantly higher fidelity than RANS at a modest difference in computational cost. This will enable higher accuracy than is currently possible for a wide range of applications. Our proposed DL closures are equally capable of training on multi-fidelity numerical data sources, statistical data, and experimental data. **Figure 1** illustrates the physics-knowledge transferability achievable by our DL closures.

The proposed research is highly interdisciplinary and will forge new connections between the HPC, CFD, and DL/DA fields. Mathematical and numerical techniques will be developed and proven across multiple disciplines, and the multi-GPU software and the DNS datasets to be developed will be made publicly available. The PIs hope that this will lead to this project's resulting software and computational methods being widely adopted across industrial, governmental, and academic research.

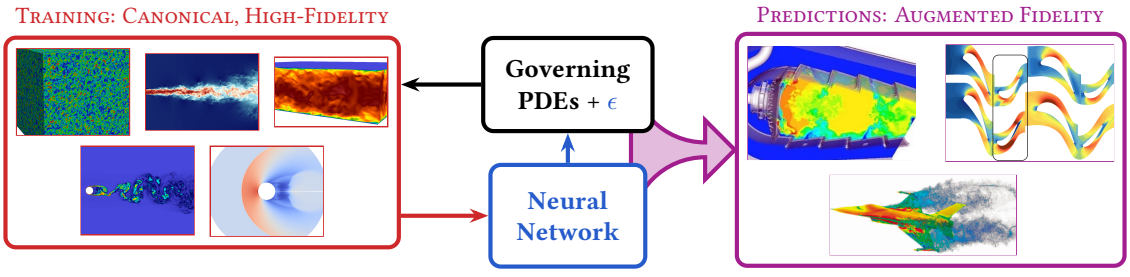


Figure 1: Schematic overview of the data assimilation method, illustrating simultaneous training on canonical turbulence cases, physics assimilation into a PDE-embedded neural network, and augmented-fidelity predictions on applications of interest.

1.2 Advances Enabled by INCITE

PDE-constrained optimization is a computationally challenging task, and the planned research will address this challenge at the exascale: we will develop DL closure models by jointly optimizing over thousands of multi-GPU simulations for hundreds of geometric and parameter variations. This will produce robust DL closures for accurate simulations across a wide range of physical regimes and geometries. The optimization task will require a leadership-class parallel filesystem and interconnect to feed the necessary amount of data and to synchronize parameter updates at the speeds required by the simulations.

The PIs have developed a scalable, multi-GPU software suite for DA-based model development and testing. A 2021 OLCF Director's Discretionary Allocation enabled developing and testing the code on *Summit*'s hybrid CPU–GPU architecture, demonstrating its scalability and readiness for the extreme-scale simulations and DA runs to be performed during an INCITE allocation, and generating initial datasets. **The planned simulations and extreme-scale DA will run entirely on *Summit*'s and *Frontier*'s GPUs.**

The PIs have demonstrated that their DA-based modeling approach outperforms current state-of-the-art LES models for several simple configurations in aerodynamics (Sec. 1.4). The next step is to increase the scale of the model training to thousands of simultaneous flow simulations, which will lead to a robust model with accuracy across a wide range of flow conditions. The leadership-class systems available through INCITE are uniquely positioned to address this grand challenge. As the realism of the training configurations increases, the required resources for DA will also increase. An INCITE allocation will enable the development and evaluation of extreme-scaled DA methods for a wide range of applications.

In addition to model training, the high-fidelity DNS data needed to train and evaluate DL models will require large-scale resources. HPC flow simulations require large amounts of data to be shared between processes over the interconnect; on previous hybrid CPU–GPU machines, the requirement to copy GPU buffers to CPUs for communication resulted in severe scaling bottlenecks. *Summit*'s support for GPU Direct communication (and *Frontier*'s Infinity Fabric) addresses this limitation unlike previous machines. This makes INCITE uniquely positioned to enable integration of extreme-scale DA into HPC flow simulations.

PI MacArt has extensive user experience on OLCF, NERSC, and NNSA computing resources. The PIs jointly have extensive user experience on *Summit* and several other current and former leadership-class systems including NCSA *Blue Waters*, TACC *Stampede*, and TACC *Frontera*. **The PIs have not previously received an INCITE award.**

1.3 Background and Motivation

To date, DL closure models for RANS and LES have relied primarily upon *a priori* optimization, in which the neural network parameters are fitted offline, without solving the governing equations [18, 20]. Standard supervised-learning methods train the DL closure model to predict the DNS-evaluated Reynolds stress/subgrid stress using the DNS-evaluated time-averaged/filtered velocities as input variables.

Specifically, let h_θ be the closure model with parameters θ which must be calibrated to data. The parameters θ are estimated by minimizing the simple objective function

$$J(\theta) = \left\| h_\theta(u^{\text{Filtered DNS}}) - \text{Unclosed Terms}(u^{\text{Filtered DNS}}) \right\|_2^2. \tag{1}$$

The objective function (1) is completely decoupled from the PDEs. Once trained, the model is substituted into the LES/RANS equations for simulation. The *a priori* optimization approach is computationally tractable but is suboptimal for DL closure model training: for example, the neural network is trained with DNS variables as inputs but will receive RANS/LES variables during predictions. From a mathematical standpoint, the *a priori* training method **interchanges** the optimization with a nonlinear function (the PDE). However, optimization will not commute with nonlinear functions such as the Navier–Stokes equations.

These shortcomings are illustrated in **Fig. 2**, where an *a priori*-trained DL model that accurately predicts the DNS-evaluated subgrid-scale (SGS) stress in decaying isotropic turbulence is unstable in LES [16].

Our training approach for embedded deep learning models in PDEs optimizes instead over the entire PDE model, which contains the embedded neural network. This optimization is nontrivial, but it is mathematically guaranteed to improve the accuracy of the PDE model based on training data, whereas *a priori* optimization method is not. **Figure 2** displays the results of our adjoint-based method in blue.

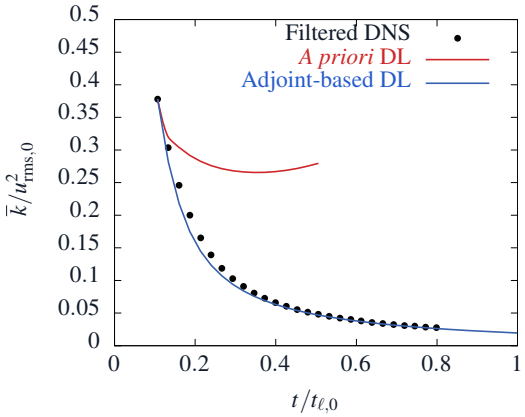


Figure 2: Comparison of an *a priori*-trained DL model to an adjoint-based DL model applied to LES of decaying isotropic turbulence; $\bar{k}$ is the TKE. The *a priori* model minimizes the mismatch between the model and the unclosed SGS term.

Adjoint optimization methods for RANS closure models have been developed in parallel [13,14]. A growing body of "scientific machine learning" literature covers DL models and methods for CFD and computational physics; a representative (although certainly not an exhaustive) list of recent articles is [21–26]. Additionally, in parallel to DL closures for RANS/LES, other classes of physics-informed DL methods have been developed for PDEs including physics-informed neural networks (PINNs) [24] and deep operator networks ("DeepONet") [25, 26]. These directly approximate a PDE solution (PINNs) or the PDE operator (deep operator networks) using neural networks. The PINN training method minimizes the PDE residual, which constrains the model to satisfy physical constraints, but does not include the same strict PDE constraints during predictions as our proposed method. PINNs have achieved substantial success across a variety of scientific applications but face challenges for unsteady problems. Our DL-LES and DL hybrid RANS–LES models, once trained, could be combined with PINN methods for predictive modeling, which would be a interesting research direction.

1.4 Preliminary Results

This research will build upon our prior work, which has developed DL closure models for incompressible RANS and LES [15–17, 19]. **Figures 3** and **4** display results for our previous deep learning LES models for several canonical turbulent flows. The deep learning LES models outperform the dynamic Smagorinsky model [5, 6], which is the current state-of-the-art LES closure model for these flows.

Directly training a deep neural network to learn complex physics from scratch is challenging due to the limited number of datasets—DNS and experimental data are costly to generate. Limited data leads to overfitting and poor out-of-sample performance. (For example, the trained model might not satisfy conservation laws.) Our deep learning approach embeds a neural network within the LES equations, which represent the largest turbulence scales and enforce conservation laws. By leveraging the governing equations, our PDE-constrained DL models *have been able to learn successfully from limited data* [16, 17, 19].

1.5 Readiness for INCITE Computational Grant

Our DL-LES equations and adjoint optimization approach have been implemented and evaluated for canonical turbulent flows—see **Figures 3** and **4**—where they outperform current state-of-the-art LES models such as the dynamic Smagorinsky model. Using our OLCF Director's Discretionary Allocation, we have tested and improved our code's parallel performance on *Summit* (see Section 3.3). Our methods and software platform (*PyFlowCL*) are ready for the scale of the proposed INCITE computations: we are prepared to commence high-fidelity DNS production and machine-scale distributed model training on Day 1 of the proposed INCITE allocation. We anticipate that the transition from *Summit* to *Frontier* will be straightforward, as *PyFlowCL* is built entirely upon the system MPI implementation and the *PyTorch* library, which has been optimized for both NVIDIA and AMD GPUs.

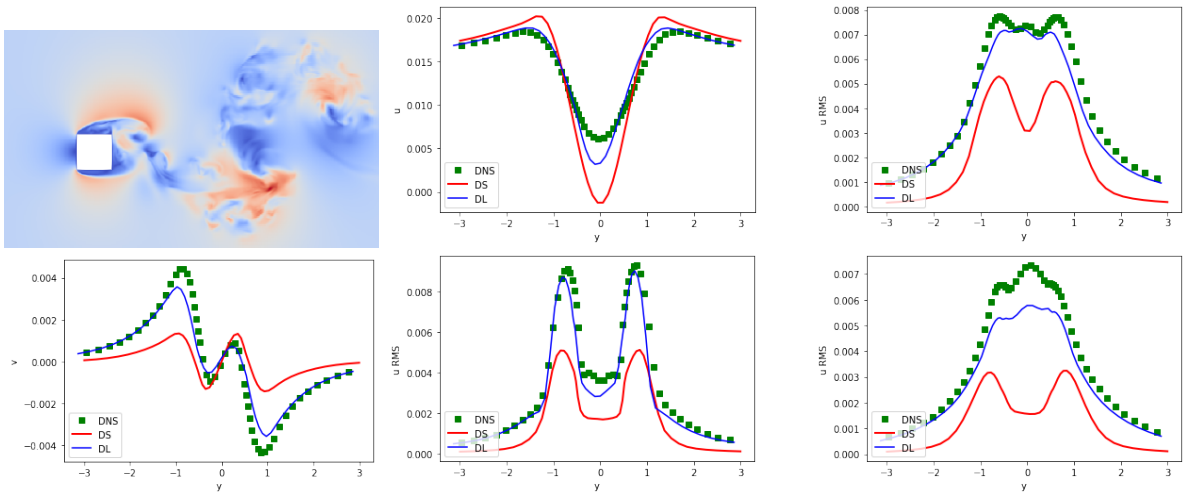


Figure 3: Comparison of deep learning (DL) and dynamic Smagorinsky (DS) LES models to fully resolved direct numerical simulation (DNS) data for a turbulent square-cylinder wake at Re = 1000. All results are out-of-sample. Results from the PIs' 2021 OLCF Director's Discretionary Allocation (publication in preparation).

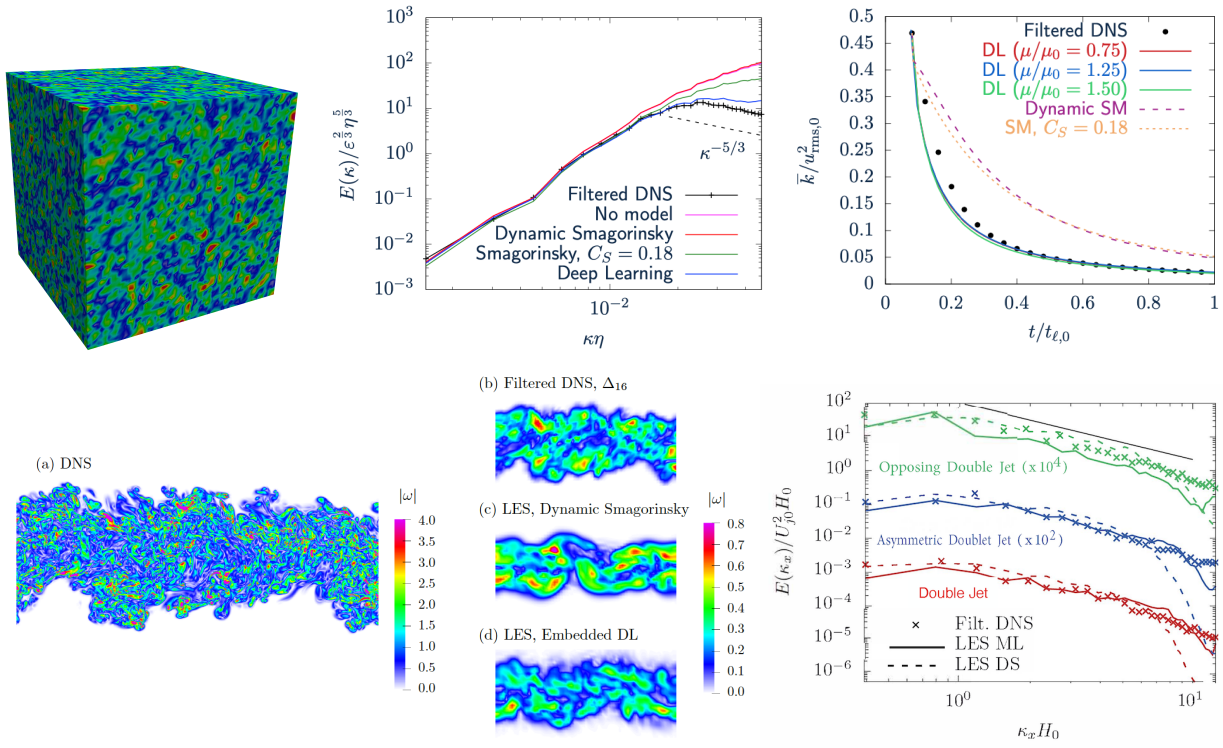


Figure 4: Comparison of out-of-sample deep learning (DL) and dynamic Smagorinsky (DS) models to fully resolved direct numerical simulation (DNS) data for decaying isotropic turbulence (top row) and several turbulent-jet configurations (bottom row). (Reproduced from the PIs' previous work [16, 17].)

2 RESEARCH OBJECTIVES AND MILESTONES

The research plan is divided into two main objectives. The timeline for these objectives and our planned computational usage (Sec. 3) is summarized in the Milestones Table. *We highlight that Tasks 1.1 and 1.2 have already been completed in advance of the proposed allocation.* The remaining Tasks will require the proposed INCITE computational resources.

Objective 1: Develop a DL closure framework for the compressible Navier–Stokes equations

- **Task 1.1:** Derive DL closure models for compressible LES
- **Task 1.2:** Develop high-performance, online adjoint-based data-assimilation methods for LES
- **Task 1.3:** Develop a multi-fidelity training framework to augment limited data

Objective 2: Implement, train, and evaluate the learned closure models vs. standard models

- **Task 2.1:** Large-scale DNS data production for a range of separated-flow parameters
- **Task 2.2:** Massively distributed model training and testing for out-of-sample flow conditions
- **Task 2.3:** Analysis and comparison of the learned energy dynamics to the "truth" data

2.1 Objective 1: DL closure of the compressible Navier–Stokes equations

Our formulation is based upon the compressible Navier–Stokes equations,

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} &= 0, \\
\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} &= -\frac{1}{\gamma \mathrm{Ma}^2}\frac{\partial \rho T}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial \sigma_{ij}}{\partial x_j}, \\
\frac{\partial \rho T}{\partial t} + \frac{\partial \rho u_j T}{\partial x_j} &= -(\gamma-1)T\frac{\partial u_j}{\partial x_j} + \frac{\gamma}{\mathrm{Re}\,\mathrm{Pr}}\frac{\partial}{\partial x_j}\left(\kappa\frac{\partial T}{\partial x_j}\right) + \frac{\gamma(\gamma-1)\mathrm{Ma}^2}{\mathrm{Re}}\Phi,
\end{aligned} \tag{2}$$

where $x \in \Omega$ is the simulation domain, ρ is the density, u_i is the velocity vector ($i = 1,2,3$), T is the temperature, $\gamma = c_P/c_v$ is the ratio of specific heats, and Re, Ma, and Pr are the scaling Reynolds, Mach, and Prandtl numbers, respectively; precise definitions of these depend on the characteristic scales of the flow. The shear stress and heat flux are $\sigma = \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^\top - \frac{2}{3}(\nabla\cdot\mathbf{u})\mathbf{I}$ and $\Phi = \sum_{ij}\sigma_{ij}\partial_{x_i}u_j$, respectively, where $\mathbf{u} = (u_1,u_2,u_3)$. We model the viscosity and thermal conductivity $\mu = \mu(T)$ and $\kappa = \kappa(T)$ using a standard power-law model. Appropriate boundary conditions for $x \in \partial\Omega$ complete the PDE system (2).

DNS of the compressible Navier–Stokes equations (2) is computationally prohibitive due to the resolution requirements of flight-relevant Reynolds numbers. LES significantly reduces this computational cost by resolving only the largest turbulence scales. The derivation of the LES equations involves a spatial filtering operation applied to the Navier–Stokes equations, which leads to terms that depend on unrepresented scales. Thus, the filtered equations are said to be unclosed. Subgrid-scale closure models must be introduced to represent the effect of unrepresented scales in terms of the resolved scales.

Define the unweighted and density-weighted filtering operations

$$\overline{\phi} \equiv \int_\Omega G(\mathbf{r},\mathbf{x})\phi(t,\mathbf{x}-\mathbf{r})\,\mathrm{d}\mathbf{r} \qquad \text{and} \qquad \widetilde{\phi} \equiv \frac{1}{\overline{\rho}}\int_\Omega G(\mathbf{r},\mathbf{x})\rho(t,\mathbf{x}-\mathbf{r})\phi(t,\mathbf{x}-\mathbf{r})\,\mathrm{d}\mathbf{r}, \tag{3}$$

respectively, where $G(\mathbf{r},\mathbf{x})$ is an appropriately normalized LES filter kernel. (Standard filter kernels include

truncated Gaussian and spatially sharp kernels.) Applying (3) to (2) yields the compressible LES equations

$$\begin{aligned}
&\frac{\partial \overline{\rho}}{\partial t}+\frac{\partial \overline{\rho}\widetilde{u}_j}{\partial x_j}=0,\\
&\frac{\partial \overline{\rho}\widetilde{u}_i}{\partial t}+\frac{\partial \overline{\rho}\widetilde{u}_i\widetilde{u}_j}{\partial x_j}=-\frac{1}{\gamma \mathrm{Ma}^2}\frac{\partial \overline{\rho}\widetilde{T}}{\partial x_i}+\frac{1}{\mathrm{Re}}\frac{\partial \overline{\sigma}_{ij}}{\partial x_j}-\frac{\partial \tau_{ij}^{\mathrm{SGS}}}{\partial x_j}+\varepsilon_{u_i},\\
&\frac{\partial \overline{\rho}\widetilde{T}}{\partial t}+\frac{\partial \overline{\rho}\widetilde{u}_j\widetilde{T}}{\partial x_j}=-(\gamma-1)\overline{T}\frac{\partial \overline{u}_j}{\partial x_j}+\frac{\gamma}{\mathrm{Re}\,\mathrm{Pr}}\frac{\partial}{\partial x_j}\left(\kappa(\overline{T})\frac{\partial \overline{T}}{\partial x_j}\right)+\frac{\gamma(\gamma-1)\mathrm{Ma}^2}{\mathrm{Re}}\overline{\Phi}-\frac{\partial f_j^{\mathrm{SGS}}}{\partial x_j}+\varepsilon_T,
\end{aligned}\tag{4}$$

where $\overline{\sigma}=\mu(\overline{T})(\nabla\overline{\mathbf{u}}+\nabla\overline{\mathbf{u}}^\top-\frac{2}{3}(\nabla\cdot\overline{\mathbf{u}})\mathbf{I}$ and $\overline{\Phi}=\sum_{ij}\sigma_{ij}\partial_{x_i}\overline{u}_j$. The filtering operation produces several unclosed terms, highlighted in red, due to the nonlinearities of (2) and several other neglected terms. Historically, the primary modeling challenge has comprised the SGS stress tensor and heat-flux vector,

$$\tau_{ij}^{\mathrm{SGS}}\equiv\overline{\rho\widetilde{u_iu_j}}-\overline{\rho}\widetilde{u}_i\widetilde{u}_j \qquad \text{and} \qquad f_j^{\mathrm{SGS}}\equiv\overline{\rho\widetilde{u_jT}}-\overline{\rho}\widetilde{u}_j\widetilde{T},$$

where the first terms on the right-hand sides are unclosed if only the filtered variables are known. Accurate modeling of these unclosed terms is crucial for accuracy of LES predictions [27, 28].

The LES equations (4) contain additional unclosed terms, denoted ε_{u_i} and ε_T, that arise due to (a) the noncommutativity of the filtering and discrete differentiation operations and (b) the use of nonlinear constitutive and transport models. These terms are almost universally neglected in LES in practice [29–31], even though they can become significant (comparable to τ_{ij}^{SGS} and f_j^{SGS}) in flows with significant compressibility, variation in transport coefficients, and/or nonuniform computational grids [28, 29]. As will be demonstrated, the proposed deep learning models account for *all* unclosed terms, including these additional terms.

2.1.1 *Task 1.1: Deep learning closure models for compressible LES*

We will model the unclosed terms with a deep neural network (DNN) $\mathbf{h}(\rho,T,\mathbf{u}_x,\mathbf{u}_{xx};\theta)$ with parameters θ to be calibrated to data. This leads to a DL-LES model with closure terms highlighted in blue:

$$\begin{aligned}
&\frac{\partial \overline{\rho}}{\partial t}+\frac{\partial \overline{\rho}\widetilde{u}_j}{\partial x_j}=0,\\
&\frac{\partial \overline{\rho}\widetilde{u}_i}{\partial t}+\frac{\partial \overline{\rho}\widetilde{u}_i\widetilde{u}_j}{\partial x_j}=-\frac{1}{\gamma \mathrm{Ma}^2}\frac{\partial \overline{\rho}\widetilde{T}}{\partial x_j}+\frac{1}{\mathrm{Re}}\frac{\partial \overline{\sigma}_{ij}}{\partial x_j}+\frac{\partial h_{ij}}{\partial x_j}(\overline{\rho},\widetilde{T},\widetilde{\mathbf{u}}_x,\widetilde{\mathbf{u}}_{xx};\theta),\\
&\frac{\partial (\overline{\rho}\widetilde{T})}{\partial t}+\frac{\partial \overline{\rho}\widetilde{u}_j\widetilde{t}}{\partial x_j}=-(\gamma-1)\overline{T}\frac{\partial \overline{u}_j}{\partial x_j}+\frac{\gamma}{\mathrm{Re}\,\mathrm{Pr}}\frac{\partial}{\partial x_j}\left(\kappa(\overline{T})\frac{\partial \overline{T}}{\partial x_j}\right)+\frac{\gamma(\gamma-1)\mathrm{Ma}^2}{\mathrm{Re}}\overline{\Phi}+\frac{\partial h_{4j}}{\partial x_j}(\overline{\rho},\widetilde{T},\widetilde{\mathbf{u}}_x,\widetilde{\mathbf{u}}_{xx};\theta).
\end{aligned}\tag{5}$$

The proposed DL-LES approach is unique in that it will select the parameters θ such that the LES solution $\mathbf{u}(t,x)$ is as close as possible to trustworthy high-fidelity data $\mathbf{V}(t,x)$, which will be available from DNS or experimental observations. Therefore, we will minimize the objective function

$$J(\theta)=\int_0^T\int_\Omega\|\widetilde{\mathbf{u}}(t,x;\theta)-\mathbf{V}(t,x)\|_2^2\,dxdt,\tag{6}$$

where $\mathbf{u}(t,x;\theta)$ is the solution of (5) for parameters θ, and $\|\cdot\|_2$ is the Euclidean norm. The advantage of this optimization approach is the ability to define the objective function (6) *arbitrarily* to focus on quantities of interest (QOIs) for specific flows. In contrast, *a priori* optimization (Section 1.3) can *only* target the unclosed terms (e.g., τ_{ij}^{SGS} and f_j^{SGS}); in general, these may be challenging to measure with sufficient accuracy, and even exactly representing them does not guarantee accurate QOI predictions when implemented in LES.

2.1.2 *Task 1.2: High-performance, online adjoint-based data-assimilation methods*

Optimizing the model parameters θ requires minimizing $J(\theta)$; standard gradient descent-type algorithms for this require evaluating the gradient $\nabla_\theta J(\theta)$. For the proposed loss function (6), doing so would be highly challenging, since it would involve differentiating a function defined by a PDE, which is in turn a function of the embedded DNN. In summary, this functional dependence is:

$$\text{Parameters } \theta \longrightarrow \text{Neural network } \mathbf{h}_\theta \longrightarrow \text{PDE (5) for } (\overline{\rho}, \widetilde{T}, \widetilde{\mathbf{u}}_i) \longrightarrow \text{Objective function } J(\theta).$$

A naïve approach that directly evaluates $\nabla_\theta J(\theta)$ would require solving a system of $d \times 5$ PDEs, where the number of PDEs is proportional to the number DNN parameters d. This is computationally intractable for typical DNNs, for which $d \sim O(10^5)$ or $O(10^6)$.

Instead, our approach solves adjoint PDEs for efficient evaluation of $\nabla_\theta J(\theta)$. The number of adjoint PDEs is equal to the number of equations in the LES model *no matter the number of DNN parameters*. The PIs have successfully implemented this optimization approach for DL-LES and DL-RANS models of incompressible turbulence [15–17] and nonequilibrium hypersonic flows [19].

Solving the adjoint PDEs for the adjoint variables $(\hat{u}_i, \hat{\rho}, \hat{T})$ enables us to calculate the gradient $\nabla_\theta J(\theta)$ via

$$\nabla_\theta J(\theta) = \int_0^T \int_\Omega \left(\sum_{i=1}^{3} \hat{u}_i \frac{\partial^2 h_{ij}}{\partial x_j \partial \theta} + \hat{T} \frac{\partial^2 h_{4j}}{\partial x_j \partial \theta} \right) dx dt. \tag{7}$$

A gradient descent algorithm for minimizing $J(\theta)$ would be

$$\theta^{(k+1)} = \theta^{(k)} - \alpha^{(k)} \nabla_\theta J(\theta^{(k)}), \tag{8}$$

where $\theta^{(k)}$ is the parameter estimate at the k^{th} optimization iteration and $\alpha^{(k)}$ is the learning rate. At each iteration, $\nabla_\theta J(\theta^{(k)})$ is evaluated by re-solving the forward and adjoint PDEs.

To accelerate training, we use a scalable multi-GPU stochastic gradient descent method where each machine, at each training iteration:

- Randomly select a short time interval $[t, t+\tau]$ and its corresponding DNS data $\mathbf{V}(t,x)$.
- Solve (5) on $[t, t+\tau]$ (with an initial condition from the DNS) and calculate the corresponding loss function $J(\theta)$.
- Solve the adjoint PDEs on $[t, t+\tau]$.
- Calculate the gradient (8) on each machine.
- Average the gradients across all machines (distributed gradient descent) and update the parameters θ.

We have implemented this algorithm for training deep learning LES models for incompressible flows [16, 17]. The proposed project would be the first to apply it to turbulence modeling for compressible flows and would be the first to harness exascale computing resources for massively parallelized, distributed training. In our experience, distributed training across many different physical configurations and parameters (Sec. 2.2.2) is crucial for model generalization; thus we expect high-performance optimization codes to be crucial for the next generation of data-assimilation tasks. The proposed INCITE allocation will be uniquely valuable in enabling these future data-assimilation tools and methods.

2.1.3 *Task 1.3: Develop a multi-fidelity training framework*

Two key factors will affect the accuracy of trained DL models: (1) the accuracy of the training data and (2) the extent of the training datasets. More training data (e.g., high-fidelity data for a greater number of geometries and/or physical regimes) would enable a more accurate model and/or one that will potentially

generalize better. However, DNS for a large number of training cases is computationally prohibitive and therefore limits the extent of the training data. Furthermore, as a PDE model becomes more reduced-order (i.e., resolving less of the physics), more data will be required to avoid overfitting. For example, RANS closures must model a wider range of spatio-temporal scales, therefore more data are required to train a DL-RANS model than would be required for a DL-LES model.

A trade-off therefore exists between the fidelity of the training data and the amount of data. Generating more data, at a slightly lower fidelity than DNS, is likely result in superior training compared to a few limited DNS datasets. In particular, if a very large number of accurate LES simulations (e.g., using the DL closure model) is generated, then a hybrid DL RANS–LES model could be trained targeting this large universe of configurations. We expect that this approach would to lead to substantial improvements in the RANS–LES model's accuracy and ability to generalize to new configurations (outside of the training set).

We will develop a new, ***multi-fidelity DL approach*** based on a hierarchy of different-fidelity DL-based models for generating data:

(i) High-fidelity DNS data will be generated for a limited number of regimes and geometries.

(ii) High-accuracy DL-LES models will be trained on the DNS data for canonical flows.

(iii) Medium-fidelity data will be generated using the high-accuracy DL-LES data [Step (ii)] for a much larger range of flow regimes and geometries.

(iv) Low-cost, coarse-grid LES and hybrid RANS–LES models will be trained on the Step (iii) data.

The DL-LES model will be much lower cost than DNS, but our goal is that, with the aid of the deep learning closure model, it will provide an accurate approximation of the DNS. This will be leveraged to generate a much larger number of training datasets spanning far more geometries and flow regimes. It is expected that this hierarchical multi-fidelity machine learning approach will significantly improve the accuracy of coarse-grid DL-LES and hybrid RANS–LES models for a much wider range of out-of-sample configurations.

2.2 Objective 2: Implement, train, and evaluate the learned closure models vs. standard models

2.2.1 Task 2.1: Large-scale DNS data production for a range of separated-flow parameters

A database for a series of aerodynamics configurations will be generated for model training and testing:

(i) Turbulent flow over rectangular cylinders at varying aspect ratios (AR) and Reynolds numbers,

(ii) Turbulent flow over circular cylinders at different Reynolds numbers,

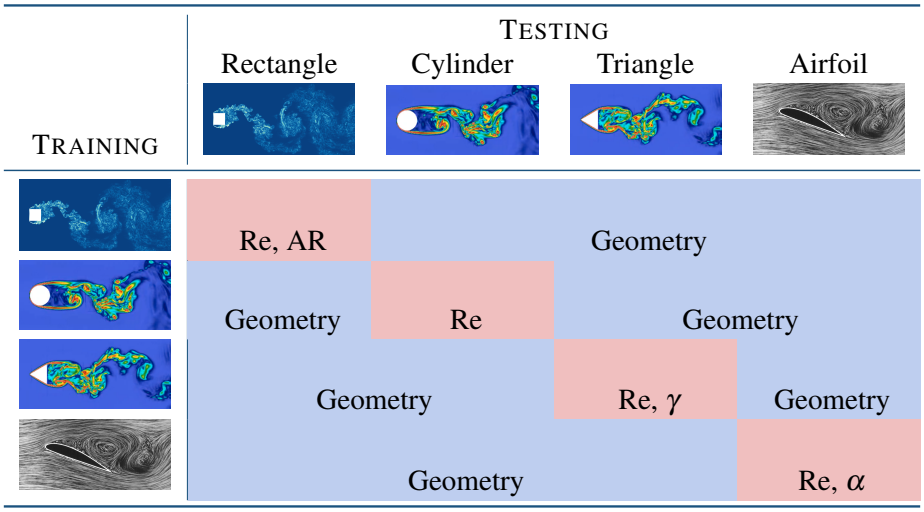
(iii) Turbulent flow over triangle-shaped flameholders for a range of different leading-edge angles (γ) and Reynolds numbers, and

(iv) Turbulent flow over airfoils, including separated turbulent flow, parameterized on angle of attack (α) and Reynolds number.

Statistically converged DNS datasets for each configuration will be generated using *PyFlowCL*, which the PIs have developed specifically for high-performance DL model development. It solves the compressible Navier–Stokes equations on curvilinear grids using fourth-order finite difference methods [32]. Stability for supersonic flows is achieved using fourth-order shock capturing [33, 34] and sixth-order implicit filtering [32]. Parallel scalability is achieved using MPI domain decomposition with full GPU offloading. The code leverages *PyTorch* [35] for high-performance machine learning and has been developed specifically for hybrid CPU–GPU architectures (see Section 3.3).

The PIs have already generated a library of DNS datasets that the proposed study will leverage: rectangular cylinders [36–39] for aspect ratios $\mathrm{AR} = (0.5, 1, 2, 4)$ and Reynolds numbers $\mathrm{Re} = (1000, 2000, 4000)$, circular cylinders at $\mathrm{Re} = (1000, 4000)$, and triangular geometries at $\mathrm{Re} = (1000, 4000)$. This proposed Task

Table 1: Proposed training/testing matrix for DL-LES model development. Rows indicate the base training configurations; columns indicate testing configurations. Cell colors indicate the properties changed for out-of-sample tests: parameter variations (red, parameters as labeled) and geometries (blue).



will augment the DNS library with additional cases: flow over triangular blockages at fixed Re and varying leading-edge angles γ, flow over semi-infinite airfoils for fixed profile geometry and varying angle of attack α, and higher-Reynolds-number cases for each geometry. The proposed resources required for these simulations are described in Sec. 3.1.

2.2.2 Task 2.2: Massively distributed model training and testing for out-of-sample flow conditions

The DNS datasets will be divided into two subsets: training and testing. Models will be trained using adjoint data-assimilation methods (Section 2.1.2). Once trained, models will be simulated out-of-sample for configurations in the testing dataset. Out-of-sample tests are essential in order to evaluate the models' accuracy for interpolation (e.g., predicting flows at an intermediate Reynolds number from those targeted during training), extrapolation (e.g., predicting flow at higher Reynolds number), and generalization to different geometries (e.g., using a model trained on a circular cylinder to predict flow over an airfoil).

Three types of out-of-sample tests will be conducted. Each will compare the accuracy of the learned models to standard LES turbulence models including the widely used dynamic Smagorinsky model [4, 5], the nonlinear gradient (Clark) model [40], and dynamic variants of the Clark model [41].

1. Models will be simulated out-of-sample for the same configuration but with different physical and geometric parameters. For example, one model will be trained on a rectangular cylinder and tested out-of-sample for rectangular cylinders at different Reynolds numbers and aspect ratios. **Table 1** lists the proposed parameter variations for each geometric configuration.

2. Models will be simulated out-of-sample for different geometric configuration. For example, one model will be trained on a rectangular cylinder and tested out-of-sample on a circular cylinder, a triangular cylinder, and an airfoil at the same nominal Reynolds number. The proposed out-of-sample geometries are indicated in **Table 1**.

3. Models will be trained on multiple configurations (e.g., on rectangular cylinders at different AR and Re and circular cylinders at different Re) and simulated out-of-sample on airfoils at different Re.

2.2.3 Task 2.3: Analysis and comparison of the learned energy dynamics to the "truth" data

In addition to comparing the learned closures to existing models, we will thoroughly quantify the kinetic energy dynamics recovered by the proposed models in order to better understand the reasons for their performance. A transport equation for the LES-resolved kinetic energy $k = \widetilde{u}_i\widetilde{u}_k$ may be obtained from the LES momentum equation [42],

$$\frac{\partial k}{\partial t} + \widetilde{u}_i \frac{\partial k}{\partial x_i} = \alpha_p + \alpha_v + \alpha_{\mathrm{SGS}}, \tag{9}$$

where the terms

$$\overline{\rho}\alpha_p = -\widetilde{u}_i \frac{\partial \overline{p}}{\partial x_i}, \qquad \overline{\rho}\alpha_v = \widetilde{u}_i \frac{\partial \overline{\tau}_{ij}}{\partial x_j}, \qquad \text{and} \qquad \overline{\rho}\alpha_{\mathrm{SGS}} = -\widetilde{u}_i \frac{\partial \tau_{ij}^{\mathrm{SGS}}}{\partial x_j}$$

are the work done by the resolved pressure, the resolved viscous stress, and the SGS turbulent stress, respectively. We will first compute statistically converged budgets (balances) of the terms in (9) as evaluated from out-of-sample DNS data, DL-LES solutions, and the comparison LES models, which will highlight overall differences in the resolved kinetic energy dynamics produced by the various models. In particular, the influence of pressure-dilatation work is known to be significant at flight-relevant (high subsonic) Mach numbers [27, 28, 42] and so will be an initial target for ensemble-averaged comparisons.

Second, we will focus on the term α_{SGS}, which is also known as the *SGS flux*. A transport equation for the subgrid-scale kinetic energy $k_{\mathrm{SGS}} = (\widetilde{u_i u_i} - \widetilde{u}_i\widetilde{u}_k)/2$ may be obtained from (4) and (9),

$$\frac{\partial k_{\mathrm{SGS}}}{\partial t} + \widetilde{u}_i \frac{\partial k_{\mathrm{SGS}}}{\partial x_i} = \alpha_p^{\mathrm{SGS}} - \alpha_{\mathrm{SGS}} + \alpha_v^{\mathrm{SGS}} + \phi_{\mathrm{SGS}}, \tag{10}$$

in which the SGS flux appears with the opposite sign as in (9). It thus signifies the two-way transfer of kinetic energy between the resolved and subgrid scales. (The other terms appearing in the RHS of (10) are unclosed under the LES variables and so will not be considered for *a posteriori* analysis.) The ensemble-averaged α_{SGS} will be computed in order to determine the ability of each model to account for local intermittency and backscatter [27]. The joint probability density functions (PDFs) of k and k_{SGS} conditioned on the cross-flow coordinates will enable further detailed interpretation of the models' performance in different regions of the flows, for example, the near and far wakes, boundary-layer turbulence, and separated airfoil flow.

Finally, the form of the DL-LES models will be interpreted by projecting the high-dimensional closures onto low-dimensional subspaces with known functional form. The PIs have developed this procedure to interpret the form of incompressible LES turbulence models [17]; the formulation will be extended for compressible turbulence with multiple, nonlinearly coupled closure terms.

3 COMPUTATIONAL READINESS

The PIs have developed a scalable, multi-GPU CFD platform, *PyFlowCL*, specifically to train and evaluate deep learning PDE models. It is fully Python-native and leverages close integration with the *PyTorch* machine learning library [35] for model optimization and automatic differentiation. *PyFlowCL* enables high-performance, parallelized data assimilation by providing seamless integration between the *PyTorch* library, adjoint-based model training, and high-fidelity DNS and experimental data.

An INCITE allocation on *Summit* and *Frontier* will uniquely enable advances in high-performance data assimilation and distributed, solver-in-the-loop deep learning. The proposed computational campaign will utilize these resources in two distinct ways:

1. High-performance, large-scale target data production will require $O(10)$ jobs spanning hundreds to thousands of nodes each.
2. Simultaneous optimization of dozens of models over hundreds of geometric and parameter variations will require $O(1000)$ interconnected simulations of size 1–4 nodes each.

The largest of the data-assimilation tasks could utilize the entire *Summit* machine, although this would not be strictly necessary. The data-assimilation tasks will additionally require *Summit*'s (and *Frontier*'s) leadership-class filesystem and interconnect to feed the necessary amount of data and to synchronize parameter updates at the speeds required by the training simulations.

Using *PyFlowCL* for model training scales to large numbers of GPUs, which is crucial for training on massive DNS datasets due to the extremely large parameter matrices and the relatively limited memory per node. The largest planned DNS runs will require 2 TB of memory per single-time flow snapshot (Sec. 3.1), which needs to be loaded into memory for adjoint-based optimization. For model training, *PyFlowCL* achieves **97 % parallel efficiency on 600 GPUs** (100 *Summit* nodes) in weak-scaling tests. For large-scale DNS production, *PyFlowCL* achieves **80 % parallel efficiency on up to 16,384 GPUs** (2700 *Summit* nodes). Further details on code scaling are given in Section 3.3. Planned development work during the INCITE allocation (Section 3.4) will further optimize *PyFlowCL* for *Summit*'s GPU Direct capabilities and *Frontier*'s Infinity Fabric/Slingshot communications model.

3.1 Use of Resources Requested

We will perform DNS of turbulent wakes in the first phase of the proposed allocation. The nondimensional parameter of interest is the Reynolds number Re, which has a direct relationship with simulation cost ($\sim \mathrm{Re}^3$). The baseline simulations will have $\mathrm{Re} = 4,000$ (the highest of our preliminary DNS data), and comparisons will be made to higher $\mathrm{Re} = 12,000$ and $\mathrm{Re} = 20,000$. We denote these sets of simulations "4N," "12N," and "20N." The computational grids for 4N, 12N, and 20N will contain approximately 134 million, 3.6 billion, and 8.6 billion mesh cells, respectively. The planned DNS calculations and DL-LES model development will be performed **entirely using *Summit*'s GPUs**. We additionally request **10,000 node-hours on *Frontier* to optimize *PyFlowCL* for the new node architecture.**

Based on our scaling studies on *Summit* (Section 3.3), *PyFlowCL* balances runtime and parallel efficiency when the domain is decomposed at 2×10^6 points per GPU; this results in approximately 96 % of the computational work being performed on the GPUs for *Summit*'s V100s. Based on this and our measured time-per-step, we expect expect each 4N simulation to require ~ 0.011 node-hours per step on 11 nodes, each 12N simulation to require ~ 0.336 node-hours per step on 288 nodes, and each 20N simulation to require ~ 0.797 node-hours per step on 1,366 nodes.

We estimate that one flow-through time will require 5,720 steps for 4N, 8,560 steps for 12N, and 11,400 steps for 20N. From our previous experience with turbulent wakes, each simulation will require at least 50 flow-through times to obtain sufficiently converged statistics. We estimate the cost per 4N simulation to be 3,146 node-hours (286 hours on 11 nodes), the cost per 12N simulation to be 143,800 node-hours (500 hours on 288 nodes), and the cost per 20N simulation to be 454,200 node-hours (330 hours on 1,366 nodes). To sufficiently augment our existing 4N databases (Section 2.2.1) and to ensure adequate model training, we plan to develop eight additional 4N databases:

- $2\times$ $\mathrm{Re} = 4,000$ for triangle flameholders for leading-edge angles $\gamma = 60^\circ$ and 30°,
- $3\times$ $\mathrm{Re} = 4,000$ airfoils for different camber and thickness ratios,
- $3\times$ $\mathrm{Re} = 4,000$ airfoils for different angles of attack α.

The total cost for 4N is 25,200 node-hours. Due to their cost, two instances of 12N will be produced. These will be "stretch" modeling targets for 4N-trained models and will also be used for cross-trained models (both 4N and 12N):

- $1\times$ $\mathrm{Re} = 12,000$ rectangular cylinder wake ($\mathrm{AR} = 1$),
- $1\times$ $\mathrm{Re} = 12,000$ triangular flameholder wake ($\gamma = 60^\circ$),

The total cost for 12N is 287,600 node-hours. Finally, due to its extreme cost, only one instance of 20N will be produced. This will be a leadership-class simulation for distribution-quality, cross-trained models:

- $1\times$ $\mathrm{Re} = 20,000$ rectangular cylinder wake ($\mathrm{AR} = 1$).

The total cost for 20N is 454,200 node-hours. The total DNS cost is **767,000** node-hours.

Distributed data assimilation will train, test, and compare DL models for up to 64 parameter variations, corresponding to the different geometric configurations and parameter sets (AR, γ, and α). Training requires 76 seconds per iteration for $N = 64^3$ meshes, which is our target for all cases. The planned distributed training runs are:

4N: 2000 iterations on 12 nodes (76 s/iteration), 64 models — 32,400 node-hours,

12N: 2000 iterations on 120 nodes (76 s/iteration), 16 models — 81,100 node-hours,

20N: 2000 iterations on 1200 nodes (76 s/iteration), 4 models — 202,700 node-hours.

The total cost for distributed data assimilation is **316,200** node-hours. **The total project request is 1,083,200 node-hours on *Summit* and 10,000 node-hours on *Frontier*.**

We plan the following usage per quarter: Q1'23, 35 %; Q2'23, 20 %; Q3'23, 35 %, and Q4'23, 10 %. The majority of the Q1'23 computational cost will be to generate DNS datasets. In parallel, during Q1'23, model training and evaluation will begin, and further development work on *PyFlowCL* for *Frontier* will be completed. Additional DNS datasets will be generated in Q2'23. Model training and evaluation will continue in Q3'23. Model training and analysis will be completed in Q4'23.

3.1.1 Data Management and Project End

The project will require a large amount of data storage. A *PyFlowCL* HDF5 binary file contains the solution state and is used both for DNS restarts and as the starting point for model training. These files are 32 GB for 4N, 864 GB for 12N, and 2.05 TB for 20N. Each 4N and 12N DNS run will produce approximately 200 binary files; only about 50 full-resolution binary files will be saved for 20N due to its extreme size. Coarse-grained data representations will be saved more frequently to be used for model training. These files would be produced on scratch and archived to HPSS for future reuse. The required space to produce the DNS databases is estimated to be 50 TB for 4N, 345 TB for 12N, and 102 TB for 20N. The storage space needed for DNS data is therefore estimated to be **497 TB**.

During production, the DNS databases will be filtered and downsampled in order to obtain the resolved and subfilter-scale LES solution fields. These fields contain all inputs necessary in the model training step. The filtering code is hybrid-parallelized using MPI for 3D domain decomposition and file I/O and OpenMP for loop threading, has demonstrated satisfactory parallel efficiency on other HPC systems, and will require negligible additional computational time. The *a priori* filtered HDF files are approximately the same size as the unfiltered DNS files but do not need to be backed up. During filtering, the original DNS binary files will be archived from scratch to HPSS, and the filtered files will take their place on the scratch disk. The filtering step therefore requires only a small additional amount of scratch space, which we estimate to be **5 TB**. Model training will require negligible additional scratch storage. In total, we request **502 TB of scratch storage** and **497 TB of archival storage**.

Toward the project's end, the archived DNS databases will be transferred to the PIs' University storage systems using Globus for future reuse. All data remaining in scratch will be reproducible without re-running the DNS and will be removed. Downsampled versions of the DNS data, both unfiltered and filtered, and the generated models will be made publicly available using the PIs' shared Google Drive, which currently hosts approximately 25 TB of published DNS data. Due to their large file sizes, the full-resolution DNS fields will be available to other researchers from the PIs upon request.

3.2 Computational Approach

DNS data production, DL-LES model training, and model evaluation will be performed using *PyFlowCL*, the PIs' high-performance, Python-native data-assimilation platform for deep learning in the fluid mechan-

ics. The code comprises two distinct feature sets, each with high degrees of novelty.

1. **High-performance flow simulation:** *PyFlowCL* is Python-native and is based entirely upon the high-performance *PyTorch* library [35] for array operations and machine learning. This close integration enables massively distributed, efficient, GPU-accelerated flow simulations and evaluation of extremely high-dimensional models, which would be computationally intractable using CPUs alone. *PyTorch* supports and is highly optimized for both *Summit* and *Frontier*'s multi-GPU architectures.

    - *State of the art:* To our knowledge, a comparable flow solver does not exist. The vast majority of flow solvers are written in compiled languages, which complicates transitioning from CPUs to GPUs and between different GPU architectures (e.g., NVIDIA versus AMD). Most flow solvers would require extensive rewrites to integrate deep learning libraries.

2. **Massive-scale, distributed data assimilation using adjoint-based optimization and differentiable programming:** *PyFlowCL*'s foundation upon *PyTorch* enables deep integration of ML tasks. These fully utilize *PyFlowCL*'s MPI runtime for massive-scale distributed learning. Additionally, *PyFlowCL* leverages *PyTorch*'s algorithmic differentiation (AD) capabilities to evaluate the derivatives of PDE-embedded closure models, which are required to solve adjoint PDEs during data assimilation. The use of AD dramatically simplifies the adjoint implementation and enables rapid prototyping of alternative sets of governing PDEs (e.g., exchanging fluid mechanics for solid mechanics).

    - *State of the art:* Differentiable programming has only recently become popular for data assimilation. To our knowledge, no other solver utilizes differentiable programming/AD for solver-embedded optimization tasks, and none can achieve the same degree of data parallelism.

PyFlowCL's GPU offloading capabilities are derived from the *PyTorch* library and are thus machine-agnostic. On *Summit*, *PyFlowCL* achieves a 40x single-node speedup using six V100 GPUs compared to 40 POWER9 CPU cores. The proposed simulations will be performed in the range of 100–2000 Summit nodes. *PyTorch* has been optimized for both NVIDIA and AMD GPUs, hence the transition from *Summit* to *Frontier* is expected to be straightforward. *PyFlowCL* performs well on x86-based platforms such as *Frontier*.

The DNS-production phase has an intense analysis component to ensure validity of the data being produced. *PyFlowCL* automatically alerts the PIs to runtime DNS problems, which minimizes wasted compute time. All verification and postprocessing codes are hybrid-parallelized using both MPI domain decomposition and OpenMP threading, and visualization is done remotely using *Paraview*. The data-filtering phase is automated using Python and shell scripts, which have been tuned to minimize disk overhead by archiving data, and ensure fault tolerance by automatically checking filtered data for validity. Likewise, model training is scripted to ensure all models are trained equally.

3.3 Parallel Performance

The PIs were granted an OLCF Director's Discretionary Allocation for testing and development of *PyFlowCL* on *Summit* in preparation for this INCITE allocation request.

Strong and weak scaling for large-scale turbulence simulations on *Summit* is shown in **Fig. 5**. Problem sizes ranging from $N^3 = 128^3$ ($\sim$ 2 million points) to $N^3 = 2048^3$ ($\sim$ 8.6 billion points) were tested on up to 16,384 *Summit* GPUs. For all cases, virtually all computation takes place on the GPUs. A narrow range of near-ideal strong scaling exists before reaching the limits of available GPU memory, which occurs at approximately 17 million mesh points per GPU. The code's weak scaling shows a minimum of 80 % parallel efficiency for the range of problem sizes tested on up to 16,384 *Summit* GPUs.

GPU utilization for fewer than 1×10^6 mesh points per GPU is low enough that CPU–GPU memory copies and inter-node communication become bottlenecks. Ongoing development work (Section 3.4) will optimize *PyFlowCL* for *Summit*'s GPU Direct inter-node communication. At this time, the impact of *Frontier*'s significantly different node and network topologies on our parallel performance is unknown.

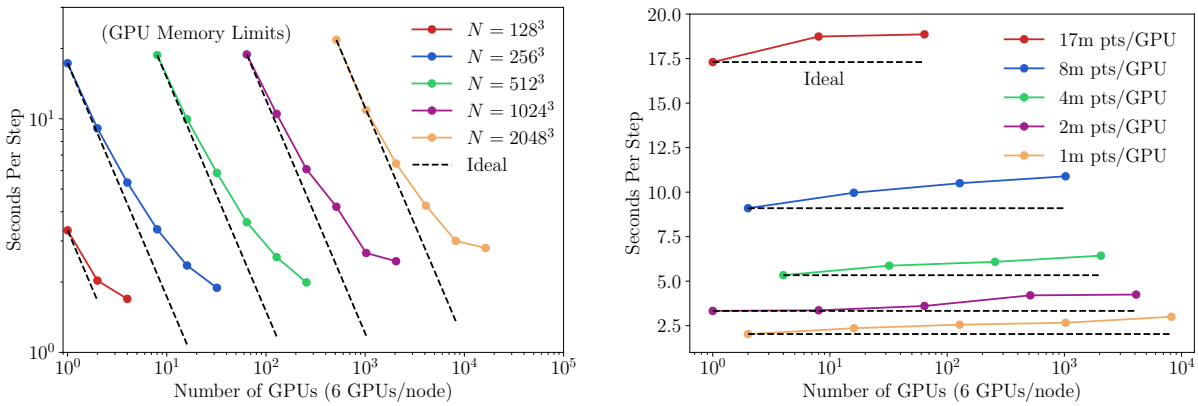


Figure 5: Strong (left) and weak (right) scaling of *PyFlowCL* on *Summit* using up to 16,384 GPUs (2731 nodes). Strong scaling results are shown for varying problem sizes N. Weak scaling results are shown for fixed work (points) per GPU.

For distributed model training, *PyFlowCL* achieves 97 % parallel efficiency on up to 3600 GPUs (600 *Summit* nodes). Weak scaling results using increasing ML problem sizes are listed in **Table 2**; this is the metric recommended by OLCF to measure the scalability of ML/AI codes. The excellent parallel efficiency of our code for training will enable the optimization of DL models over massive DNS datasets, which is particularly important given the long DNS simulation time-horizons necessary to achieve convergence to the statistical steady state (i.e., to converge time-averaged statistics such as mean profiles).

Table 2: Model optimization: Computational time per optimization iteration and optimization parallel efficiency on *Summit*. Parallel efficiency is calculated using the weak scaling formula $\frac{\text{Time for training on } M \text{ DNS data samples on 1 node}}{\text{Time for } N \times M \text{ DNS data samples on } N \text{ nodes}} \times 100\%$. Each GPU can optimize over $M = 8$ DL-LES simulations of a time length $[t, t+\tau]$. Problem size in the table below is reported as the number of DNS data samples (i.e., the size of the dataset) over which optimization was performed.

Problem Size	# Nodes	# GPUs	Walltime (s)	Parallel Efficiency
48	1	6	76.31	
480	10	60	76.65	99.6%
2,400	50	300	78.50	97%
4,800	100	600	78.55	97%
9,600	200	1200	78.91	97%
19,200	400	2400	78.80	97%
28,800	600	3600	78.73	97%

3.4 Developmental Work

Our development efforts to date have focused on memory performance and cache locality in order to optimize for SIMD performance on GPUs, as well as inter-GPU parallel performance (e.g., minimizing the number of CPU–GPU memory copies). This has been done on *Summit*, which is our primary proposed INCITE resource. We will similarly optimize for *Frontier* during the proposed allocation.

During the proposed INCITE campaign, we will continue to optimize *PyFlowCL* for *Summit*'s GPU Direct capabilities, which will be done in Q1'23. Preliminary "codelet"-based testing by PI MacArt indicates that rewriting and verifying several communications and memory-management kernels should result in significant scalability improvements (Fig 5). The large-scale simulation campaign development will be accelerated by these planned developments, and additional DNS databases will be possible with the improved efficiency. The project will not be severely impacted by the code's current efficiency. Milestone 4 will depend partially on the implementation of the hybrid multi-fidelity DL training method, which will also be done in Q1'23.

References

[1] F. Menter, A. Huppe, A. Matyushenko, D. Kolmogorov, An overview of hybrid RANS–LES models developed for industrial CFD, *Applied Sciences* 11 (2021) 2459.

[2] P. Durbin, B. Pettersson, *Statistical Theory of Turbulent Flows*, John Wiley and Sons, 2003.

[3] K. Hanjalic, B. Launder, *Modelling Turbulence in Engineering and the Environment: Second-Moment Routes to Closure*, Cambridge University Press, 2011.

[4] J. Smagorinsky, General circulation experiments with the primitive equations I. The basic experiment, *Monthly Weather Review* 91 (3) (1963) 99–164.

[5] M. Germano, U. Piomelli, P. Moin, W. H. Cabot, A dynamic subgrid-scale eddy viscosity model, *Physics of Fluids* 3 (1991) 1760–1765.

[6] D. K. Lilly, A proposed modification of the Germano subgrid-scale closure method, *Physics of Fluids* 4 (1992) 633–635.

[7] P. Moin, K. Squires, W. Cabot, S. Lee, A dynamic subgrid-scale model for compressible turbulence and scalar transport, *Physics of Fluids* 3 (11) (1991) 2746–2757.

[8] S. T. Bose, G. I. Park, Wall-Modeled Large-Eddy Simulation for Complex Turbulent Flows, *Annual Review of Fluid Mechanics* 50 (2018) 535–561.

[9] H.-J. Kaltenbach, H. Choi, Large-eddy an airfoil simulation of flow around on a structured mesh, in: Center for Turbulence Research Annual Research Briefs, 1995, pp. 51–60.

[10] U. Piomelli, E. Balaras, Wall-Layer Models for Large-Eddy Simulations, *Annual Review of Fluid Mechanics* 34 (2002) 349–374.

[11] P. Spalart, V. Venkatakrishnan, On the role and challenges of CFD in the aerospace industry, *The Aeronautical Journal* 120 (1223) (2016) 209–232.

[12] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *Journal of Fluid Mechanics* 807 (2016) 155–166.

[13] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence Modeling in the Age of Data, *Annual Review of Fluid Mechanics* 51 (2019) 357–377.

[14] K. Duraisamy, Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence, *Physical Review Fluids* 6 (5) (2021) 050504.

[15] J. Sirignano, J. F. MacArt, K. Spiliopoulos, PDE-constrained Models with Neural Network Terms: Optimization and Global Convergence, Invited Revision at *Journal of Computational Physics* (2021) arXiv:2105.08633.

[16] J. Sirignano, J. F. MacArt, J. B. Freund, DPM: A deep learning PDE augmentation method with application to large-eddy simulation, *Journal of Computational Physics* 423 (2020) 109811.

[17] J. F. MacArt, J. Sirignano, J. B. Freund, Embedded training of neural-network subgrid-scale turbulence models, *Physical Review Fluids* 6 (2021) 050502.

[18] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *Journal of Fluid Mechanics* 807 (2016) 155–166.

[19] J. F. MacArt, J. Sirignano, M. Panesi, Deep Learning Closure of the Navier–Stokes Equations for Transitional Flows, in: AIAA SciTech Forum, 2022.

[20] J. Ling, R. Jones, J. Templeton, Machine learning strategies for systems with invariance properties, *Journal of Computational Physics* 318 (2016) 22–35.

[21] M. P. Brenner, J. D. Eldredge, J. B. Freund, Perspective on machine learning for advancing fluid mechanics, *Physical Review Fluids* 4 (10) (2019) 100501.

[22] S. L. Brunton, B. R. Noack, P. Koumoutsakos, Machine Learning for Fluid Mechanics, *Annual Review of Fluid Mechanics* 52 (1) (2020) 477–508.

[23] D. Kochkov, J. Smith, A. Alieva, Q. Wang, M. Brenner, S. Hoyer, Machine learning-accelerated computational fluid dynamics, *Proceedings of the National Academy of Sciences* 118 (21) (2021) 2101784118.

[24] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Fluid Mechanics* 378 (2019) 686–707.

[25] P. C. Di Leni, L. Lu, C. Meneveau, G. Karniadakis, T. A. Zaki, *DeepONet prediction of linear instability waves in high-speed boundary layers* (2021) arXiv:2105.08697.

[26] L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nature Machine Intelligence* 3 (3) (2021) 218–229.

[27] C. Meneveau, J. Katz, Scale-Invariance and Turbulence Models for Large-Eddy Simulation, *Annual Review of Fluid Mechanics* 32 (2000) 1–32.

[28] B. Vreman, B. Geurts, H. Kuerten, Subgrid-modelling in LES of compressible flow, *Applied Scientific Research* 54 (3) (1995) 191–203.

[29] S. Ghosal, An Analysis of Numerical Errors in Large-Eddy Simulations of Turbulence, *Journal of Computational Physics* 125 (1) (1996) 187–206.

[30] A. G. Kravchenko, P. Moin, On the Effect of Numerical Errors in Large Eddy Simulations of Turbulent Flows, *Journal of Computational Physics* 131 (2) (1997) 310–322.

[31] F. K. Chow, P. Moin, A further study of numerical errors in large-eddy simulations, *Journal of Computational Physics* 184 (2) (2003) 366–380.

[32] S. K. Lele, Compact finite difference schemes with spectral-like resolution, *Journal of Computational Physics* 103 (1) (1992) 16–42.

[33] B. Fiorina, S. K. Lele, An artificial nonlinear diffusivity method for supersonic reacting flows with shocks, *Journal of Computational Physics* 222 (1) (2007) 246–264.

[34] S. Kawai, S. K. Lele, Localized artificial diffusivity scheme for discontinuity capturing on curvilinear meshes, *Journal of Computational Physics* 227 (22) (2008) 9498–9526.

[35] B. Steiner, Z. DeVito, S. Chintala, S. Gross, A. Paszke, F. Massa, A. Lerer, G. Chanan, Z. Lin, E. Yang, A. Desmaison, A. Tejani, A. Kopf, J. Bradbury, L. Antiga, M. Raison, N. Gimelshein, S. Chilamkurthy, T. Killeen, L. Fang, J. Bai, Pytorch: An imperative style, high-performance deep learning library, in: NeurIPS 2019, 2019.

[36] R. W. Davis, E. F. Moore, A numerical study of vortex shedding from rectangles, *Journal of Fluid Mechanics* 116 (1982) 475–506.

[37] C. Norberg, Flow around rectangular cylinders: Pressure forces and wake frequencies, *Journal of Wind Engineering and Industrial Aerodynamics* 49 (1-3) (1993) 187–196.

[38] D. Lyn, S. Einav, W. Rodi, J.-H. Park, A laser-Doppler velocimetry study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder, *Journal of Fluid Mechanics* 304 (1995) 285–319.

[39] S. C. Luo, Y. T. Chew, Y. T. Ng, Characteristics of square cylinder wake transition flows, *Physics of Fluids* 15 (9) (2003) 2549–2559.

[40] R. A. Clark, J. H. Ferziger, W. C. Reynolds, Evaluation of subgrid-scale models using a fully simulated turbulent flow, *Journal of Fluid Mechanics* 91 (1) (1979) 1–16.

[41] Y. Fabre, G. Balarac, Development of a new dynamic procedure for the Clark model of the subgrid-scale scalar flux using the concept of optimal estimator, *Physics of Fluids* 23 (11) (2011) 115103.

[42] J. O'Brien, C. A. Z. Towery, P. E. Hamlington, M. Ihme, A. Y. Poludnenko, J. Urzay, The cross-scale physical-space transfer of kinetic energy in turbulent premixed flames, *Proceedings of the Combustion Institute* 36 (2) (2017) 1967–1975.

PERSONNEL JUSTIFICATION AND MANAGEMENT PLAN

PERSONNEL JUSTIFICATION

The proposed research is highly interdisciplinary, requiring the development of models merging the fields of CFD and HPC (MacArt) with machine learning and optimization (Sirignano). **PI MacArt** has extensive experience with massive-scale multiphysics simulations, LES, and HPC software development. He served as a member of the DoE/NNSA PSAAP II Center for Exascale Simulation of Plasma-coupled Combustion at the University of Illinois at Urbana–Champaign, where his role was the development of nonequilibrium supersonic-combustion models and the development and implementation of high-performance scientific deep learning algorithms. **Co-PI Sirignano** brings experience in optimization, deep learning, and their application to PDEs. His recent research focuses on developing deep learning methods for PDE applications. He was also a co-PI on the DoE/NNSA PSAAP III Center at the University of Illinois at Urbana–Champaign from 2020–2021.

The proposed research will build upon a *longstanding collaboration between the PIs*, which has produced a number of articles on deep learning closure models for fluid dynamics [15–17, 19].

Jonathan MacArt (PI, Aerospace and Mechanical Engineering, Notre Dame, jmacart@nd.edu) is an expert in massive-scale multi-physics simulations, large-eddy simulation for turbulent combustion, and high-performance computing software development.

Justin Sirignano (co-PI, Mathematics, University of Oxford, and Industrial & Enterprise Systems Engineering, University of Illinois at Urbana–Champaign, justin.sirignano@maths.ox.ac.uk) is a researcher in applied mathematics, stochastic modeling, machine learning, and deep learning.

MANAGEMENT

The Notre Dame PI and Oxford PI will collaborate closely on the proposed research. The Notre Dame team will lead the computational fluid dynamics and high-performance computing aspects of the proposed research, including generating DNS data for the proposed DL-LES models. The Oxford team will lead on the development of the deep learning closure model, optimization methods for training the deep learning LES and deep learning hybrid RANS–LES models, and training the models.

Two Ph.D. students (one each at Notre Dame and Oxford) will be supervised over the course of the project and will be involved in the computational tasks under the direct supervision of the PIs. The interdisciplinary team already collaborates closely, with multiple weekly meetings, email chains, and Slack channels. The team meets annually for intense, two-week summer collaboration sessions as part of externally funded projects and with University funding.

PI MacArt will be responsible for technical progress reports, which will be authored quarterly. These will include task-by-task updates, progress on the estimated timeline, pacing risks and mitigation strategies, and outcomes from journal publications and conference presentations. A Final Report will be delivered upon project completion. As with their results and software products to date, the PIs will publish the sponsored results in highly cited journals and will make the resulting codes available online with open-source licenses.

The PIs' research and the development of *PyFlowCL* is supported by the Office of Naval Research and the PIs' internal University research funding. No further funding sources are needed.

MILESTONE TABLE

Proposal Title: Extreme-Scale Data Assimilation for Predictive Flow Simulations

Milestone	Details	Dates
1. Develop Deep Learning LES (DL-LES) models for Separated Wakes	**Resource:** *Summit* **Node-hours:** 598,000 **Filesystem TB:** 502 **Archival TB:** 497 **Software Application:** *PyFlowCL* **Tasks:** • Generate DNS datasets for rectangular cylinders for different aspect ratios and Reynolds numbers. • Divide DNS datasets into **training** and **test** (§2.2.2). • Train DL-LES models on the training datasets using the adjoint PDE optimization methods that we have developed. • Evaluate performance on test datasets and compare against existing LES models. **Dependencies:** None, but Milestone #2 will improve performance.	Q1'23–Q2'23
2. Software Design and Development	**Resource:** *Frontier* **Node-hours:** 10,000 **Filesystem TB:** 10 **Archival TB:** 0 **Software Application:** *PyFlowCL* **Tasks:** • Optimize parallel solver for *Frontier*'s Cray/AMD nodes. • Develop and incorporate hybrid multi-fidelity training in *PyFlowCL*. **Dependencies:** None	Q1'23
3. Evaluate DL-LES models for Wakes in Additional Geometries	**Resource:** *Summit* **Node-hours:** 225,700 **Filesystem TB:** 362 **Archival TB:** 362 **Software Application:** *PyFlowCL* **Tasks:** • Generate DNS datasets for circular cylinders and triangular flameholders at different Reynolds numbers. • Using the trained model for the rectangular cylinder, simulate out-of-sample on the new configurations and evaluate performance in comparison to existing LES models. **Dependencies:** Completion of Milestone #1	Q3'23
4. Develop DL-LES Models for Airfoil Flows	**Resource:** *Summit* **Node-hours:** 259,500 **Filesystem TB:** 502 **Archival TB:** 497 **Software Application:** *PyFlowCL* **Tasks:** • Generate DNS datasets for airfoil flows at different Reynolds numbers and angles-of-attack. • Train DL-LES models using adjoint-PDE optimization methods. • Evaluate performance on test datasets and compare against existing LES models. • Evaluate hybrid multi-fidelity training methods using lower-fidelity DL-LES data and compare against models trained using DNS data. **Dependencies:** Completion of Milestones #2 and #3	Q4'23

JONATHAN F. MACART
369 Fitzpatrick Hall of Engineering
University of Notre Dame, Notre Dame, IN 46556
(574) 631-6676
jmacart@nd.edu

Professional Preparation

B.S., Aerospace Engineering	University of Notre Dame	2013
M.A., Mechanical and Aerospace Engineering	Princeton University	2015
Ph.D., Mechanical and Aerospace Engineering	Princeton University	2018
Postdoctoral Scholar	Univ. Illinois Urbana–Champaign	2018–2020

Appointments

Assistant Professor, University of Notre Dame, Notre Dame, IN 2020–present

Five Publications Most Relevant to This Proposal

1. MacArt JF, Sirignano J, Panesi M. Deep Learning Closure of the Navier–Stokes Equations for Transitional Flows. In: AIAA SciTech proceedings, 2022.
2. MacArt JF, Sirignano J, Freund JB. Embedded training of neural-network subgrid-scale turbulence models. Physical Review Fluids. 2021; 6(5):050502.
3. Sirignano J, MacArt JF, Freund JB. DPM: A deep learning PDE augmentation method with application to large–eddy simulation. Journal of Computational Physics. 2020; 423:109811.
4. Sirignano J, MacArt JF, Spiliopoulos K. PDE-constrained Models with Neural Network Terms: Optimization and Global Convergence. Forthcoming, arxiv:2105.08633.
5. Nunno AC, Perry BA, MacArt JF, Mueller ME. Data-driven dimension reduction in turbulent combustion: Utility and limitations. AIAA Scitech 2019 Forum. 2019.

Research Interests and Expertise

Dr. MacArt's research interests are in turbulent reacting flows, high-speed and plasma-coupled combustion, heterogeneous reactive materials, novel statistical and modeling frameworks, and deep-learning methods for closure modeling and flow control. He has expertise in extreme-scale, massively parallel predictive simulations of turbulence and combustion, multiphysics and multiscale modeling of reacting systems, integration of high-fidelity turbulence, transport, and kinetic models, and advanced physics-constrained deep learning methods for turbulence and combustion.

Synergistic Activities

1. Lead developer of *PyFlowCL*, a GPU-accelerated, Python-based DNS/LES platform for training and testing deep learning-based models for turbulence simulations (with J. Sirignano).
2. Development and validation of multi-physics simulations of laser-induced breakdown (LIB) and subsequent ignition in canonical scramjet-relevant geometries (in collaboration with J.B. Freund, M. Panesi, and G. Elliott).
3. Development and validation of deep-learning closures for nonequilibrium effects on Navier–Stokes models in the transitional regime of hypersonic flow (with J. Sirignano and M. Panesi).

Collaborators (*past 5 years including name and current institution*)

David Buchta, Johns Hopkins University
Clayton Byers, Trinity College (CT)
Gregory S. Elliott, Univ. Illinois Urbana–Champaign
Jonathan B. Freund, Univ. Illinois Urbana–Champaign

Temistocle Grenga, RWTH Aachen University
Marcus Hultmark, Princeton University
Jinyoung Lee, Princeton University
Michael E. Mueller, Princeton University
Alessandro Munafo, Univ. Illinois Urbana–Champaign
Munetake Nishihara, FGC Plasma Inc.
Austin C. Nunno, Argonne National Laboratory
Marco Panesi, Univ. Illinois Urbana–Champaign
Bruce A. Perry, National Renewable Energy Laboratory
Pavel P. Popov, San Diego State University
Justin Sirignano, University of Oxford
Konstantinos Spiliopoulos, Boston University
Jonathan M. Wang, Stanford University

Bio-Sketch

Justin Sirignano
E-mail: Justin.Sirignano@maths.ox.ac.uk

Professional Preparation

Princeton University, B.S.E. 2006-2010
Princeton, New Jersey
Major: Operations Research & Financial Engineering

Stanford University 2010-2015
Stanford, California
PhD in Management Science & Engineering

Chapman Fellow, Imperial College London 2015-2016
London, United Kingdom
Department of Mathematics

Appointments

Associate Professor, University of Oxford July 2020-
Mathematical Institute

Assistant Professor, University of Illinois at Urbana-Champaign Aug. 2016-
Department of Industrial & Enterprise Systems Engineering
Currently on a leave of absence

Five Publications Most Relevant to this Proposal

1. J. Sirignano and K. Spiliopoulos. "DGM: A Deep Learning Algorithm for solving Partial Differential Equations." *Journal of Computational Physics*, 375, 1339–1364, 2018.

2. J. Sirignano, J. F. MacArt, and J. Freund. "DPM: A deep learning PDE augmentation method with application to large-eddy simulation." *Journal of Computational Physics*, 423, 2020.

3. J. F. MacArt, J. Sirignano, and J. Freund. "Embedded training of neural-network sub-grid-scale turbulence models. *Physical Review of Fluids*, 6 (5), 2021.

4. J. Sirignano and K. Spiliopoulos. "Mean Field Analysis of Neural Networks: A Law of Large Numbers". *SIAM Journal on Applied Mathematics*, 80 (2), 725-752, 2020.

5. J. Sirignano and K. Spiliopoulos. "Online Adjoint Methods for Optimization of Partial Differential Equations". arXiv:2101.09621. *Applied Mathematics and Optimization*, In Press, 2022.

Research Interests and Expertise

Justin Sirignano's research is at the intersection of Applied Mathematics, Machine Learning, and High Performance Computing. His recent research has focused on the mathematical theory and applications of Deep Learning. He is developing deep learning methods for constructing partial differential equation (PDE) models from data, which has a variety of applications in science and engineering. This includes recent work on developing deep learning-based PDE models as reduced-order simulations for "computationally-challenging physics" involving turbulent flows, whose accurate modeling is critical for flight vehicle design. In joint work with Jonathan MacArt and Jonathan Freund, he has developed deep learning closure models for large-eddy simulation (DL-LES). The DL-LES model results have been published in the *Journal of Computational*

Physics and *Physical Review of Fluids*. Justin also works on a range of other applied mathematics topics, including: computational methods and stochastic models in financial mathematics, asymptotic analysis of deep neural networks (e.g., law of large numbers, central limit theorems, and global convergence analysis), and stochastic online algorithms for optimizing over high-dimensional computationally-intensive simulations. His research has been previously supported by computational grants on the Blue Waters supercomputer and the Summit supercomputer.

Synergistic Activities

- Faculty member in the Data Science Group in the Mathematical Institute at the University of Oxford
- Faculty member in the Center for Hypersonics and Entry Systems Studies (CHESS)
- Faculty member in the Center for Doctoral Training (CDT) in Mathematics of Random Systems
- Instructor for Deep Learning courses at UIUC and Oxford
- I was a Co-PI for a $16.5 million Department of Energy PSAAP III Center at UIUC (2020-2021), where my role was to lead the development of scientific machine learning models. (I have left my role in the Center due to joining the University of Oxford.)

Collaborators

- Jonathan F. MacArt, University of Notre Dame
- Jonathan Freund, UIUC
- Marco Panesi, UIUC
- Konstantinos Spiliopoulos, Boston University
- Samuel Cohen, University of Oxford
- Rama Cont, University of Oxford
- Kay Giesecke, Stanford University
- Apaar Sadhwani, Google
- Gustavo Schwenkler, Santa Clara University
- Ziheng Wang, University of Oxford
- Deqing Jiang, University of Oxford
- Xiaobo Dong, UIUC
- Lei Fan, UIUC

Section 6: Software Applications and Packages

Question #1

Please list any software packages used by the project, and indicate if they are on open source or export controlled.

Application Packages

Package Name

PyTorch

Indicate whether Open Source or Export Controlled.

Open Source

Package Name

mpi4py

Indicate whether Open Source or Export Controlled.

Open Source

Package Name

HDF5

Indicate whether Open Source or Export Controlled.

Open Source

Package Name

h5py

Indicate whether Open Source or Export Controlled.

Open Source

Section 7: Wrap-Up Questions

Question #1

National Security Decision Directive (NSDD) 189 defines Fundamental Research as "basic and applied research in science and engineering, the results of which ordinarily are published and shared broadly within the scientific community, as distinguished from proprietary research and from industrial development, design, production, and product utilization, the results of which ordinarily are restricted for proprietary or national security reasons." Publicly Available Information is defined as information obtainable free of charge (other than minor shipping or copying fees) and without restriction, which is available via the internet, journal publications, textbooks, articles, newspapers, magazines, etc.

The INCITE program distinguishes between the generation of proprietary information (deemed a proprietary project) and the use of proprietary information as input. In the latter, the project may be considered as Fundamental Research or nonproprietary under the terms of the nonproprietary user agreement. Proprietary information, including computer codes and data, brought into the LCF for use by the project - but not for generation of new intellectual property, etc., using the facility resources - may be protected under a nonproprietary user agreement.

Proprietary Information

Are the proposed project and its intended outcome considered Fundamental Research or Publicly Available Information?

Yes

Will the proposed project use proprietary information, intellectual property, or licensing?

No

Will the proposed project generate proprietary information, intellectual property, or licensing as the result of the work being proposed?

If the response is Yes, please contact the INCITE manager, INCITE@doeleadershipcomputing.org, prior to submittal to discuss the INCITE policy on proprietary work.

No

Question #2

The following questions are provided to determine whether research associated with an INCITE proposal may be export controlled. Responding to these questions can facilitate - but not substitute for - any export control review required for this proposal.

PIs are responsible for knowing whether their project uses or generates sensitive or restricted information. Department of Energy systems contain only data related to scientific research and do not contain personally identifiable information. Therefore, you should answer "Yes" if your project uses or generates data that fall under the Privacy Act of 1974 U.S.C. 552a. Use of high-performance computing resources to store, manipulate, or remotely access any national security information is prohibited. This includes, but is not limited to, classified information, unclassified controlled nuclear information (UCNI); naval nuclear propulsion information (NNPI); and the design or development of nuclear, biological, or chemical weapons or of any weapons of mass destruction. For more information contact the Office of Domestic and International Energy Policy, Department of Energy, Washington DC 20585, 202-586-9211.

Export Control

Does this project use or generate sensitive or restricted information?

No

Does the proposed project involve any of the following areas?

i. Military, space craft, satellites, missiles, and associated hardware, software or technical data

ii. Nuclear reactors and components, nuclear material enrichment equipment, components (Trigger List) and associated hardware, software or technical data

iii. Encryption above 128 bit software (source and object code)

iv. Weapons of mass destruction or their precursors (nuclear, chemical and biological)

No

Does the proposed project involve International Traffic in Arms Regulations (ITAR)?

No

Question #3

The following questions deal with health data. PIs are responsible for knowing if their project uses any health data and if that data is protected. Note that certain health data may fall both within these questions as well as be considered sensitive as per question #2. Questions regarding these answers to these questions should be directed to the centers or program manager prior to submission.

Health Data

Will this project use health data?

No

Will this project use human health data?

No

Will this project use Protected Health Information (PHI)?

No

Question #4

The PI and designated Project Manager agree to the following:

Monitor Agreement

I certify that the information provided herein contains no proprietary or export control material and is correct to the best of my knowledge.

Yes

I agree to provide periodic updates of research accomplishments and to acknowledge INCITE and the LCF in publications resulting from an INCITE award.

Yes

I agree to monitor the usage associated with an INCITE award to ensure that usage is only for the project being described herein and that all U. S. Export Controls are complied with.

Yes

I understand that the INCITE program reserves the right to periodically redistribute allocations from underutilized projects.

Yes

Section 8: Outreach and Suggested Reviewers

Question #1

By what sources (colleagues, web sites, email notices, other) have you heard about the INCITE program? This information will help refine our outreach efforts.

Outreach

Question #2

Suggested Reviewers

Section 9: Testbed Resources

Question #1

The ALCF and OLCF have test bed resources for new technologies, details below. If you would like access to these resources to support the work in this proposal, please provide the information below. (1 Page Limit)

The OLCF Quantum Computing User Program is designed to enable research by providing a broad spectrum of user access to the best available quantum computing systems, evaluate technology by monitoring the breadth and performance of early quantum computing applications, and Engage the quantum computing community and support the growth of the quantum information science ecosystems. More information can be found here: https://www.olcf.ornl.gov/olcf-resources/compute-systems/quantum-computing-user-program/quantum-computing-user-support-documentation.

The ALCF AI Testbed provides access to next-generation of AI-accelerator machines to enable evaluation of both hardware and workflows. Current hardware available includes Cerebras C-2, Graphcore MK1, Groq, Habana Gaudi, and SambaNova Dataflow. New hardware is regularly acquired as it becomes available. Up to date information can be found here: https://www.alcf.anl.gov/alcf-ai-testbed.

Describe the experiments you would be interested in performing, resources required, and their relationship to the current proposal. Please note, these are smaller experimental resources and a large amount of resources are not available. Instead, these resources are to explore the possibilities for these technologies might innovate future work. This request does not contribute to the 15-page proposal limit.

2022_INCITE_Testbed.pdf
The attachment is on the following page.

TESTBED RESOURCES

Testbed resources are not required.