Cover Page

Sponsoring Institution	Brookhaven National Laboratory
Mailing Address	2 Center St, P.O.Box 5000, Upton, NY 11973-5000
Proposal Title	A Scalable and Distributed Machine Learning Service for Data-Intensive Applications
Principal Investigator	Torre Wenaus
Phone Number	631 681 7892
Email Address	wenaus@gmail.com (or wenaus@bnl.gov)
Key Personnel	Alexei Klimentov Meifeng Lin Tadashi Maeno Paul Nilsson
Collaborating Institutions	**Institutional PIs** and Collaborators
BNL Physics Department	**Torre Wenaus**, Alexei Klimentov, Paul Laycock, Tadashi Maeno, Paul Nilsson, Brett Viren
BNL Computational Science Initiative (CSI)	Meifeng Lin
UT Arlington	**Kaushik De**, FaHui Lin, Fernando Barreiro Megino
U Wisconsin Madison	**Rui Zhang**, Wen Guan
U Massachussetts Amherst	**Rafael Coelho Lopes de Sa**, Verena Martinez Outschoorn
Argonne National Laboratory	**Evangelos Kourlitis**
Jefferson Lab	**Markus Diefenthaler**, Malachi Schram

A Scalable and Distributed Machine Learning Service for Data-Intensive Applications

T. Wenaus, Brookhaven National Laboratory (Principal Investigator)
K. De, University of Texas Arlington (Co-Investigator)
R. Zhang, University of Wisconsin Madison (Co-Investigator)
R. Coelho Lopes de Sa, University of Massachussetts Amherst (Co-Investigator)
E. Kourlitis, Argonne National Laboratory (Co-Investigator)
M. Diefenthaler, Jefferson Lab (Co-Investigator)

Artificial Intelligence and Machine Learning (AI/ML) applications are in a period of rapid growth and innovation in High Energy Physics (HEP), on the foundation of a powerful and rapidly evolving toolset. Access to much larger scale processing resources than is today's practice could greatly accelerate developing and refining AI/ML applications by shortening optimization and training latencies by orders of magnitude, enabling a transformative expansion of scientific creativity and innovation in conceiving and developing AI/ML applications. The DOE's Advanced Scientific Computing Advisory Committee produced a report in 2020 addressing priority research directions which for HEP cited as the first challenge, *Create usable tools for large-scale distributed training and optimization of ML models to enable physicists to scale up the complexity of their models by several orders of magnitude above the current "laptop-size"*. Addressing this is the central objective of this proposal.

This proposal assembles a team of scientific Workflow/Workload Management System (WMS) developers, computational scientists and domain scientists to work together on building a system providing easy, uniform access to diverse large scale computing resources including regional to global grids, HPCs and academic and commercial clouds. Our WMS team has world leading expertise and capability as the developers of the PanDA workload management system. Our domain science team brings together researchers experienced in AI/ML from ATLAS, DUNE, Belle II and the Jefferson Lab experimental program. Our computational scientists are experts in leveraging the largest HPC platforms, which we will integrate using a new Function as a Service capability that will integrate very well with PanDA. The delivered software will offer the community a low threshold of entry to a service that scales across geographically distributed sites and platform types, opening access to the widest array of processing resources accessible at a given time, while fully supporting the extensive AI/ML toolset researchers already use within sites.

An example of the transformative impact large scale resources can have is in ML-based per-event likelihood analysis. With highly scalable resources, very large ensembles of deep and wide NNs can be trained to provide for the first time an unbiased approximation of the exact likelihood ratio for large datasets such as those of the LHC, with large scale NNs also describing the effect of systematic uncertainties. This use case and several others drawn from our domain science teams are described in the proposal.

This proposal focuses its investment primarily upon supporting early career researchers (mostly postdocs) embedded in experimental teams working on AI/ML enabled analysis, and working with the (existing and independently funded) WMS team to bring the emerging large scale AI/ML services to bear on their research. A primary objective for the project is that it be a strong contributor to training an early career work force at the leading edge of large scale HEP AI/ML applications.

The project would deliver an operating large scale ML service instance available to the HEP community as part of the HEP AI/ML ecosystem, sited at BNL and easily accessible via federated login, packaged and documented such that interested parties can bring up their own instances. The service would be able to use resources on DOE HPCs/LCFs, commercial clouds (Google, Amazon, possibly NVIDIA), and a flexible array of cluster and grid resources. The scalability target is a 75k concurrency level for a single workflow.

Contents

1 Introduction **1**

2 Project Objectives **3**

3 Proposed Research and Methods **3**

3.1 TA1: Core Infrastructure . . . 4
3.1.1 PanDA and iDDS . . . 4
3.1.2 Resource-side Services . . . 6
3.2 TA2: Platform Support . . . 6
3.3 TA3: Experiment Agnostic Large Scale ML Services . . . 8
3.3.1 ML Model Training and Hyperparameter Optimization . . . 8
3.3.2 Simulation-Based Inference . . . 9
3.3.3 Evaluating Systematic Uncertainties . . . 9
3.3.4 Reinforcement Learning . . . 10
3.3.5 Managing User Code . . . 10
3.3.6 Interactivity and User Interface . . . 10
3.4 TA4: Applying Large Scale ML Services in the Experiments . . . 11
3.4.1 Practical per-event likelihood analysis using machine learning in ATLAS . . . 11
3.4.2 Large scale hyperparameter optimization for ATLAS and beyond . . . 13
3.4.3 ML in Integrated End-To-End Workflows: ATLAS Fast Chain . . . 14
3.4.4 Applying Large Scale ML Services at Jefferson Lab . . . 14
3.4.5 Applying Large Scale ML Services in the Intensity Frontier: DUNE and Belle II . . . 15
3.5 Project Methodology and Execution . . . 16
3.5.1 Impact on the HEP AI/ML Ecosystem . . . 16
3.5.2 Building Early Career Expertise . . . 16

4 Deliverables **17**

5 Timeline and Personnel **18**

Appendix 1: Biographical Sketches **21**

Appendix 2: Current and Pending Support **22**

Appendix 3: Bibliography and References Cited **23**

Appendix 4: Facilities and Other Resources **25**

Appendix 5: Equipment **31**

Appendix 6: Data Management Plan **32**

Appendix 7: Computational Resources **33**

Appendix 8: Recruitment and Retention of Students and Early-Stage Investigators **38**

1 Introduction

Artificial Intelligence and Machine Learning (AI/ML) applications are in a period of rapid growth and innovation in High Energy Physics (HEP) as in related fields and the wider world, on the foundation of a powerful and rapidly evolving open source toolset. Developing applications typically takes place on the desktop or local cluster, sometimes facilitated by modestly scaled GPU acceleration. Ready access to large scale resources could greatly accelerate developing and refining current applications that require substantial processing, by shortening optimization and training latencies by orders of magnitude. More importantly, such access could enable a transformative expansion of scientific creativity and innovation in conceiving and developing AI/ML applications. Lifting the practical constraints of working at the scale of owned/local resources can unshackle not just the applications but of conceptualizing them in the first place.

This is particularly true of HEP, with its large and complex datasets and corresponding processing demands, today and even more so in the future with the advent of the HL-LHC. The DOE's Advanced Scientific Computing Advisory Committee (ASCAC) empanelled a Subcommittee on AI/ML, Data-intensive Science and High-Performance Computing that produced a report in 2020 [1] addressing priority research directions. For HEP, the first item on the list of challenges (page 27) is *Create usable tools for large-scale distributed training and optimization of ML models to enable physicists to scale up the complexity of their models by several orders of magnitude above the current "laptop-size."*. Addressing this top priority HEP AI/ML ecosystem research challenge is the central objective of this proposal.

This proposal assembles a team of scientific Workflow/Workload Management System (WMS) developers, computational scientists and domain scientists to work together on building a transformative contribution to the HEP AI/ML ecosystem offering analysts easy, proven, uniform access to diverse large scale computing resources, including regional to global grids, HPCs and academic and commercial clouds. Our WMS team has world leading expertise and capability in scientific workflow management at the largest scales, built over the last 15 years in developing the PanDA [2] workload management system and the ecosystem around it, recently extending to complex workflows including AI/ML. Our domain science team brings together researchers experienced in AI/ML who have applications today as well as the vision for applications in the future for which the large scale ML services developed through this proposal will be transformative. Our computational scientists are experts in leveraging the largest HPC platforms, particularly DOE's Leadership Computing Facilities (LCFs), the integration of which will complete the spectrum of large scale resources made available in the infrastructure we develop.

This proposal focuses its investment and added value primarily upon supporting early career researchers – postdocs and graduate students – who will work within their experimental teams on AI/ML enabled analysis, and with the WMS team on bringing the emerging large scale AI/ML services to bear on their research. Their work will tightly couple the WMS development to the real science needs throughout the development process and across several experiments, ensuring that the project deliverables constitute the widely applicable HEP AI/ML ecosystem extension we aim for. An important objective for the project is that it be a strong contributor to building our future workforce and expertise base.

In our WMS objectives, with modest investment we will build on the existing PanDA workload management system and its ecosystem including the Intelligent Data Delivery Service (iDDS) [3] that already delivers key functionalities at scale for the ATLAS experiment. We will extend the functionality to support the full AI/ML development process: conceptualization, rapid prototyping, iterative development, training, optimization and application in analysis and production. Development will proceed concurrently with applying the available functionality and refining it iteratively, through collaboration between our WMS and domain science teams. The delivered software will offer AI/ML developers and users a low threshold of entry to a tool suite emphasizing usability and scalability, with powerful automation and monitoring tools offering to the researcher the widest array of processing resources accessible at a given time.

We focus on delivering highly scalable AI/ML services because we anticipate a proliferation of HEP AI/ML use cases for large scale services throughout the ecosystem. The domain scientists on our team are at the leading edge of this trend, and our early career work force will be trained at the intersection of large scale HEP AI/ML applications and the services we develop that are their enablers. For example in analysis, ML has to date been mostly limited to training classifiers to select data that are then analyzed with traditional statistical methods. However the best use of ML methods would be to directly approximate the exact likelihood ratio. A description of the exact likelihood ratio would build the test statistic per-event, extracting all the available information in our datasets. Such ML based per-event likelihoods have not been used to date in a real HEP analysis because the processing demands are much too great. With highly scalable resources, very large ensembles of deep and wide NNs can be trained to provide an unbiased approximation of the exact likelihood ratio, including additional NNs to describe the effect of systematic uncertainties. Through the proposed project we will implement these ideas in the ATLAS H→ZZ→4l analysis in the search for off-shell production, where preliminary results obtained using Google Cloud Platform resources show a result that is four times more sensitive than traditional methods [4].

This example encompasses the key features of our proposed research. We identify a domain science application for which highly scalable AI/ML resources would be transformative in extending the scientific reach of our experiments. Through a collaboration between our core WMS experts and the early career researchers supported through this project who work both with us and the experiment's analysis teams, we establish a real world testbed applying the project's tool set to the analysis problem, iteratively refining both in the course of this research, towards end deliverables of a tool set of demonstrated capability and young researchers versed in the latest future-directed techniques.

The objectives of the project are summarized concisely in Section 2. Section 3 presents the proposed research and methods, organized into Task Areas covering core infrastructure, the services we'll build, and how we'll apply them in the experiments, largely through the engagement of our postdoc team. Participating experiments bringing their expertise, use cases, and preparedness to host and train the project's early career work force include ATLAS, DUNE, Belle II, and the Jefferson Lab experimental program. We also summarize and highlight the impact on the HEP AI/ML ecosystem that this proposal targets. We describe our plans for a primary objective of the project, building early career expertise. In Section 4 we summarize the high level deliverables constituting the end products of the project. Finally we conclude the proposal with a specific and detailed timetable of activities in Section 5, indicating where we apply on-project effort and where we leverage off-project effort and resources, and in Section 5 we summarize the roles and responsibilities of participants and the personnel planned to be supported by the project.

The project funding we request is entirely for personnel (with a small travel component), with an emphasis on early career scientists embedded in experiment teams, and totalling slightly more than $1M/year for the three year duration of the project. We have come to the effort profile and funding level through a bottom up process of building our team of technical experts and experiment participants, planning where we can leverage externally supported effort and where we can most effectively apply supported effort towards our objectives – including prominently, training early career scientists. The PI will be Torre Wenaus (leader of BNL's Nuclear and Particle Physics Software Group), with participation from ATLAS, Belle II and DUNE collaborators at BNL; AI/ML and HPC experts from BNL's Computational Science Initiative (CSI); ATLAS groups at UT Arlington, U Wisconsin Madison, U Mass Amherst, and ANL; and scientists at Jefferson Lab pursuing AI/ML applications in their experimental programs.

The PI will take a coordinating role in collecting use cases and establishing a development program to address them. Postdocs/students will investigate use cases, developing testbeds using existing infrastructure in the first year and the evolved software in years two and three. The evolution will be informed by testbed experience, with the development work carried out in a collaboration between project-supported personnel and experts supported outside the project (the core PanDA/iDDS team, AI/ML professionals and science domain experts).

2 Project Objectives

1. Apply our team's world leading expertise and capability in scientific workflow management at the largest scales to offer ML analysts easy and uniform access to diverse large scale computing resources.
2. By enabling ready access to large scale ML processing resources, accelerate the iterative development and refinement process for ML applications by shortening optimization and training latencies by orders of magnitude.
3. Empower researchers employing AI/ML towards a transformative expansion of scientific creativity and innovation in conceiving and developing AI/ML applications, by making applications and models of much greater depth, complexity and power tractable.
4. Leverage work on advanced workflows and associated high quality user interfaces, both existing and underway in our experiment teams, towards creating a suite of experiment agnostic tools and services for scalable ML.
5. Leverage and add value to the powerful existing open source AI/ML tool set in the domain of distributed, multi-facility large scale applications.
6. Because of the existing capability we can draw on, this project can achieve its objectives by investing primarily in the support of early career researchers (mostly postdocs) to work on applying existing core infrastructure in real world AI/ML application testbeds.
7. By applying project resources primarily to the training of young postdocs in large scale AI/ML applications, working closely both ML services developers and experiment teams working actively on physics analysis, we will make this training a primary objective and deliverable of the project. The strong off-project leveraging of existing experts in our experiment teams makes this realizable.
8. The ultimate objective and deliverable of the project is a demonstrated experiment-agnostic capability, through several at-scale testbeds across several experiments, of a suite of scalable ML services that can be applied throughout the HEP AI/ML ecosystem.

3 Proposed Research and Methods

In this section we present details of our proposed research and methods. We organize our research activities into four main **Task Areas (TAs)** as follows.

- **TA1: Core infrastructure:** Development of workflow/workload management software in support of large scale ML services
- **TA2: Platform support:** Extending and adapting ML services and workflows to diverse platforms including HPCs/LCFs, commercial clouds, and accelerator-equipped clusters and grids

- **TA3: Experiment agnostic large scale ML services:** Developing, generalizing, scaling, and hardening of experiment agnostic ML services on the core infrastructure foundation.
- **TA4: Experiment applications:** Demonstrator applications in participating experiments utilizing large scale ML services

Our proposed plans and methods for TA1 and TA2 involve primarily core software experts from the PanDA and iDDS teams, together with the CSI team with their HPC expertise, and are presented in Section 3.1 and Section 3.2. TA3 involves both core software experts and scientists from our experiment teams developing AI/ML applications, and is presented in Section 3.3. TA4 is the principal focus of participating scientists on our experiment teams, particularly the early-stage investigators who make up most of the supported effort of the project, and is presented in Section 3.4.

3.1 TA1: Core Infrastructure

3.1.1 PanDA and iDDS

Our proposal has the ambitious objective of adding transformative value to the existing AI/ML tool set available within the HEP AI/ML ecosystem [5], by providing highly scalable ML services across diverse physically distributed computing facilities, with low barrier of entry and high usability. Today's ecosystem includes tools enabling large scale use of individual facilities, but does not include tools to integrate multiple facilities for the use of one AI/ML application that can benefit from the integrated scale and the dynamic availability of diverse resources. This is where our proposal adds value: adding a cross-facility capability to scale up workflows, while within individual facilities they can leverage the full existing tool set.

What makes this objective attainable is the strong leverage and technology base we have in the PanDA software stack and ecosystem. PanDA had its inception in 2005 focused on the needs of ATLAS. Since its adoption ATLAS-wide a few years later, it has been in continuous production for all managed workflows as well as individual analysis, benefiting from initial choices in architecture and technology stack that remain best in class for distributed services (database backed web services implemented in Python with REST-like distributed APIs). The stability and longevity of this WMS foundation has enabled ATLAS and the PanDA team to continuously refine, extend and modernize the capabilities and use cases the system covers, with high efficiency in terms of effort and cost thanks to a small and stable core team of expert developers. It has also enabled extending PanDA's application, with modest incremental effort, to experiments and use cases beyond ATLAS. The 2012-2015 DOE ASCR-supported BigPanDA project built on ATLAS PanDA to provide experiment-agnostic capability targeting HPCs and DOE LCFs. This effort brought OLCF Titan into PanDA and ATLAS, and led to OLCF extending PanDA Titan usage to other science domains. BigPanDA was also instrumental in the COMPASS experiment at the CERN SPS accelerator adopting PanDA, and in establishing a testbed integration of PanDA in the LSST's workflow management system. BigPanDA also seeded collaborations with Google and Amazon.

This record of leveraging and diversifying PanDA beyond ATLAS is recently bearing fruit in ways applicable to the AI/ML ecosystem. The HPC/LCF directed efforts led to the development of the PanDA Event Service enabling highly efficient utilization of these platforms, followed by the development of the Intelligent Data Delivery Service (iDDS) which provides experiment-agnostic capability for orchestrating complex fine-grained workflows. In 2021 the Rubin Observatory adopted PanDA, a decision driven by iDDS's powerful workflow support and PanDA's scalability. The complex workflow intrest in ATLAS, Rubin and the NSF's IRIS-HEP project (which joined in development of iDDS) led to applying PanDA/iDDS complex workflow support to ATLAS AI/ML applications, which in turn is the foundation for this proposal.

The iDDS service extends the workflow system with a fine-grained, dynamic responsiveness to complex data driven workflows. The first iDDS use case, ATLAS's "data carousel" [6], quickly demonstrated iDDS's utility and has been used in ATLAS production since 2020. The data carousel supports rapidly processing staged data as it appears from tape such that it can be quickly unpinned and deleted, thus enabling the sliding window of disk-resident data to be kept to a small fraction of the full tape-resident sample. This is one of the important tools by which ATLAS is addressing the HL-LHC storage challenge. The second production iDDS application is a foundation for this proposal, hyperparameter optimization (HPO), providing a fully automated HPO platform on top of geographically distributed resources. It is used with the first ML application to reach production in the ATLAS fast simulation; the new AtlFast3 simulation that began production in 2021 incorporates FastCaloGAN, discussed below. iDDS also supports a growing list of complex analysis workflows from the ATLAS analysis community, including Monte Carlo toy based confidence limits estimation (requiring multiple steps of grid scans, where current steps depend on previous steps), active learning (requiring complex logic between tasks in an analysis workflow), and end-to-end production+analysis workflows including REANA integration.

While it is in ATLAS that these iDDS based workflows have been applied thus far, their application is not limited to ATLAS, and through this proposal we aim to apply iDDS beyond ATLAS, as happened with Rubin, and is underway with the sPHENIX experiment at BNL's Relativistic Heavy Ion Collider (RHIC) which has recently adopted PanDA/iDDS. PanDA with iDDS offers a powerful set of existing capabilities we will leverage and build on in this proposal:

1) A distributed system able to operate transparently and coherently across heterogeneous resources.

2) Portability of development environments and applications.

3) Support for the full suite of open source AI/ML tools including those for utilizing large scale resources.

4) Convenient authentication supporting modern (OAuth2) and legacy (grid certificate) mechanisms.

5) Control, automation and monitoring systems that put the user in well informed control of the system.

6) Interactive latencies to support fast iterative development and quasi real-time applications.

7) A high quality user interface leveraging community tools, particularly the Python stack and Jupyter.

We have used these key attributes to implement support for large scale AI/ML workflows, currently in ATLAS production in the FastCaloGAN component of ATLAS's AtlFast3 [7] fast simulation discussed below, and in analysis prototypes. Crucially for the tractability of this proposal and its moderate, postdoc-directed workforce, our project will leverage the existing PanDA based capability and ongoing interest in AI/ML workflows within ATLAS, but the project will not directly support this ongoing work. ATLAS has long recognized that its own interests are well served by supporting generalization and wider use of its open source software (Rucio [8], PanDA, and ACTS [9] are prominent examples in the HEP community).

The large scale AI/ML workflow capability in PanDA is built from experiment agnostic components and can evolve to serve the HEP community ecosystem if supported to do so as this project proposes. (The last support for generalizing PanDA beyond ATLAS was the ASCR BigPanDA project that ended in 2018.) We propose to drive the development of an experiment and domain agnostic software stack by coupling the existing and externally supported core development team to a postdoc-based workforce working with multiple experiments and analysis teams to develop real-world testbeds that will establish both the generality and the capability of the core infrastructure. The mature state of the infrastructure to be leveraged, and the synergy with parallel efforts within ATLAS on expanded AI/ML services and pseudo-interactive analysis, make it realistic to attain this objective over the three year duration of the project.

3.1.2 Resource-side Services

Resource-side services that manage the interaction of workload and data management systems with the resources of a facility are important mechanisms to provide uniform capability across diverse resources, and to encapsulate their differences in order to present as simple and uniform interface as possible to higher levels of the system. Two important such components in PanDA are the PanDA Pilot [10], and the Harvester edge service [11, 12]. On a conventional system such as a batch scheduler based system, the pilot is the batch job that once running acquires payloads from PanDA for execution, encapsulating and insulating from PanDA the dynamic and highly varied worker node environment. Similarly at the level of an entire facility, the Harvester service encapsulates the mechanisms by which processing resources are acquired and utilized (e.g. by submitting pilots to batch queues), storage is managed and results are returned, and so on. Across most scientific computing facilities the similarities are greater than the differences, and tailoring resource-side services to a particular facility is straightforward. However for some, most notably DOE HPCs and particularly LCFs, every facility is substantially unique with respect to others, and implementing resource-side services like the pilot and Harvester is a highly involved and effort-intensive task.

DOE HPCs/LCFs are large resources that we want our software and our users to be able to effectively leverage; in fact whether they can do so will be an important success metric of the project. Future generations of these facilities are expected to have AI/ML as a principal application target. Accordingly, developing capable, effort-efficient mechanisms to integrate these machines is an important project task. We identify in particular the rapidly evolving Function as a Service (FaaS) area as a possible solution, and the funcX [13] project in particular. FuncX is a high-performance FaaS system to orchestrate scientific workloads across heterogeneous resources, from laptops to clusters to LCFs. It is an extension of Parsl [14], a Python parallel programming toolkit. FuncX operates as a persistent service serving as a gateway to route workloads to compute nodes. Users authenticate using standard OAuth2 and register their workloads with the funcX client which delivers them to the remote target endpoint asynchronously, and receives results back. The funcX team anticipates making it available on DOE HPCs including LCFs. Given the similarities in architecture and technology between PanDA and Parsl/funcX, integrating the two (via a Harvester plugin) should be straightforward, such that funcX provides a secure, capable access path for PanDA workloads to reach HPC/LCF platforms and be executed by a trusted locally resident service. This core task of interfacing to HPCs and LCFs, particularly via FaaS and by exploring Parsl/funcX, will be a focus particularly of our CSI based team of experts on HPCs, AI/ML and the application of AI/ML on HPCs.

3.2 TA2: Platform Support

A principal objective of the proposed project is to support multiple platforms and facilities for large scale AI/ML workflows, abstracting the details and complexities away from users as much as possible. A submitted user workflow may run on some combination of local cluster, grid, cloud and/or HPC resources depending on workflow requirements and dynamic availability of resources. There are three key layers and components involved: a friendly User Interface (UI) at the front end that abstracts away complexity while providing needed system comprehension, control and diagnostic capability; the intermediate provisioning layer that mates submitted workflows with the resources to execute them; and the back end execution environment(s) where particular platforms and facilities meet the workflow management infrastructure.

The UI will use Jupyter notebook technology, as a widely adopted choice in scientific data analysis including AI/ML, particularly suited to applications written in Python. PanDA (itself written in Python) has been integrated with Jupyter as a highly capable user front end. Several of our participating institutes offer and/or extensively use JupyterHub services, such as those provided by the BNL Scientific Data and Computing Center (SDCC), JLab and other National Laboratories and supercomputing centers, including NERSC. This will facilitate access for our ML service framework to these data centers.

Once workflow submissions are received through the UI, an intermediate provisioning layer performs the matchmaking between user request and available resources. Typically this is performed by PanDA and iDDS, with PanDA managing matchmaking and iDDS the granular execution of the workflow. Efficient execution requires intelligent decision making on execution environment even at the fine-grained level; e.g. subcomponents of the workflow may be sufficiently light in their processing demands but heavy in their potential latency contributions that they are best run on local resources (PanDA/iDDS includes the capability for such decision making). Performance analysis of the workflows will need to be implemented in order to automate workflow optimization. Exploring and evaluating a wide sampling of AI/ML workflows and use cases will be important to inform this optimization; this will be provided through the participation of many experiment teams, bringing their applications and use cases, in this proposal. Data handling is a further aspect that can have strong platform dependencies, particularly on novel platforms. We expect to leverage PanDA's tight integration with the Rucio distributed data management system, and Rucio's broad application in HEP, to ameliorate data handling issues. The provisioning services reside at a single location, likely to be the BNL Scientific Data and Computing Center (SDCC).

Particular features and unique complexities of platforms are addressed in the back end execution layer. An advantage of building on PanDA is that experience exists with virtually all platform types that will be encountered. A current exception is DOE LCFs; as discussed in Section 3.1.2 we will direct dedicated effort there. The ML service back end will perform the heavy lifting of the services where computation-intensive tasks such as ML model training and hyperparameter optimization will take place. Important portability and scalability aspects of the ML service framework pertain to the back end execution layer component.

- Portability. Given that the software environments and hardware architectures may vary between different back ends, the most portable solution is to dispatch the workflows in a containerized environment with portable ML frameworks. The challenge is, how do we turn simple user requests, likely in just a few high-level functions, into an executable that can be run on the compute nodes, likely with GPUs? Here the FaaS paradigm can be an asset for executing efficiently and portably on different architectures, by encapsulating and optimizing for the heterogeneity.
- Scalability. For distributed ML training, existing frameworks such as Apache Spark, Horovod and MPI_Learn are often employed. However, the suitability and scalability of each framework may depend on the target platform and the applications. Fast convergence of ML algorithms on large HPC machines requires a good communication model, synchronization mechanism and parameter selection methodology. The aim is to avoid scanning a huge search space to find the best solution. Our team is working on active learning approaches to optimize and minimize the search space in real-time. Building this capability into our scalable services, for AI/ML and other large scale analysis workloads, will substantially enhance the attainable application scale. We will draw on CSI work developing tools that can analyze the performance characteristics of applications and model their outcomes on different platforms, to optimally utilize a wide spectrum of computing resources.

Among the platforms we target we have discussed the challenges of HPCs, requiring significant effort particularly from our CSI team. The other platforms we will target are as follows. Information on specific facilities the project will use is in Appendix 4.

- Clusters and Grids are the low hanging fruit among platforms. They encompass the most widely available and easily supported (for a distributed computing system like PanDA) platform. New workflows will often be implemented here first, and will always be ported here.

- HPCs and LCFs can be a complex special case, as discussed, but many HPCs (NSF facilities, most facilities in Europe) are not greatly more complicated than grid resources, and can be readily integrated for a large throughput and capability return on investment.
- Google Compute Platform (GCP) will be an important platform as we can leverage the Google-ATLAS R&D project. We can spike usage to large scales to test scalability. GCP has been a fertile platform for exploratory R&D on analysis tools and approaches, including AI/ML.
- Amazon Web Services (AWS) will be important, drawing on the US ATLAS - Amazon collaboration. Via this project AWS has been a productive platform for analysis and HL-LHC computing R&D including AI/ML. Our PanDA team has a decade of experience using both Amazon and Google clouds.
- The NVIDIA HPC*AI Platform for HEP draws on NVIDIA's recent interest and investments in scientific data processing, ARM HPCs, and C++ and python ecosystems. Several of our long time DOE Lab collaborators have joined and lead NVIDIA's science program, giving us a valuable entry into this promising development. Discussions on collaboration begin in June 2022.

3.3 TA3: Experiment Agnostic Large Scale ML Services

Widely used distributed deep learning frameworks such as Tensorflow [15], PyTorch [16], and Horovod [17] use various strategies to distribute computation workloads over multiple GPUs, CPUs and compute nodes *within* a single cluster or site. Users with large scale compute intensive applications that they want to iteratively develop, train, optimize, refine and re-engineer over time are constrained in their ambitions and scientific creativity by the resources available on individual accessible resources. We propose to develop ecosystem-level services available to the community to submit machine learning workflows not just to single sites but across multiple sites which may be geographically distributed and of different platform types. By so doing we will greatly expand the scaling reach available to researchers, and lower the often substantial barriers between developing applications in a small highly usable environment and moving them to large scale resources with the capacity to process the deep and complex models that may give the greatest scientific return.

In the preceding two sections describing TA1 and TA2 we have described the capabilities, software, external leverage and platform expertise we can bring to bear on this objective. We describe here and address in Task Area 3 what we plan to build on this foundation and deliver in terms of experiment agnostic large scale ML services, tools and techniques. The following section describing TA4 is the heart of this proposal and locus of the majority of the supported effort, the domain science applications empowered by these services that will guide development, validate utility, and serve as examples and testbeds for community use.

3.3.1 ML Model Training and Hyperparameter Optimization

Clear candidates for large scale ML services are ML model training and hyperparameter optimization (HPO). HPO was the first ML service the PanDA/iDDS team implemented for ATLAS, and has been used for various analyses [18–21]. Generally there are two approaches to implementing HPO, both of which are supported by PanDA/iDDS. The most basic is to generate many hyperparameters in one round, randomly or over a grid, and then evaluate all of them to select the one producing the best results. This can be inefficient because it can result in searching many hyperparameters that have no chance of being a solution.

Another approach is to search hyperparameters iteratively, informed by previous results, such as Bayesian Optimization. This requires a master application to collect results and, informed by previous results, trigger successive rounds of hyperparameter generation and evaluation. The PanDA/iDDS team has developed an HPO service on this basis to manage the search through hyperparameter generations, distributing ML tasks to CPUs/GPUs on potentially geographically distributed resources to evaluate the hyperparameters. This constitutes a powerful tool for physicists to develop sophisticated, processing- and data-intensive ML applications, and will be an enabler for the scale of ML applications to be found at the HL-LHC.

In order to provide scalable cross-site capability for ML model training as well as HPO, we plan to extract from the current HPO service the machinery to run training and enhance it as a platform allowing users to manage ML models and run training interactively or in bulk, without the complexity of building and maintaining the infrastructure on top of heterogeneous geographically-distributed resources. We will also provide client tools and interfaces to help users to develop complex ML models including deep NNs and inject them into the platform.

3.3.2 Simulation-Based Inference

Large scale ML services can be an enabler for a powerful approach to a mainstay of HEP analysis, likelihood analysis. The traditional approach to hypothesis testing is to identify an observable sensitive to the hypothesis, make bins and assign probabilities based on the frequency observed in simulation. However the frequency of events in each bin can be very different from the actual probability of each event inside that bin. ML methods are used in many analyses to mitigate this problem using classifiers that distinguish signal from background processes; binned ML classifiers will naturally aggregate events that have similar probability in the same bin. However in this approach as well there is information loss. The simulation-based inference [22] approach takes a large and processing-intensive step further, by taking advantage of the ability of AI/ML methods to learn the actual probability densities and build the analysis **event by event**, extracting all the information available in the data collected. To date, no analysis at the LHC has been performed with these methods because only small scale toy examples have been possible. We propose here to build a highly scalable simulation-based inference capability able to support real-world analysis, exercise it in a real analysis, and make it available to the community. This is described in detail in Section 3.4.1.

3.3.3 Evaluating Systematic Uncertainties

A big challenge in simulation-based inference is the description of the effect of systematic uncertainties on the probability densities. In binned analyses, systematic uncertainties are described by the change in the frequency of a bin when using a varied simulation. In the simulation-based inference literature, the most common proposal is to describe systematic uncertainties by adding nuisance parameters as inputs to the neural network that predicts the probability density. While parameterized neural networks might work in principle, the application would require simulated samples that are orders of magnitude larger than those currently available.

This proposal will develop a more viable approach for the description of systematic uncertainties by extending the factorization between parameter of interest and nuisance parameters commonly assumed in traditional analyses to the case of per-event analyses. In this approach, each systematic variation would be interpreted as an additional probability density ratio estimation. In total, a typical analysis would require the training and optimization of a few thousand neural networks, which requires accessible scalable infrastructure. But once the tools and infrastructure are available, every analysis that would benefit from per-event analysis could readily make a shift and become more powerful. This will be pursued as part of the program described in Section 3.4.1.

3.3.4 Reinforcement Learning

A further example of the enabling value of large scale services is in Reinforcement Learning (RL). HEP has a long history of using supervised learning, through boosted decision trees or other neural network techniques for data analysis. Unsupervised learning has also made inroads in HEP in recent years. However, in order to enable broader use and adoption of ML, we also need to enable the effective use of RL by making tools and frameworks available and easy to use at the large processing scales they demand. Reinforcement Learning can expand ML to many new use cases in HEP, since learning is not a separate step. PanDA with iDDS can be a powerful enabler of new scientific results through adoption of RL on large scale distributed platforms.

3.3.5 Managing User Code

An essential element of the geographically distributed ML services we propose is provisioning the user's software, environment, and ML models where the processing will take place. We plan to develop a simple and efficient mechanism to ship user codes to the cluster, grid, HPC and cloud resources where our services operate. Machine learning software varies a great deal from user to user and evolves rapidly, making a flexible and dynamically adaptable solution important. Our aim is to minimize the user's exposure to the particulars of the worker nodes they utilize, which from our long PanDA experience is a substantial challenge. Currently we are using containers to address this, but in the present implementation it remains difficult for users to update their codes. In further development we will improve the user's insulation from the containerization that encapsulates the remote environments, making it easier to transfer user codes smoothly from development to operational environments.

In addition, an easy to user interface for container management will be developed. Currently users need to create a new container when they have new code, and the work required to create a container is not familiar and simple for many physicists. We plan to use Jupyter+Container to simplify the interface, making it much more convenient for users to adapt their codes for remote running. Users in Jupyter (popular and familiar among physicists) will be asked to select a container image as a base for their codes. When submitting jobs, the user codes and the container tag information will be shipped to the PanDA+iDDS system. At remote worker nodes, a container corresponding to the tag will be started to run the user codes.

3.3.6 Interactivity and User Interface

Empowering users in creating large scale AI/ML applications requires strong support for interactivity. We will use Jupyter [23], popular across HEP and the AI/ML community, building on the Jupyter integration already implemented in PanDA/iDDS. We will also use Dask [24] to interactively provision scalable resources on capable platforms. The user will request an interactive session on a resource for a desired number of workers, and gain access via the Jupyter session when the resources are allocated. The Jupyter ML services interface together with the PanDA monitor UI can then be used to run, control, dynamically monitor and visualize workflows, and collect and assess results. For cross-site workflows collecting the results is a nontrivial requirement that is supported by our infrastructure. We will draw on community tools such as MLFlow [25] in our visualization and monitoring support.

3.4 TA4: Applying Large Scale ML Services in the Experiments

3.4.1 Practical per-event likelihood analysis using machine learning in ATLAS

Machine learning methods have become ubiquitous in experimental HEP, with ML methods used in simulation, triggering, reconstruction and analysis. Generative models have found a natural application in HEP simulations. Triggering has become the standard application in HEP for ML methods in embedded systems with ASICs and FPGAs. Even though simulation and triggering found their natural niche inside the ML community, there is a lot of unexplored potential applications in data analysis. The task of offline data analysis is to assign a probability to each event selected given a hypothesis being tested, whether it is the measured value of a fundamental constant, or the existence of a new particle. However, given the complexity of HEP experiments and HEP itself, it is impossible to obtain an analytical expression for these probability densities. All that is available in HEP are simulated events. Simulators in HEP are multi-stage programs involving first-principle calculations, phenomenological models, and detailed detector simulation. These simulations contain hundreds of latent parameters that are needed to describe the observed distributions.

These many latent variables are not used directly in data analysis; only the properties of the final-state particles are used. The main task of statistical data analysis in HEP is then to estimate the marginal probability of each event based on the reconstructed variables. The traditional method to estimate these probabilities is to search for a single observable, a summary statistic, make bins and assign a probability based on the frequency observed in simulation. This multinomial distribution approach has been used in almost every analysis in ATLAS to date, but it throws away a lot of information contained in the datasets collected by the experiment. The loss in information comes from the fact that the frequency of events in each bin can be very different from the actual probability of each event inside that bin. ML methods have been used extensively to mitigate this problem by creating classifiers that distinguish signal from background processes. Binned ML classifiers will naturally aggregate events that have similar probability in the same bin. Almost every result from ATLAS today uses a form of ML classifier to increase the sensitivity. Despite the gains obtained by using simple classifiers, this is still not everything ML methods can offer. When binning within the ML classifier score there is still some information loss; smaller than before, but binning always loses information. In addition, estimating the frequency of very rare events cannot be done precisely with the typical size of the simulation samples available. The binned approach is common because it provides an accurate, even if not always precise, calibration of the probability in every bin based on the simulated frequency of events. AI methods can do better than that since they can learn the actual probability densities and build the analysis event per event, extracting all the information available in the data collected.

These ideas have become known as “likelihood-free inference” or “simulation-based inference”. In the past, some analyses have tried to approximate the true probability density of each event by using complicated analytical functions. This method became known as matrix element method and has allowed for precise measurements of fundamental quantities such as the top-quark mass with very few events just after the discovery [26]. These analyses have been regarded as special cases because of the challenges in determining the analytical functions representing the per-event probability. ML methods can automate this process and provide much more accurate representations of the probability density. It would open a new era in HEP data analysis. Several ML methods have been proposed to perform simulation-based inference and create per-event probability densities. They can be especially powerful if used in conjunction with theoretical inputs about the hypothesis being tested [27]. However, to date, no analysis at the LHC has been performed with these methods because the proposals have only worked with toy examples. They have not addressed practical challenges that will be addressed in this proposal. The ML method can only be used to perform a real analysis if it predicts the probability density accurately. In the case of binned analyses, there is no problem to use a biased classifier since the probability is estimated from the frequency observed in simulation. In fact, most ML classifiers in traditional analyses aim to reduce variance, which tends to introduce biases on purpose.

A ML method used in a simulation-based inference should have not only low variance but also low bias. This is achieved by training very deep neural networks (NN) to reduce the bias and very large ensembles to reduce variance. Even for the simplest analyses, without a scalable ML infrastructure, developing and training these ML methods would be impossible. This infrastructure would also be necessary to verify the accuracy of the probability density estimation, which can be quite challenging given the multi-dimensionality of the probability density. ML methods can also provide these tests by training adversarial networks.

In order to develop the infrastructure needed to perform a simulation-based inference analysis, the off-shell Higgs boson measurement in the $H \to ZZ \to 4\ell$ final state has been chosen. This analysis has a simple final state but is severely limited by the small cross section for Higgs boson production away from the resonance peak. Performing this analysis with a per-event probability density can improve the result significantly. The reason why the cross section is so small, despite the resonant $2m_Z$ threshold, is the large destructive interference between signal and background. Destructive quantum interference has no direct probabilistic interpretation as an independent process, and therefore simple ML classifiers cannot be used in this case. A per-event probability density estimation, on the other hand, can easily handle quantum interference.

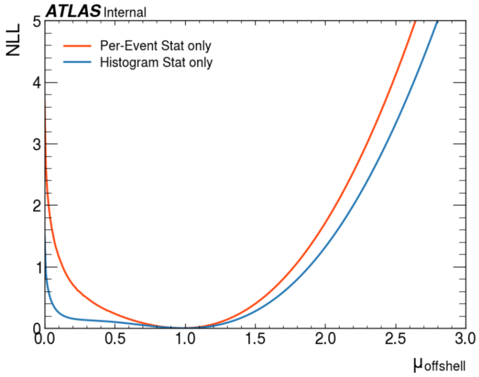

Figure 1: Comparison between the expected likelihood ratio from the off-shell Higgs measurement (orange) in the case of a per-event probability estimates and (blue) and the traditional binned approach. Both results use the same simulated data set equivalent to the integrated luminosity of Run 2 of the LHC.

A preliminary result, using a reduced simulation sample in a restricted phase-space, was obtained by performing the training of the NNs on a single GPU. An ensemble of 200 NNs was trained. Each individual NN was very deep and wide taking over 6 hours to train. The comparison of this preliminary exercise with the traditional analysis can be seen in Fig. 1. Especially for low values of the signal strength, the per-event analysis can be 3–4 times more powerful. When using all the simulated events available and extending the phase-space in the training, it is estimated that $\approx 1,000$ NNs will need to be trained, for an approximated total of 300 GPU-days. This development can only be performed in scalable ML systems. This proposal aims to develop the tools to make simulation-based inference analyses possible in real experiments. Preliminary tools for performing this kind of training in the GCP are being pursued.

Another big challenge is the description of the effect of systematic uncertainties on the probability densities. In binned analyses, systematic uncertainties are described by the change in the frequency of a bin when using a varied simulation. In the simulation-based inference literature, the most common proposal is to describe systematic uncertainties by adding nuisance parameters as inputs to the neural network that predicts the probability density. While parameterized neural networks might work in principle, the application would require simulated samples that are orders of magnitude larger than those currently available. This proposal will develop a more viable approach for the description of systematic uncertainties by extending the factorization between parameter of interest and nuisance parameters commonly assumed in traditional analyses to the case of per-event analyses. In this approach, each systematic variation would be interpreted as an additional probability density ratio estimation, that can be tackled by the same methods and tools described above. In total, a typical analysis would require the training and optimization of a few thousand neural networks, which requires a scalable infrastructure. But once the tools and infrastructure are available, every analysis that would benefit from this approach could readily shift and become more powerful.

During this project, the plan is to bring the off-shell Higgs boson analysis to completion as a demonstration of the tools that will be made public for this new approach. Other analyses that benefit from this approach may also be pursued at this early stage. The planned deliverables are described below:

1. Architecture to train per-event likelihood ratio estimates starting from standard fully simulated samples using the PanDA infrastructure.
2. Tools for assessment of accuracy and precision of per-event likelihood ratio estimates.
3. Tools for describing systematic uncertainties in per-event likelihood estimates.
4. Tools for hypothesis testing with per-event probability densities.

3.4.2 Large scale hyperparameter optimization for ATLAS and beyond

Machine learning applications often employ many hyperparameter dimensions, with O(10) typical today and the number growing over time as applications become increasingly sophisticated. Searching for an optimized hyperparameter set can be time and resource intensive, growing rapidly worse with the dimensionality, making it challenging and ultimately untenable to train models in sequential linear workflows, especially when the training time on a single model is long.

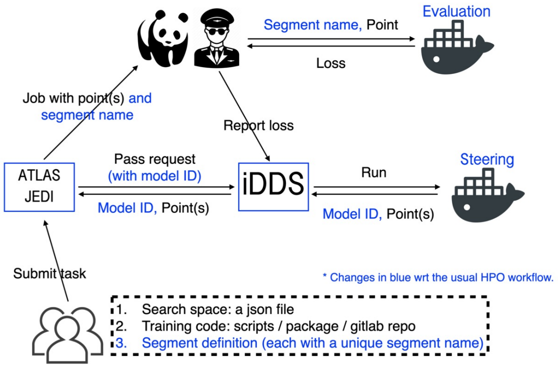
Segment name, Point
Evaluation
Loss
Job with point(s) and segment name
Report loss
Pass request (with model ID)
Run
Steering
ATLAS JEDI
iDDS
Model ID, Point(s)
Model ID, Point(s)
Submit task
* Changes in blue wrt the usual HPO workflow.
1. Search space: a json file
2. Training code: scripts / package / gitlab repo
3. Segment definition (each with a unique segment name)

Figure 2: The PanDA/iDDS based ATLAS HPO service

Hyperparameter optimization (HPO) can potentially improve the performance. For example from 2019, ATLAS has performed HPO at scale on their offline flavor-tag DL algorithms using Grid GPUs. The resulting algorithm provided 5(15)% higher c(light)-jet rejection for the same b-jet efficiency[18]. HPO has also been applied in many different analyses, such as [19–21], and has achieved large performance improvements. However even when using HPO the resource needs can be large. For example, a scan for the ATLAS ttH[19] analysis can require more than 1000 CPU cores for a day to get useful results. The required resources remain large even after shrinking the needs by constraining the hyperparameter search after some initial scanning, and repeated scans are needed for updated input data sets. The needs will be much greater for the HL-LHC, with an order of magnitude more data and more sophisticated machine learning models with more hyperparameters.

The solution we have pursued is to exploit parallelism both within a site and across sites to maximally leverage available resources and reduce the search time. A PanDA/iDDS based HPO service has been developed for ATLAS that provides a fully automated platform for hyperparameter optimization across geographically distributed cluster, grid, HPC and cloud resources, as shown in Figure 2. This service has been applied in the FastCaloGAN fast calorimeter simulation, now part of ATLAS's production fast simulation AtlFast3 [28], where it is used to accelerate optimization of the 300 generative adversarial networks (GANs) requiring 100 GPU-days for a single optimization pass.

We propose in this project to generalize and extend the HPO service as an experiment agnostic, community level service. This will involve work in areas previously described, including user code management (Section 3.3.5), user interface and visualization (Section 3.3.6).

3.4.3 ML in Integrated End-To-End Workflows: ATLAS Fast Chain

Within the ATLAS collaboration, considerable effort is being put towards increasing the rate at which simulated MC events can be produced during HL-LHC operation by developing fast alternatives to the algorithms used in the standard Monte Carlo (MC) production chain. The Fast Chain [29] has been designed to be flexible in combining fast and full simulation tools. It should not only meet the computational requirements but, first and foremost, produce accurate physics modeling. This optimization of fast methods, including Fast Track Simulation (FATRAS) [30], Fast Digitization, and Fast Reconstruction of pile-up tracks (track-overlay), can be done using decision roulette based on a ML approach. A simple approach would be to start from fast simulation and correct for residual differences, whenever the accuracy is not sufficient. This could be achieved by training on events produced by the standard MC and its fast alternative to learn the corresponding transformation function.

A more robust design would begin by identifying the phase-space characteristics that make a particular fast simulation approach accurate and prepare the inputs accordingly. An appropriate metric would need to be designed to label an event as well-modelled. Then, neural networks could be trained on all combinations of pairs of full-fast events, as well as on various different representations of physical processes. This would require extensive training, only possible with scalable ML infrastructure. Finally, one could use particle level information before the detector simulation and conditions during data-taking to predict if and which fast method could be used. An example of a physical process where ML can provide significant improvement is the modelling of nuclear interactions in FATRAS. The parameterization that is currently implemented in the MC is not able to reproduce exact details of the distributions and the use of generative models could provide a more accurate alternative. As part of this proposal, the scalable ML infrastructure will be applied to pursuing the robust strategy in order to quantify the gain that can be achieved in the rate of MC event simulation. The capability will be generalized and integrated into the ML infrastructure for use by others.

3.4.4 Applying Large Scale ML Services at Jefferson Lab

The participation of Jefferson Lab allows us to extend the application of this work to the nuclear physics (NP) program there, which offers use cases for large scale ML services. Involving NP scientists will contribute to broadening the usability and scope of our ML services, without overrreaching thanks to the close affinity of HEP and NP.

One of the NP facilities that is connected to the HEP science program is the Electron-Ion Collider (EIC) [31, 32]. The more comprehensive understanding of the strong interaction that will be gained at the EIC will allow to reduce systematic uncertainties of measurements at the LHC and neutrino experiments. In the last years the international EIC community has established the physics case and the resulting detector requirements for the EIC and is currently working on the first experiment [33]. The components of the EIC detector are being designed and integrated into a complete large scale detector system with the leading involvement of Jefferson Lab as one of the two host labs for the EIC (with BNL). The detector R&D requires simulations of the detector response with high precision and high accuracy.

As part of the project, PanDA will be deployed at Jefferson Lab and integrated in the workflows for designing and optimizing detectors for the EIC. For the detector simulations, the new eA Simulation Toolkit (eAST) initated by principals on this proposal (Diefenthaler, Wenaus) will be used that allows to combine fast and full simulations in one application [34]. The full simulations are based on Geant4. For the fast simulations, DNN models can be used that provide far more accurate results than parameterizations of the detector response [35, 36]. The DNN approach is used in HEP, e.g., Fast Chain at the ATLAS experiment. Individual detector components can be modeled in full detail and in a fast DNN model (trained from the detailed model), with eAST allowing the optimal model for a particular use case to be selected, subdetector by subdetector. Even more powerfully, and with yet greater processing demands, DNN modeling can extend across multiple subdetectors, for example in particle flow approaches.

The quality of the detector modeling depends on the architecture of the DNN, e.g., its depth, and the amount of training data. Using the large scale ML service will allow for exhaustive neural architecture search (NAS) and HPO for precise and accurate modeling. This will be demonstrated for the design and integration of two EIC detector components, an electromagnetic calorimeter and a dual radiator Ring Imaging Cherenkov (RICH) detector. For both detector components, models will be developed using the large scale ML service. A focus of the work will be on tests and validation of a distributed training capability that will allow detector experts to combine opportunistic computing resources, e.g., at Jefferson Lab and at their home institution for rapid turnaround of the studies of their detector component in an accurate simulation of the experiment. As a last step, the large scale ML service will be used for overall optimization of the detector and its parameters. The EIC community has started to use AI/ML for the optimization of the detector design [37]. The iterative workflow for the AI/ML supported optimization will be tested with the large scale ML service and validated for the optimization of the detector components as well as the whole experiment.

3.4.5 Applying Large Scale ML Services in the Intensity Frontier: DUNE and Belle II

Our BNL team includes collaborators on DUNE and Belle II who have identified ML applications within those experiments for which our scalable ML services can be valuable. As our project progresses we would aim to draw in more AI/ML applications and analyses that can benefit, from these and other experiments.

In DUNE, the BNL R&D project "Large Scale Scientific Simulation Systematics GAN" (LS4GAN) [38] uses generative adversarial networks (GANs) to train a model that translates simulated HEP data to real data, such that a given simulated event will then differ from its translated version in ways that can reveal real inaccuracies, artifacts, resolutions and biases in the simulation. The data volumes and neural networks involved are large, requiring large scale processing for training and inference that can benefit from large scale ML services. Current small-scale studies consume of order 16 GPU weeks for one pass; moving to realistic LArTPC images will be much more intensive.

In Belle II, flavor identification or “tagging" is often a crucial ingredient for CP violation measurements. One AI/ML based flavor tagging approach uses a multilayer perceptron (MLP) architecture with features sensitive to the kinematic attributes of the particles, such as momentum and polar angle. Work thus far has focused on B-tagging, involving an eight-layer MLP trained with 10M event samples on a single GPU, taking 48hrs [39]. A new charm meson application of the technique must contend with the shorter lifetime of charm mesons and corresponding lower separation power of MLP features. Considerably larger training samples and longer training times can be expected, making optimisation of the flavour tagger prohibitively expensive unless the analysts have access to substantially more computing resources.

3.5 Project Methodology and Execution

3.5.1 Impact on the HEP AI/ML Ecosystem

Through this proposal and project we aim to add to the HEP AI/ML ecosystem a transformative capability to scale their ML applications across geographically distributed resources and the full range of platform types to achieve otherwise unattainable processing scales, emphasising high usability and minimal intrusion upon a researcher's time. Scientists employing or evaluating AI/ML approaches today are constrained in their ambitions and scientific creativity by the resources available on individual accessible resources. We plan to develop ecosystem-level services available to the community to submit machine learning workflows across sites and across platform types. By so doing we will greatly expand the scaling reach available to researchers, and lower the often substantial barriers between developing applications in a small highly usable environment and moving them to large scale resources with the capacity to process the deep and complex models that may give the greatest scientific return.

In the preceding sections we have described the capabilities, software, external leverage and platform expertise we bring to bear; the suite of services and capabilities we plan to build; and the broad set of experiment applications that will be the seedbeds and testbeds for developing this new suite for the HEP AI/ML ecosystem. The center of gravity of project effort is in these experiment teams, primarily postdocs who will bridge experiment-grounded applications to the developing services, insuring the services we deliver meet the needs of real analysis, and receiving training at the leading edge of AI/ML as applied to our domain. We focus on domain science applications that will be empowered by our services, applications that will guide development, validate utility, and serve as examples and testbeds for community use.

3.5.2 Building Early Career Expertise

All activities of this proposal are designed to engage students, postdoctoral researchers and other early-career scientists. The individual and team activities described in detail in Sec. 3 will create the infrastructure for a more widespread applications of AI/ML tools and this project will only be successful if data-intensive researchers are trained with the skills to use them. Table 1 summarizes the goals for participation in training activities associated to this project. In total, we aim to engage more than 100 researchers in training activities, either as visitors to the participating institutions or in tutorial sessions.

Table 1: Summary of anticipated project participant goals and characteristics in terms of early-career (student, postdoc, junior faculty).

Program Description	Participants from this Proposal	External Early-Career Researchers
PanDA/iDDS core workshops (TA1)	15	10
Tutorials on ML Platforms (TA2-TA3)	5	50
Tutorials on Physics Applications (TA4)	5	50

Scheduling events and research opportunities Inclusiveness is of particular importance in managing the calendar of events and opportunities in an international setting. The workshops and tutorials will account for both US and European time zones and academic/holiday calendars of universities and laboratories. All tutorial events will be recorded and posted online so that participants in other time zones can also engage in the proposed activities. The events calendar will be actively managed by the lead PIs at BNL.

Virtual and in-person events The experience during the pandemic demonstrated that some activities can be held in hybrid mode. The in-person experience can be valuable for networking and for direct communication between the researchers involved, but offering workshops and tutorials virtually can provide broader reach and greater visibility for the proposed activities. During the pandemic, scientists reported far greater possibilities for participation in virtual events than in-person. Whenever possible, the activities will be hybrid in order to foster a more inclusive environment. The vast experience of the lead PIs organizing meetings will be used to broaden participation in our events.

Advertising events Events will be scheduled in advance and advertised via partner network channels. Given the collaborative nature of our science, these channels are already well developed in general. The lead PIs will track where advertisements have been made and actively seek additional channels. Feedback and assessment from previous events will be used to identify gaps and areas for further outreach.

Building in EDI All workshops and tutorials will follow guidelines for best practices from the EDI perspective, making sure that attendees represent the diversity of the collaboration, and that every decision includes a clear description of roles and expectations. The events will actively recruit members of underrepresented groups to ensure broad participation. Events will be organized in a way that promotes equitable practices including considerations about the make-up of the organizing committee for events, the content to collect on the registration form, the information provided on the website, the accessibility of the venue and event resources to support involvement.

Career development An important aspect of engaging with early-career researchers is understanding that many will pursue careers outside of academia. Past experience shows that research software developers and computing experts will have numerous career choices available. Participants in the activities of this proposal will have several opportunities to work with partners outside academia, including large scale companies such as Google and Amazon. The skills developed by students and early-career scientists will be invaluable in physics research and beyond, including not only applications in industry, but across scientific applications. This proposal will engage them with the broader data-science community.

Tracking success It is important to understand where early-career scientists that participate in the activities of this proposal go for their next position. The career paths of students, postdoctoral researches and early-career scientists that participate in this proposal will be documented. Longer term they may become contacts for new academia-industry partnerships and provide real-world feedback on necessary skill sets.

4 Deliverables

The high level deliverables constituting the end products of the project we propose are summarized here. A timeline of detailed deliverables is provided in the next Section.

1. As a contribution to the HEP AI/ML ecosystem, an experiment agnostic suite of ML Services (MLS) demonstrated to be highly scalable across geographically distributed resources that can support workflows running across facilities of a range of platform types including clusters, grids, clouds, HPCs and LCFs, with the infrastructure aware of and leveraging accelerator resources where available. The Year 3 scalability target is a 75k concurrency level for a single workflow, demonstrated on cloud and (allocations permitting) LCF resources.
2. An MLS component specifically tailored to provide integration of and convenient access to LCF resources, based on function as a service (FaaS) software, anticipated to be Parsl/funcX.
3. An interactive environment for MLS based on Jupyter and supported by PanDA/iDDS monitoring systems augmented for MLS, including support for pseudo-interactive processing on compatible resources (such as clouds and 'owned' clusters).

4. GitHub based open source repositories hosting the MLS software. The MLS software has as dependencies PanDA, iDDS and ancillary software (e.g. Harvester) all of which are also open source.
5. At least one operating, available, community level instance of the MLS system, conveniently accessible via cross-institution federated login. We anticipate the instance being located at BNL SDCC; additional instances will be welcome. The community instance will have the capability to use resources (subject to availability) on DOE HPCs including the LCFs, commercial clouds (Google, Amazon, and possibly NVIDIA), and a flexible array of cluster and grid resources.
6. Packaging and documentation for the MLS software, and its dependencies, supporting interested parties in bringing up their own instances of the service suite.
7. The MLS suite will include experiment agnostic support for highly scalable hyperparameter optimization, ML model training, hypothesis testing via per-event likelihood analysis and associated systematics evaluation, and reinforcement learning.
8. We will also deliver example testbeds for all the MLS service types, as well as generalized domain science examples in GAN based fast simulation, and an infrastructure for scaled, parallelized evaluation and optimization of complex end-to-end workflows utilizing ML (such as the ATLAS Fast Chain).
9. With the bulk of the supported effort going to postdocs developing testbeds and applications by bridging between experiment analysis groups and the core/MLS developers, a valuable end product will be a cadre of young researchers trained at the leading edge of challenging AI/ML applications in HEP.

5 Timeline and Personnel

The timeline of deliverables is summarized in Table 2. "MLS" refers to the ML Services infrastructure delivered by the project, which is progressively augmented in scope and capability as indicated by the deliverables. The timeline and deliverables have been scaled and scoped appropriately for the available effort, both project-supported (task areas of supported personnel are shown in Table 4) and off-project (the core PanDA/iDDS team). We do not attempt to resource load the timeline/deliverables at this stage.

Table 3 shows the project participants with their roles, and Table 4 shows project supported personnel with their task areas and roles. Some planned postdoc personnel can already be identified, and are named. This will be a great asset for a quick start with experienced people. Our timeline foresees completing the postdoc team during the first year.

For reference the task areas are

- TA1 - Core infrastructure
- TA2 - Platform support
- TA3 - Experiment agnostic large scale ML services (the MLS suite)
- TA4 - Applying large scale ML services in the experiments

Year 1	
TA1, TA3	MLS initial version implemented as experiment-agnostic generalization of ATLAS cross-site HPO service, and in use by non-ATLAS team members
TA1, TA3	MLS instance established at BNL SDCC for project internal use
TA1-TA4	Introductory workshop (Tutorials on PanDA/iDDS MLS, ML platforms; Physics app plans)
TA2	MLS platform support for clusters, grid, GCP, and HPC (NERSC)
TA2	Harvester implementation using Parsl/funcX to manage workflow execution
TA2, TA3	MLS scaling demonstrated at 10k concurrency level on cloud resources
TA2-TA4	Postdoc/grad student team 30% complete at month 1, 60% by month 6, 100% by month 9
TA3	MLS Jupyter interactive environment, CLI and UI for all supported workflows
TA3	User code management with simplified code transfer to execution environment
TA3	explicit deliverables for community tool support: tensorflow, horovod, mlflow, etc
TA3, TA4	MLS applied to train per-event likelihood ratio estimates using standard full simu samples
TA3, TA4	ATLAS use case of simulation-based inference implemented in MLS
TA3, TA4	Generic (experiment agnostic) HPO implemented in MLS
TA3, TA4	MLS integrated in the ATLAS FastChain workflow to test fast simulation strategies
Year 2	
TA1, TA3	Dask based interactive provisioning of scalable resources on clouds
TA1, TA3	ML model training platform extension to the MLS HPO service implemented
TA1-TA4	MLS applications workshop 1 (MLS services/ML platforms tutorials, physics applications)
TA2	MLS platform support extended to AWS and NVIDIA
TA2	MLS delivering user workloads to one LCF via Parsl/funcX
TA2, TA3	MLS scaling demonstrated at 30k concurrency level on cloud resources
TA3	MLS service instance at BNL SDCC access extended to early adopters
TA3	User code management extended with container management UI in Jupyter
TA3, TA4	Tools for assessment of accuracy and precision of per-event likelihood ratio estimates
TA3, TA4	ATLAS use case of evaluating systematic uncertainties implemented in MLS
TA3, TA4	Generic HPO support in MLS applied in non-ATLAS use case
TA3, TA4	Generic reinforcement learning use case implemented in MLS
TA3, TA4	MLS used to quantify gains from ATLAS FastChain simu strategies
Year 3	
TA1-TA4	MLS applications workshop 2 (MLS services/ML platforms tutorials, physics applications)
TA2	MLS on at least one LCF available to community, with ALCC allocation
TA2, TA3	MLS scaling demonstrated at 75k concurrency level across cloud and LCF resources
TA3	MLS service instance at BNL SDCC opened for general use
TA3, TA4	Tools for hypothesis testing with per-event probability densities
TA3, TA4	General formalism for describing systematic uncertainties in per-event likelihood estimates
TA3, TA4	MLS FastChain analysis generalized for quantified simu strategy evaluation

Table 2: Timeline of deliverables.

*Project Participants (**key in bold**)*	*Roles (**lead in bold**)*
Torre Wenaus (BNL Physics)	**PI**, oversight and coordination, postdoc supervision
Tadashi Maeno (BNL Physics)	**PanDA/iDDS** core and ML Services development
Alexei Klimentov (BNL Physics)	**Google Cloud Platform** integration and deployment
Paul Nilsson (BNL Physics)	**Interactivity** and PanDA core developer
Paul Laycock (BNL Physics)	Belle II liaison and use case studies
Brett Viren (BNL Physics)	DUNE liaison and use case studies
Meifeng Lin (BNL CSI)	**HPC/LCF integration**, postdoc supervision
Kaushik De (UT Arlington)	**Cloud Platform** integration and deployment, physics use cases
FaHui Lin (UT Arlington)	PanDA core developer
Fernando Barreiro Megino (UT Arlington)	PanDA core developer
Rafael Coelho Lopes de Sá (UMass Amherst)	**Application Support**
Verena Martínez Outschoorn (UMass Amherst)	**Training**
Rui Zhang (U Wisconsin Madison)	**HPO workflow** and development of visualization tools
Wen Guan (U Wisconsin Madison)	iDDS main developer
Markus Diefenthaler (JLab)	JLab lead, use case studies, postdoc supervision
Malachi Schram (JLab)	ML services development, postdoc supervision

Table 3: Roles of project participants.

Project Supported Personnel	*FTEs*	*Roles*
Torre Wenaus (BNL Physics NPPS Group Leader)	0.25	PI
Postdoc (BNL Physics)	0.5	TA3, TA4 - MLS for experiment use cases
Meifeng Lin (HPC Group Leader, BNL CSI)	0.1	TA2 - HPC/LCF support and FaaS integration
Postdoc (BNL CSI)	0.5	TA2 - HPC/LCF support and FaaS integration
Postdoc (UT Arlington)	1.0	TA3, TA4 - MLS for experiment use cases
Rui Zhang (U Wisconsin Madison)	0.2	TA3, TA4 - MLS workflow and app development
Postdoc (U Wisconsin Madison)	0.5	TA3, TA4 - MLS for experiment use cases
Grad student (U Wisconsin Madison)	0.5	TA3, TA4 - MLS for experiment use cases
Martina Javurkova (Postdoc, UMass Amherst)	1.0	TA4 - Develop analysis, fast simu tools with MLS
Evangelos Kourlitis (Postdoc, ANL)	1.0	TA2-TA4 - MLS use cases, ANL/ALCF facilities
Postdoc (JLab)	1.0	TA3, TA4 - MLS for experiment use cases

Table 4: Project supported personnel.

Appendix 1: Biographical Sketches

This appendix provides biographical sketches ordered with the PI first, then key personnel listed alphabetical by last name.

Appendix 2: Current and Pending Support

This appendix provides declarations of current and pending support ordered with the PI first, then key personnel listed alphabetical by last name.

Appendix 3: Bibliography and References Cited

References

[1] "US DOE Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee Report on AI/ML, Data-intensive Science and High-Performance Computing." https://science.osti.gov/-/media/ascr/ascac/pdf/meetings/202009/AI4Sci-ASCAC_202009.pdf, 2022.

[2] "The PanDA Production and Distributed Analysis Workload Management System." https://panda-wms.readthedocs.io.

[3] "The Intelligent Data Delivery System (iDDS)." https://idds.readthedocs.io.

[4] R. Coelho Lopes de Sa et al., "Off-shell Higgs Measurement Using a Per-Event Likelihood Method, ATLAS-Google Technical Meeting, Feb 16 2022." https://indico.cern.ch/event/1130033/contributions/4742590/attachments/2392761/4090662/ATLAS_Google_16thFeb22_mod.pdf.

[5] A. Castaneda, " Overview of ML and Big data tools at HEP experiments, 7th Edition of the Large Hadron Collider Physics Conferenc (LHCP 2019)." https://indico.cern.ch/event/687651/contributions/3427643/attachments/1849430/3036967/LHC2019_MLBigData_final.pdf.

[6] M. Barisits et al., "ATLAS Data Carousel," EPJ Web Conf., vol. 245, p. 04035, 2020.

[7] ATLAS Collaboration, "AtlFast3: The next generation of fast simulation in ATLAS," Comput. Softw. Big Sci., vol. 6, p. 7, 2022.

[8] C. Serfon, M. Barisits, T. Beermann, V. Garonne, L. Goossens, M. Lassnig, A. Nairz, and R. Vigne, "Rucio, the next-generation data management system in atlas," Nuclear and Particle Physics Proceedings, vol. 273-275, pp. 969–975, 2016. 37th International Conference on High Energy Physics (ICHEP).

[9] "The A Common Tracking Software (ACTS)." https://acts-project.github.io.

[10] P. Nilsson, A. Anisenkov, D. Benjamin, D. Drizhuk, W. Guan, M. Lassnig, D. Oleynik, P. Svirin, and T. T. Wegner, "The next generation PanDA Pilot for and beyond the ATLAS experiment," tech. rep., CERN, Geneva, Nov 2018.

[11] T. Maeno, F. H. Barreiro Megino, D. Benjamin, D. Cameron, J. T. Childers, K. De, A. De Salvo, A. Filipcic, J. Hover, F. Lin, and D. Oleynik, "Harvester : an edge service harvesting heterogeneous resources for ATLAS," tech. rep., CERN, Geneva, Nov 2018.

[12] F. H. Barreiro Megino, A. Alekseev, F. Berghaus, D. Cameron, K. De, A. Filipcic, I. Glushkov, F. Lin, T. Maeno, and N. Magini, "Managing the ATLAS Grid through Harvester," tech. rep., CERN, Geneva, Feb 2020.

[13] "funcX - Federated Function as a Service." https://funcx.readthedocs.io/en/latest/.

[14] Y. Babuji et al., "Parsl: Pervasive parallel programming in python," in 28th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC), 2019.

[15] "TensorFlow: An end-to-end open source machine learning platform." https://www.tensorflow.org.

[16] "PyTorch: An open source machine learning framework." https://pytorch.org/.

[17] "Horovod, a distributed deep learning training framework for TensorFlow, Keras, PyTorch, and Apache MXNet." https://horovod.readthedocs.io/en/stable/.

[18] "Hyper Parameter Scan with the Deep Learning Heavy Flavour Tagger (DL1)." https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/FTAG-2019-001/.

[19] ATLAS Collaboration, "Observation of Higgs boson production in association with a top quark pair at

the LHC with the ATLAS detector," Physics Letters B, vol. 784, pp. 173–191, 2018.

[20] ATLAS Collaboration, "Search for Higgs boson pair production in the $\gamma\gamma bb$ final state with 13 TeV pp collision data collected by the ATLAS," J. High Energ. Phys., vol. 2018, p. 40, 2018.

[21] ATLAS Collaboration, "Measurement of the Higgs boson mass in the $H \to ZZ \to 4\ell$ and $H \to \gamma\gamma$ channels with $\sqrt{s} = 13$ TeV pp collisions using the ATLAS detector," Physics Letters B, vol. 784, pp. 345–366, 2018.

[22] K. Cranmer, J. Brehmer, and G. Louppe, "The frontier of simulation-based inference," 2020.

[23] "JupyterLab: A Next-Generation Notebook Interface." https://jupyter.org/.

[24] M. Rocklin, "Dask: Parallel computation with blocked algorithms and task scheduling," in Proceedings of the 14th python in science conference, no. 130-136, Citeseer, 2015.

[25] A. Chen et al., "Developments in MLflow: A System to Accelerate the Machine Learning Lifecycle," in Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning, DEEM'20, (New York, NY, USA), Association for Computing Machinery, 2020.

[26] D0 Collaboration, "A precision measurement of the mass of the top quark," Nature, vol. 429, pp. 638–642, 2004.

[27] J. Brehmer et al., "A Guide to Constraining Effective Field Theories with Machine Learning," Phys. Rev. D, vol. 98, no. 5, p. 052004, 2018.

[28] ATLAS Collaboration, "AtlFast3: The Next Generation of Fast Simulation in ATLAS," Comput. Softw. Big Sci., vol. 6, p. 7, 2022.

[29] M. Javurkova et al., "The Fast Simulation Chain in the ATLAS Experiment." http://cds.cern.ch/record/2774052, Jun 2021.

[30] K. Edmonds et al., "The Fast ATLAS Track Simulation (FATRAS)." https://cds.cern.ch/record/1091969, Mar 2008.

[31] A. Accardi et al., "Electron Ion Collider: The Next QCD Frontier: Understanding the glue that binds us all," Eur. Phys. J. A, vol. 52, no. 9, p. 268, 2016.

[32] National Academies of Sciences, Engineering, and Medicine, An Assessment of U.S.-Based Electron-Ion Collider Science. Washington, DC: The National Academies Press, 2018. https://www.nap.edu/catalog/25171/an-assessment-of-us-based-electron-ion-collider-science.

[33] R. Abdul Khalek et al., "Science Requirements and Detector Concepts for the Electron-Ion Collider: EIC Yellow Report," Mar 2021. https://www.arxiv.org/abs/2103.05419.

[34] "eAST simulation toolkit." https://eic.github.io/east/.

[35] P. Bedaque et al., "A.I. for Nuclear Physics," Eur. Phys. J. A, vol. 57, no. 3, p. 100, 2021.

[36] A. Boehnlein et al., "Machine Learning in Nuclear Physics." https://arxiv.org/abs/2112.02309, Dec 2021.

[37] C. Fanelli et al., "AI-assisted Optimization of the ECCE Tracking System at the Electron Ion Collider." https://arxiv.org/abs/2205.09185, May 2022.

[38] D. Torbunov et al., "UVCGAN: UNet Vision Transformer cycle-consistent GAN for unpaired image-to-image translation." https://arxiv.org/abs/2203.02557, Mar 2022.

[39] F. Abudinén et al., "B-flavor tagging at Belle II," Eur. Phys. J. C, vol. 82, no. 4, p. 283, 2022.

Appendix 4: Facilities and Other Resources

Collaborative and Experimental Resources

Many of the principals on this proposal are active members of large participating collaborations (ATLAS, Belle II, DUNE) and other collaborations that can benefit in the future (Rubin Observatory, sPHENIX, EIC). The extensive computing resources for offline data processing, Monte Carlo production, and user analysis within these collaborations will be accessible to the participants in this proposal (see the attached letter from ATLAS SW&Computing Coordinator, Dr. Alessandro Di Girolamo). Development and testing of the new AI/ML software stack proposed here will be carried out at these collaborating facilities as well as the resources mentioned explicitly below. In addition, close collaboration with physicists and software developers within these collaborations, for example those working on core PanDA and iDDS, will be beneficial to this proposal.

Brookhaven National Laboratory

As the lead institution of the project, BNL has a wide range of computing facilities and resources to support the software and computing activities in the proposal.

Computational Science Initiative (CSI)

While long focused on timely analysis, interpretation, and overall management of high-volume, high-velocity heterogeneous data to pursue solutions for the national and international scientific community, the Computational Science Initiative (CSI) at the U.S. Department of Energy's (DOE) Brookhaven National Laboratory additionally excels at integrating computer science, applied mathematics, and computational science with broad domain science expertise to tackle problems and advance knowledge impacting scientific discovery. CSI's expertise and investments across the Brookhaven Lab, including its connectivity to flagship physics and materials science facilities that attract thousands of scientific users each year, are tackling the most pressing big data and science challenges. These efforts now are being augmented by CSI's growing high-performance computing (HPC), applied mathematics, and quantum science capabilities.

Scientific Data and Computing Center (SDCC)

Brookhaven's Scientific Data and Computing Center (SDCC) combines advanced expertise in high-throughput, high-performance, and data-intensive computing with data management and preservation in a centralized computing facility. It provides varied services to local and international clients that require stable, reliable computing capabilities for processing, storing, and analyzing large-scale data sets, along with HPC resources for increased computing power. The SDCC houses systems for high-performance and data-intensive computing, data storage, and networking, offering everything from novel research platforms to highly reliable production services. The facility stores 215 PB of data. In 2021, the SDCC processed 1.1 exabytes of data and transferred in and out more than 180 PB of data, ranking it among the top 10 data archives in the world.

The SDCC currently operates the institutional cluster (IC) at Brookhaven Lab. The IC clusters include:

Annie Cluster- 216 compute nodes with:

- HPE ProLiant XL190r Gen9
- Two CPUs Intel Xeon® CPU E5-2695 v4 @ 2.10 GHz

- 2x NVIDIA K80 (108 compute nodes) or 2x P100 (108 compute nodes) per node (4 K80 or two P100 devices per node)
- 256 GB Memory
- InfiniBand EDR connectivity
- Two submit nodes
- Two master nodes

Central storage for the IC is provided by IBM's Spectrum scale file system (GPFS), a high-performance, clustered file system that supports full POSIX semantics. • 1.9 TB of local disk storage per node • 1 PB of GPFS distributed storage • A GPFS-based storage system with a bandwidth of up to 24 GB/s is connected to the IC.

Francis (Knights Landing) Cluster- 142 compute nodes with

- KOI S7200AP
- One Intel Xeon Phi CPU 7230 @ 1.30 GHz
- NUMA node0 CPU(s): 0-255
- Thread(s) per core: Four
- Core(s) per socket: 64
- Socket(s): One
- NUMA node(s): One
- 192 GB Memory
- Dual Rail Omni-Path (Gen1) connectivity
- Two submit nodes
- Two master nodes

Skylake Cluster- The cluster consists of 64 compute nodes with:

- Dell PowerEdge R640
- Two CPUs Intel Xeon Gold 6150 CPU at 2.70 GHz
- NUMA node0 CPU(s): 0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34
- NUMA node1 CPU(s): 1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35
- Thread(s) per core: One
- Core(s) per socket: 18
- Socket(s): Two
- NUMA node(s): Two
- 192 GB Memory
- InfiniBand EDR connectivity

AI Cluster

The CSI AI Cluster is intended to process artificial intelligence (AI) workloads. It is equipped with eight NVIDIA V100 Tensor Cores. Five AI compute nodes are connected via EDR InfiniBand. Each node includes:

- HP Apollo 6500 Gen10 Server
- Eight NVIDIA V100 GPUs with 32 GB memory interconnected with NVLink
- Two Intel Xeon Gold 6248 (Cascade Lake) CPUs.
- Two NUMA nodes
- Sockets: 2

- Cores Per Socket: 20
- 768 GB memory
- Dual 10 GigE to Storage

Central Storage BNLBox is SDCC's cloud storage service that enables users to store their scientific data and documents locally, providing accessibility that is available from anywhere in the world via Internet access. The files stored in BNLBox can be shared with other BNLBox users, as well as external collaborators.

Computational Science Laboratory

The Computational Science Laboratory, also part of CSI, operates a collaborative laboratory for advanced algorithm development and optimization. The focus is to develop tools and techniques to solve problems in computational physics, biology, chemistry, materials science, and energy and environmental sciences by effectively using increasingly faster supercomputers.

HPC1 Cluster The HPC1 cluster consists of 21 nodes with 336 cores, each with 128 GB dynamic random-access memory (DRAM), connected with FDR InfiniBand. HPC1 Includes 12 NVIDIA GPUs and 8 Intel Phi coprocessors with 0.5 PB Lustre storage.

Field Programmable Gate Array (FPGA) Development Server CSI also has a few FPGA development servers with:

- 1 Intel Xeon CPU E5-2620 with eight cores
- 256 GB Memory
- One Intel Altera 10 GX FPGA card
- One NVIDIA Pascal 100 GPU
- Two NVIDIA Volta 100 GPUs

Computing for National Security

Part of Brookhaven's CSI, the Computing for National Security Department is at the crossroads of research conducted to understand how architectures and systems can tackle the challenges posed by data-intensive computing and its impacts on both scientific and national-security-motivated problems.

Advanced Computing Lab (ACL)

The ACL is being developed as a focal point and collaborative environment at Brookhaven for research and development (R&D) of architectures and systems, as well as for siting and enabling advanced computing testbeds. The ACL will consist of state-of-the-art equipment capable of facilitating R&D on computing technologies at different development and maturity stages, from materials to devices to subsystems to early full-system prototypes. High-resolution, accurate static and dynamic instrumentation for performance (time), power, and reliability will be included on specialized workbenches. The ACL will include machine room space properly designed for optimal operation and safe access to experimental testbeds. Network connectivity for external users will be provided as required. The ACL will include office space for technology providers and external collaborators.

NVIDIA DGX-2 Artificial Intelligence Cluster

A state-of-the-art NVIDIA DGX-2 is fully operational in the Advanced Computing Lab (ACL). The system is powered by two Intel Xeon Platinum CPUs and 16 fully interconnected Tesla® Tensor Core V100 graphics processing units (GPUs), each with 32 GB of HBM2 memory. Total system memory is 1.5 TB of DDR4. Storage is 30 TB of non-volatile storage class memory. The system is connected to Brookhaven's Scientific Data and Computing Center (SDCC) via dual 100 GigE networks. The system software consists of the original operating system, a "singularity" container, and Slurm scheduler. Brookhaven Lab was among the earliest adopters of the system, nicknamed Minerva.

NVIDIA DGX Station A100

ACL also has a new NVIDIA DGX A100 station, equipped with 4 NVIDIA A100 GPUs with 40 GB GPU memory each, and an AMD 7742 CPU with 64 cores.

Argonne National Laboratory

With the role of a participating institute to this proposal, ANL is capable of providing a significant part of the computing resources to support the proposed service. These resources are mainly supercomputers or computer clusters leveraging hardware acceleration for AI applications. The LCF providing the resources, along with the planned usage, is described below.

Argonne Leadership Computing Facility (ALCF)

The Argonne Leadership Computing Facility (ALCF), a U.S. DOE Office of Science User Facility located at Argonne National Laboratory, enables breakthroughs in science and engineering by providing supercomputing resources and expertise to the research community. Supported by the DOE's Advanced Scientific Computing Research (ASCR) program, the ALCF and its partner organization, the Oak Ridge Leadership Computing Facility, operate leadership-class supercomputers that are orders of magnitude more powerful than the systems typically used for open science. ALCF computing resources—available to researchers from academia, industry, and government agencies—support large-scale, computationally intensive projects aimed at solving some of the world's most complex and challenging scientific problems. Through awards of supercomputing time and support services, the ALCF enables its users to accelerate the pace of discovery and innovation across disciplines. The ALCF's team of computational scientists, data scientists, performance engineers, system administrators, software developers, visualization experts, and support staff help ensure users get the most out of the facility's high-performance computing systems. The ALCF also provides training and expertise to prepare researchers for the next generation of leadership computing resources.

The proposed service aims to harvest computational power for ML applications from hardware acceleration, in particular GPUs. Thus we focus on the ThetaGPU system of ALCF.

ThetaGPU The ThetaGPU supercomputer supports complex and diverse workloads involving simulation, data analytics, AI, and machine learning. ThetaGPU is an NVIDIA DGX A100-based system. It is comprised of 24 NVIDIA DGX A100 nodes. Each DGX A100 node comprises eight NVIDIA A100 Tensor Core GPUs and two AMD Rome CPUs that provide 22 with 320 GB of GPU memory and two nodes with 640 GB of GPU memory (8320 GB aggregately) of GPU memory for training AI datasets, while also enabling GPU-specific and -enhanced HPC applications for modeling and simulation. The ThetaGPU is integrated into a larger system, called Theta, via the ALCF's Cobalt HPC scheduler and shares access to a 10-petabyte Lustre filesystem. A 15-terabyte solid-state drive offers up to 25 gigabits per second in bandwidth. The dedicated compute fabric comprises 20 Mellanox QM9700 HDR200 40-port switches wired in a fat-tree topology. Table 5 summarizes the capabilities of ThetaGPU compute nodes.

Component	*per Node*	*Aggregate*
AMD Rome 64-core CPU	2	48
DDR4 Memory	1 TB on 320 GB & 2 TB on 640 GB	26 TB
NVIDIA A100 GPU	8	192
GPU Memory	22 nodes w/ 320 GB & 2 nodes w/ 640 GB	8,320 GB
HDR200 Compute Ports	8	192
HDR200 Storage Ports	2	48
100GbE Ports	2	48
3.84 TB Gen4 NVME drives	4	96

Table 5: ThetaGPU Compute Nodes Overview.

Jefferson Lab

Jefferson Lab operates an extensive computational infrastructure in support of its experimental and theoretical program. The Scientific Computing Department at Jefferson Lab works closely with the Experimental Nuclear Physics Division (Diefenthaler) and Data Science Department (Schram), whose members have onsite access to a range of equipment and technical support, including but not limited to staff workstations and software development tools; rack space, power and cooling for hardware; 298 node Intel/AMD compute cluster; 3 node AI/ML development cluster – 4x NVIDIA Titan-RTX GPUs per node; 3 node 40 GPU AI/ML cluster – NVIDIA T4 GPUs 16+16+8 configuration; Infiniband, OmniPath and Ethernet networking; disk managed storage for data and code; and robotic tape library for backup.

Google Cloud Platform

ATLAS is provisioning 15 months of computing resources, through unlimited-usage subscription pricing, on the Google Cloud Platform (GCP) for 2022 and 2023 as a pilot demonstration project at production scale. This is the continuation of an ATLAS-Google GCP R&D project which was provisioned during the previous three years, from 2019 to 2022. Two of the principals on this proposal, Klimentov and De, led the GCP R&D project. The pilot project will have average resources equivalent to an ATLAS computing site with 7000 cores and 7 PB of storage. Access to these resources in the GCP cloud will be available through the collaborating PIs in the proposal. The resources have an elastic spiking capability, able to grow to far more cores for a short period of time, that we will use for our ML services scaling tests. In addition, Google will provide training and deployment support for personnel in this proposal.

Amazon Web Services

US ATLAS has obtained substantial credit on Amazon Web Services (AWS) to demonstrate usability for physics analyzers. This is an active project led by De. These resources will be available to support ATLAS users if this proposal is funded. In addition, scientists and software developers involved in this proposal will benefit from the valuable experience in using AWS for the past few years.

NVIDIA

ATLAS is collaborating with NVIDIA as a part of the ExaTrkX R&D project. ATLAS and NVIDIA plan to increase their collaboration in the AI/ML domain in coming years. Discussions on future collaboration begin in June 2022. The collaboration will include access to a new ARM-based center at the CSCS Swiss National Supercomputing Centre in Switzerland, which contributes to ATLAS pledged resources. Two principals on this proposal, Klimentov and De, collaborated with CSCS as part of the DOE ASCR funded BigPanDA project.

Appendix 5: Equipment

This is an effort-driven project entailing software and algorithm development, testing and validation, using distributed computing resources as described in the proposal and Appendix 4. The project does not provision any physical hardware. Participating institutes with supported personnel are expected to equip their personnel with computers.

Appendix 6: Data Management Plan

In keeping with OSTI guidance on data management, we plan to manage our data in the most open manner allowed. We may make use of open data sets from the experiments which this proposal describes, primarily the products of official simulation campaigns (typically ROOT format) performed by the experiment collaboration. Such use is governed under the data management plans[1] of each experiment which we must and will honor. Every experiment with such data has at least one proposal team member who is a collaborator and thus we do not expect any impediments in honoring the experiment data management plans.

Data, Metadata and Results The data produced by the proposed work will consist of derivatives of experiment data (thus subject to the above) or come from simulations or performance measurements which this effort uniquely generates. This data will span a variety of scale and format and will be made publicly available utilizing appropriate storage and distribution mechanisms.

Data Access and Sharing We will leverage existing data systems at BNL and the other participating Laboratories, used in their intended conventional ways, to provide centralized storage and distribution points. Where feasible, and to keep with the guidance, we will also investigate ways to augment these centralized facilities with more publicly distributed mechanisms such as IPFS and "data repository" systems. Metadata, configuration data, software versions and source and documentation will also be provided in order to enable reproducing results.

Data in published results (figures, plots, etc) will be provided in a manner closely associated with the published document, such as embedded in the PDF. Where not feasible, these data will be provided as auxiliary files to the publication. We will in all cases include instructions on how to obtain the data from the centralized and/or publicly distributed systems described above.

Continuous Integration We expect a limited set of concise data files will be curated for use as input to ongoing automated code testing as run through the continuous integration system. As feasible, these files will be distributed alongside the source code, or where that is not feasible they will be available online from publicly accessible servers.

Software Preservation The long term data preservation requirements are modest. Our preservation will be in the form of code and configuration and the metadata needed to connect specific input data to results through code and configuration. We thus expect long-term preservation to rest with the code public repositories used by the project, namely GitHub. We will maintain a mirror to guard against the unlikely loss of the primary.

Confidential and Personal Data We do not expect any data used or produced by this project to contain information which is personally identifiable or otherwise sensitive of a nature which requires special protection.

[1] BNL Physics department maintains a centralized collection of these experiment data management plans at https://www.phy.bnl.gov/computing/index.php/Data_Management_Plans. This collection will be updated as needed for this proposal.

Appendix 7: Computational Resources

Computational resources needed to support the proposed research include CPU, GPU and (to a lesser degree) storage resources. As described in Appendix 4, a wide range of powerful national and international computing resources will be available to the project if funded, access made possible through the participation of major HEP laboratories and experiments, and experiment-industry collaborations.

BNL SDCC will host the large scale ML services (MLS) suite produced by the project, as a development platform early in the project and transitioning to an open community service by the project's completion. The resources directly required for service hosting are small, but the responsibility for hosting a secure, high availability, openly accessible service is large, and BNL SDCC has deep experience in this. BNL will in addition provide AI/ML processing resources, CPU and GPU, primarily via the resources assigned to the involved experiments (ATLAS, Belle II, DUNE). We will leverage the involvement of BNL's Computational Science Initiative (CSI) with its strong connections to ASCR and the full DOE supercomputing complex to leverage the complex for the HPC development the project will involve.

Jefferson Lab will also provide CPU and GPU processing resources via their EIC oriented participation in the proposed project. Both BNL and JLab host JupyterLab services that support their research communities; the project will be able to draw on these services for the deployment of its Jupyter-based interactivity services to the community. Both BNL and JLab have provided letters below indicating their willingness to collaborate should this proposal be funded.

The project will also benefit greatly from the involvement of ANL, by which we will gain access to the resources and expertise of in particular ALCF. Our effort plan includes support for an ANL postdoc who will be charged with serving as a bridge between the project's research and development program to integrate and leverage ANL computing resources including ALCF, in addition to participating in the development of experiment use cases and applications for large scale AI/ML services.

Our plan for DOE and other supercomputer access (such as CSCS in Switzerland) is to gain near term small scale access via our closely connected collaborators on the proposal (Lin, Kourlitis, Klimentov), possibly including discretionary allocations we will request (as described below for ALCF), and in the longer term (years two and three) formally apply for supercomputer allocations in support of the growing needs of the project. Given that the proposal collaborators have close involvement with multiple supercomputing facilities across the DOE and internationally, as well as guaranteed access to large scale commercial cloud resources than can also host the large-scale scaling exercises the project will require, we have many avenues for backup plans should any given facility not yield an allocation.

The storage resources required by the project are those needed to host the training datasets of the project's testbeds and applications. They will be provided via the resource allocations of the associated experiment. Apart from this the project proper will have negligible storage needs.

BNL Scientific Data and Computing Center (SDCC)

BNL SDCC has a variety of computing resources as detailed in Appendix 4, with new hybrid clusters expected in FY23. We plan to use SDCC to perform initial code prototyping for the HPC support and integration task, and will have access to the CPU, GPU and storage resources as well as software services provided by SDCC through an MOU (see the attached letter from Chair of SDCC, Eric Lancon). We expect the required computing resources for the initial development and testing will be minimal.

Argonne Leadership Computing Facility (ALCF)

To utilize ALCF ThetaGPU we plan to request allocation time via a major competitive LCF allocation program. As a preparation phase, in order to achieve computational readiness for the major allocation program, we plan to request "start-up" time via the ALCF Director's Discretionary Allocation Program (DDAP). This program is designed for such purpose and it will allow us to explore and develop capable, effort efficient mechanisms to integrate and leverage LCF HPC resources, as described in Section 3.1.2 and Section 3.2. The amount of requested time would be 10,000 GPU-hours (translating to about 1,300 node-hours) with an initial duration of the DDAP at 6 months. As a reference, the complete training of the ATLAS FastCaloGAN took about 2,500 GPU hours [28]. That allocation time will cover the relevant Year 1 (TA2) needs, as described in Table 2.

Towards the end of the 6 month period of the DDAP, we will be ready to apply for a major allocation award, such as DOE Innovative and Novel Computational Impact on Theory and Experiment program or ASCR Leadership Computing Challenge program (ALCC). We plan to scale up our request by a factor of 100 with a duration covering the remaining time of this proposal, until August 2025. With this larger award we will be able to focus on scaling as well as capability of the proposed service, as described in Section 3.3. Furthermore, we will extensively test the infrastructure and demonstrate its significance within this proposal via the use cases from experiments that are listed in Section 3.4. That allocation will cover computational needs of Year 2 and 3 (TA2-TA4), as described in Table 2.

Jefferson Lab

Jefferson Lab operates an extensive computational infrastructure in support of its NP program as described in Appendix 4. For the work on TA3 and TA4, in particular applying the large scale ML service to the experimental NP use case, we plan to use the resources at Jefferson Lab. We will have access to the CPU, GPU and storage resources as well as software services provided by the Scientific Computing Department through an MOU (see the attached letter from Chair of Graham Heyes). We expect the required computing resources for development, testing, and validation testing will be minimal.

European Organization for Nuclear Research
Organisation européenne pour la recherche nucléaire

Alessandro Di Girolamo,
CERN-IT
Computing Co-Coordinator, ATLAS Collaboration
40-3D-20
CERN CH 1211 Switzerland
Email Alessandro.Di.Girolamo@cern.ch

Jeremy Love
Office of High Energy Physics
Office of Science
Department of Energy
Washington, DC

Geneva, 23 May 2022

To Whom It May Concern:

This letter is to confirm that if the proposal submitted by Dr. Torre Wenaus entitled "Distributed Machine Learning Service for Data-Intensive Applications" is selected for funding, ATLAS members of the project team will have access to the appropriate computational resources (Grid, HPC and commercial clouds) as ATLAS R&D HL-LHC project according to the ATLAS Computing Shares Policy.

Yours sincerely,

Alessandro Di Girolamo,
ATLAS Computing Co-Coordinator
CERN Senior Staff

Computational Science Initiative

Brookhaven National Laboratory

2 Center St
P.O. Box 5000
Upton, NY 11973-5000

managed by Brookhaven Science Associates
for the U.S. Department of Energy

www.bnl.gov

May 17, 2022

Jeremy Love
Office of High Energy Physics
Office of Science
Department of Energy
Washington, DC

Dear Jeremy:

This letter is to confirm that if the proposal submitted by Dr. Torre Wenaus entitled “Distributed Machine Learning Service for Data-Intensive Applications” is selected for funding, the project team will have access to appropriate computational resources, subject to an MOU agreement with the Scientific Data and Computing Center (SDCC) at Brookhaven National Laboratory.

Sincerely,

Eric Lancon, Ph.D.
Chair, Scientific Data and Computing Center
Computational Science Initiative
Brookhaven National Laboratory
Upton, New York 11973-5000

Jefferson Lab
Thomas Jefferson National Accelerator Facility

May 20, 2022

Jeremy Love
Office of High Energy Physics
Office of Science
Department of Energy
Washington, DC

Dear Jeremy:

This letter is to confirm that if the proposal submitted by Dr Torre Wenaus, of BNL, and Dr Markus Diefenthaler, of TJNAF, entitled “Distributed Machine Learning Service for Data-Intensive Applications” is selected for funding, the project team will have access to appropriate computational resources, subject to an MOU agreement with the Computational Sciences and Technologies Division of TJNAF.

Sincerely,

WGHeyes

Dr Graham Heyes, Ph.D.
Scientific Computing Department Head
Computational Sciences and Technologies Division
Thomas Jefferson National Accelerator Facility
Newport News, Virginia.

heyes@jlab.org

Appendix 8: Recruitment and Retention of Students and Early-Stage Investigators

This proposal will fund several early-career scientists, mainly postdoctoral researchers, and will involve them in a variety of roles including developers of software and computing, designers of infrastructure for physics applications, and leaders in training courses.

Recruiting: Some early-career researchers that will initially contribute to the project are already identified. As they move on to their next position, new early-career researchers will be identified following practices that promote equity and diversity. All new positions will be publicized across many experimental collaborations well in advance, and the senior personnel will also actively recruit candidates. These practices are essential to ensure that candidates from underrepresented minorities are not at a disadvantage when compared to candidates that are already well connected to the several laboratories and participating universities. Each postdoc will have a primary mentor associated to their position (typically from the home institution) and but will also have close interaction with senior researchers at BNL, where the project is centered.

Mentoring: Effective mentoring by senior scientists is essential for early-career researchers to progress in their career, which requires increasing research expertise and recognition, and an understanding of how to seek out and prepare for available opportunities. All senior personnel on this project have considerable experience working with and mentoring postdocs, including postdocs focusing both on physics analysis and scientific software. The project senior personnel also have substantial experience working in large international teams. Postdocs and senior personnel would meet regularly to discuss professional development and career counseling topics, using resources such as "Making the Right Moves: A Practical Guide to Scientific Management for Postdocs and New Faculty" (Howard Hughes Medical Institute and the Burroughs Wellcome Fund, 2nd edition, 2006). Postdocs would also be able to mentor and work with students in research.

Retention: Postdoctoral researchers will be involved in software and computing development projects, as well as applications to the experiment's research activities. Activities tied to research lead most immediately to publications and are therefore given the most weight by institutions hiring junior faculty member. A balance between technical work and research activities is critical for the development of technical expertise and projects that lead to publications. It will help the postdoctoral researcher achieve high-impact results which will contribute to their success when applying for positions inside academia or in industry, while developing the infrastructure described in this proposal.

Transferable Skills: Not all early-career scientists will stay in academia. Training events and research projects will include the development of skills that are more broadly transferable, relying for example on the use of data science tools and techniques that have wide applicability. The technical expertise gained in software projects is particularly useful for career prospects both in industry and academia.

Leadership: Postdoctoral researchers involved in this proposal will be highly visible within the software and computing communities of the experimental collaborations involved. These projects will provide opportunities to improve presentation and team communication skills, as well as to broaden the academic social network. Early-career researchers will be given opportunities to present at international workshops and conferences, and will be encouraged to prepare short author list conference proceedings and journal papers reporting their results. Postdocs would also be given management and leadership opportunities in the context of the proposal's activities. We aim for the project to be a seedbed for future experts and leaders in AI/ML-empowered physics analysis and scientific software development.