**Program Outline:** Students participating in this traineeship program will be expected to take certain dedicated computer science courses, participate in training modules designed specifically for this program, engage in research projects with DOE laboratory scientists and engineers, and participate in a yearly summer program.

### Foundational Coursework

Each trainee will take some number of computer science and/or electrical engineering courses to build foundational computational knowledge. The specific courses taken will vary by university, by prior undergraduate background and will be dovetailed into the planned student's R&D project, but example courses and the expected coursework certification goals are:

**UW-Madison:** CS 319 Data science programming for research, CS 320 Data science programming II, CS 354 Machine Organization and programming, CS 506 Software Engineering, CS 552 Introduction to Computing Architecture, CS 750 Real time Computing Systems, CS 755 VLSI System Design, CS 759 High Performance Computing for Applications, and CS 784 Foundations of Data Management. Traineeship participants will be required to satisfy the UW Computational Minor (distributed) requirement [6]. In addition, an advanced course specific for computational physics "Topics in Computational Physics" is being developed to focus on topics specific to computational physics which will also constitute three credits and can be taken for partial fulfilment of the computational minor.

**Princeton:** COS226 Algorithms & Data Structures, COS320 Compiling Techniques, COS323 Computing and Optimization for the Physical & Social Sciences, COS342 Introduction to Graph Theory, COS/ECE 375 Computer Architecture & Organization, COS423 Theory of Algorithms, and COS 306/ ECE 206 Contemporary Logic Design. For the Physics PhD program participants will satisfy the requirements of the Graduate Certificate in Computational Science and Engineering [7]. Masters students will satisfy the course requirements of the CS/RSE masters program.

**UMass-Amherst:** ECE 665 Computer Algorithms, ECE 658 VLSI Design Principles, ECE 668 Computer Architecture, and COMPSCI 514 Algorithms for Data Science. UMass-Amherst students will complete the requirements for a Graduate Certificate in Statistical and Computational Data Science. The credits can also be used towards a master program in Physics.

It should be noted that the program is intentionally flexible to have the best match between the needs of particular area and that of the trainees. As a concrete example, a Wisconsin student planning a GPU project might form a computational minor by taking the three courses CS 354 Machine Organization and programming, CS 552 Introduction to Computing Architecture, and CS 759 High Performance Computing for Applications. While a Wisconsin student planning on analysis facility work might take CS 319 Data science programming for research, CS 506 Software Engineering, and CS 784 Foundations of Data Management. This flexibility will allow a specific path for individual students to best meet their training needs.

### Advanced and Practical Training modules

Training modules will be co-developed by the faculty and the laboratory scientists. These will include both lectures and coding labs. Hackathons will be organized as part of an annual summer program. Experts from the laboratory will be recruited to augment hands-on training sessions as needed. Industrial contacts with Nvidia and Xilinx also exist for arranging training sessions utilizing their software suites. All training materials will be accessible online such that students at each of the partnering institutes can access them. The lecture and lab components will be taught in a hybrid setting with students from all 3 universities participating either in-person or via Zoom. Elements of microlearning using an online platform will also be included and provide students with instructional videos and training quizzes that can be completed in short bursts of time.

**Training Module 0: Software Engineering for Scientific Computing:** This training module covers the core aspects of software development needed to write good quality, bug free and maintainable code. For this, the long-running "Software Engineering for Scientific Computing" course at Princeton [8] is planned to be reused and will be taught in the coming academic years by Henry Schreiner, a PhD particle physicist and research software engineer in the Princeton group. This course includes an overview of relevant compiled and interpreted languages, build tools and source managers, design patterns, design of interfaces, debugging and testing, profiling and improving performance, portability, and an introduction to parallel computing in both shared memory and distributed memory environments. The course content will be augmented by practical material specific to the experiments' software environments, e.g. Ed Moyse (UMass-Amherst) and J. Elmsheuser (BNL) will contribute for ATLAS.

**Training Module 1: Programming for heterogeneous architectures (GPUs):** The software & computing challenge of the HL-LHC era requires us to modernize software and design novel algorithms taking advantage of hardware accelerators (e.g. GPUs). This training module will introduce students to scientific computing on heterogeneous architectures and will discuss implications of heterogeneous hardware on the design and architecture of scientific software. Students will learn how to program for GPUs taking into account thread and memory organization and will assemble basic building blocks of a GPU program. They will learn how to debug and profile a GPU application and will be taught best practices. The development of this training module will be coordinated by T. Bose (UW) with help from postdoc Charis Koraka (UW) and M. Kortelainen (FNAL). Koraka has already acquired experience programming using CUDA (a software development toolkit used for programming on Nvidia GPUs) and is working with students to adapt the CMS electron/photon reconstruction code to run efficiently on GPUs. UW-Madison will provide curriculum development support with the majority being developed in Year 1 and smaller updates and improvements in Years 2-5. Connections to Nvidia experts will also be leveraged (see letter of collaboration, Tom Gibbs) to provide additional guidance and training for GPU programming.

**Training Module 2: Programming for heterogeneous architectures (FPGAs):** Modern High Level Synthesis (HLS) packages provided by FPGA vendors are beginning to be used extensively in algorithm development by physicists. For instance the Xilinx Vivado HLS is being used by Wisconsin, Princeton and Fermilab groups to build firmware for CMS experiment trigger and data acquisition systems. Wisconsin group pioneered development of core firmware in which the HLS based Intellectual Property (IP) can be integrated for a broad set of applications based on custom electronics boards sporting top-of-the-line FPGAs. There are several IPs integrated in to "bitfiles" for the Wisconsin CTP7 and APx series hardware. Knowledge also exists in the group to further develop firmware using these HLS techniques for the FPGA-based accelerators for the commodity computers as well. Existing training materials will be integrated, revised and augmented in particular to add Vivado environment and operational instructions. The module will begin with simple implementations exploiting massively parallel processing paradigm. Postdocs Pallabi Das, Rimsky Rojas Caballero and Varun Sharma, Faculty Sridhara Dasu and Verena Martinez Outschoorn will be coordinating this module. Nhan Tran (FNAL) will be a valuable resource in this development as his group has used similar techniques to build a generic framework called the HLS4ML [9].

**Training Module 3: Data Analysis Systems and Facilities:** Data analysis frameworks for HEP software are often built by the physicists. Currently several C++-based frameworks, often controlled by Python, are in use. The frameworks are also integrated closely with data formats used for archival, for example ROOT [10]. Newer frameworks in the Python ecosystem, e.g. Scikit-HEP [11], are also being developed which leverage developments in the larger data science ecosystem. HEP students & postdocs are often familiar with using these tools, but typically know little about how they work under the hood. This training module will review the evolving HEP analysis ecosystem, including the performance and I/O limitations and tradeoffs as well as newer strategies, e.g.,

columnar data analysis, and implications for designing modern analysis facilities. The goal of the module will be to prepare the students to develop scalable, performant and innovative tools within the ecosystem. P.Elmer (Princeton), K.Cranmer (UW), L.Gray (FNAL) and J.Elmsheuser (BNL) will coordinate development of this module, with curriculum development support at Princeton.

**Training Module 4: Scalable infrastructure:** Ready access to large scale resources greatly enhances the development and refinement of applications that require substantial processing, by shortening the development, optimization and validation latencies by orders of magnitude. The datasets in experimental HEP are large and complex, and they have similarly large processing demands, which will only increase in the future with the advent of the HL-LHC and DUNE, etc. This training module will familiarize participants with the tools for large scale distributed processing including diverse large scale computing resources, such as regional and global grids, HPCs and academic and commercial clouds, including new GPU workflows. Examples include the PanDA workload management system [12], the Intelligent Data Delivery Service (iDDS) [13], etc. Several example applications will be provided that are related to the research projects proposed. The goal is to train data-intensive researchers with the skills to use large scale computing resources. Rafael Coelho Lopes de Sá (UMass-Amherst) and Torre Wenaus (BNL) will coordinate the development of this module.

### R&D projects:

Each student will be involved in an R&D project to further enhance their training in computational HEP. These projects will leverage expertise both at the university and also at one of the labs. The different projects and their integration within the traineeship program are described in Section 5.

**Summer program:** Every summer, all of the trainees will meet for a two-week program that will include lectures, hands-on exercises and a hackathon. The site for the summer program will rotate each year amongst the three universities and the two labs (BNL and FNAL). The summer program will include components specifically targeting the preparation of students to be effective and impactful researchers, including exposure to broader tools and developments in the field of computational HEP, as well as other vital skills such as effective science communication and writing for publications and proposals (using for example materials in Ref. [14]). These components seek to build a cohort of students with the goal of increasing retention of a diverse group of graduate students. Events will be included to provide networking opportunities and develop professional connections (career panels, etc). Participants will also benefit from the connections between universities and national laboratories, and will be made aware of available opportunities in both, such as fellowships, postdoctoral positions, scientific computing positions, etc. These events will not only introduce students to HEP experiments and computational aspects in particular, but also introduce them to the national laboratories and also help form a community of students that spans several experiments.

## 5 Integrating the Traineeship Program with Existing R&D Efforts

As part of the proposed traineeship program, each student will get involved in an existing R&D effort in collaboration with laboratory personnel. Students will primarily be university-based, will interact with laboratory personnel via video-conferencing and messaging tools (e.g. Slack) and take advantage of the proximity to the labs to make trips as/when needed (e.g. extended stays during the summer). These trips will be funded via the FNAL LPC Guest & Visitor program, the US ATLAS ATC program or through the university. Example projects are described in this section.

### 5.1 Topical Area: High Performance Software and Algorithms

**Project: GPU-based reconstruction & workflow development**

*University mentors: T. Bose, S. Dasu, J. Olsen; Lab mentors: O. Gutsche, M. Kortelainen*

R&D topics include the development and porting of reconstruction algorithms to run on GPUs

(e.g. egamma reconstruction, tau reconstruction) for real-time CMS high level trigger and offline reconstruction; performing feasibility studies and evaluating performance of GPU-based algorithms; performing benchmarking studies of different workflows using different GPU configurations - both on-host and remote access; comparing network usage for different configurations and the development and integration of GPU workflows to run on High Performance Computing (HPC) sites. The efforts will leverage the existing GPU cluster at UW-Madison, the expertise of the local CMS Tier-2 team and postdoc Charis Koraka who is engaged in the development and integration of GPU-based production workflows for CMS. These R&D projects will collaborate with scientists & engineers at FNAL who will provide domain expertise and connections to other projects such as the DOE-funded HEP-CCE project. In addition to formal courses, Training Module 1 will be designed to provide the necessary training to the students for working on the projects listed above.

**Project: Future Collider Backgrounds Simulation on GPUs**
*University mentors: S. Dasu, K. Black, I. Ojalvo; Lab mentor: S. Jindariani*
Simulation of backgrounds due to beam induced backgrounds at future colliders is of immense interest in the design of the shielding and suitable detectors with optimal location and segmentation, for mitigating the beam induced background. This BIB simulation and mitigation program is especially important for the emerging muon collider option for a future O(10) TeV science case. Traditionally the FORTRAN based MARS program is used for simulations. Generation of even a few individual beam crossings requires days of processing time. Given the large backgrounds anticipated, rewriting using modern GPU-based parallel processing is highly desirable. Trainees will learn about simulation of both electromagnetic & hadronic processes when particles interact with matter. The technical expertise needed for the project includes parallel processing framework development with GPUs and the use of CUDA for developing GPU-optimized code. This project could be extended to speed up the MARS simulation used by the DUNE experiment (to model the radiation and heat deposition from the LBNF beam) and to port it to GPUs. Both formal courses and Training Modules 0 & 1 will be used to provide the necessary preparation.

**Project: Muon reconstruction for HL-LHC reconstruction and beyond**
*University mentors: S. Willocq, E. Moyse; Lab mentor: J. Elmsheuser*
The ACTS (A Common Tracking Software) software framework [15] is a complete set of track reconstruction software that does not depend explicitly on a particular experimental setup and is being developed as an open-source project by a team of active developers from a number of different projects. ATLAS plans to migrate all of its track reconstruction software to this new code base by the start of the HL-LHC. The UMass group is actively developing the foundation to test the difficult geometry of the ATLAS muon spectrometer and the navigation through this complex geometry for track reconstruction purposes. Several projects are foreseen: implementation of the geometry with the goal of enabling fast navigation for track reconstruction purposes (i.e. Tracking Geometry) and migration of the data preparation chain, including formation of drift circles for MDTs, clusters for RPCs, (s)TGCs, etc. This is a natural area since the UMass group has primary responsibility for the muon software EDM. Implementation of pattern recognition at chamber level (finding and fitting track segments) and at muon spectrometer level. Training Modules 0 and 1 will provide a key introduction for this project.

**Project: Data reduction for Dark Energy Science Collaboration (DESC)**
*University mentor: K. Bechtol; Lab mentor: A. Drlica-Wagner*
Another project in this area uses synthetic source injection at survey scale for the Rubin Observatory LSST DESC. Synthetic source injection is used to calibrate the sensitivity/response of astronomical imaging surveys such as LSST. Synthetic source injection involves injecting synthetic astronomical objects into on-sky pixel-level images and re-running the data reduction pipeline.

**Project: GPU algorithms for DUNE**

*University mentor: B. Rebel; Lab mentor: A. Norman*

One project that falls in this category relates to the measurements of neutrino oscillation parameters. These parameters are subject to several physical boundary conditions, for example the squared sine of the mixing angles cannot lie outside the range 0 to 1, and to correctly report confidence intervals related to these parameters requires a computationally intensive frequentist strategy. These corrections require enormous CPU resources to perform. The project would explore the use of GPUs and multithreaded algorithms to improve the efficiency of determining the correct confidence intervals.

## 5.2 Topical Area: Collaborative Software Infrastructure

**Project: Services and Tools for Analysis Facilities for the HL-LHC**

*University mentors: T. Bose, V. Martinez Outschoorn; Lab mentors: L. Gray, J. Elmsheuser*

Innovative facilities are being designed to meet the needs of users as LHC datasets grow in complexity and by order-magnitude in size over the next few years. New services will be needed for data reduction, effective data manipulation in common ecosystems and other analysis tools. Projects in this area include: developing new services and tools for common analysis frameworks for example those based on columnar analysis within the python ecosystem; performing benchmarking, scalability and usability studies. Students will also work with Dr. Brian Bockelman (Morgridge Institute at UW and co-PI of IRIS-HEP) and contribute to data analysis challenges. Bockelman's support letter is appended. These projects will build upon formal coursework and Training Module 3.

**Project: Tools for DESC analysis operations**

*University mentor: K. Bechtol; Lab mentor: A. Drilica-Wagner*

A similar project to the one above is also proposed for the Rubin Observatory LSST Dark Energy Science Collaboration (DESC). The project involves developing tooling to reliably and efficiently produce sets of value-added data products and distribute. These operations involve touching almost all rows of tabular data containing billions of rows, and also efficiently querying subsets of these data. The operations should be reproducible and should be able to run in multiple environments.

**Project: Future Collider Parallel Processing Framework**

*University mentors: S. Dasu, T. Bose; Lab mentor: L. Gray*

An opportunity exists to design a parallel processing framework from scratch for future colliders, leveraging columnar data organization from the very early stages of data acquisition. Presently event builders are used to aggregate the data from the detector, package it and store in a single raw-data custom blob per event. Organized event reconstruction reads the full raw-event data, processes it and stores its output in several new tiers. The analyzers are often restricted to small fraction of "reconstructed" data in the smallest "nano-AOD" tier. Any custom processing by physicists of the lowest tier raw data will always need to read the full raw-event blob from large, often archived, files. Unpacking this data to select a small fragment of interest is a prohibitive operation. Innovative algorithms using raw data tiers are therefore discouraged. Columnar organization from the start, tagged by the event time stamp, may enable much faster (highly parallel) processing. This project will propose a data organization scheme and create a framework for parallel data processing with relatively easy access to any tier of data. The columnar raw data organization may also enable efficient reconstruction algorithms in highly parallel heterogeneous environments. Project will build upon Training modules 3 & 4.

**Project: Elastic Computing Infrastructure**

*University mentors: R. Coelho Lopes de Sá, V. Martinez Outschoorn; Lab mentor: T. Wenaus*

The Computer Science department at UMass-Amherst is developing a project in collaboration with Boston University (BU) called OCT/ORCI with the goal to develop a Elastic Secure Infrastructure (ESI) for computer clusters. The goal of the OCT/ORCI project is to create shared facilities

where the resources can be shared between different projects. The project is developed in the Massachusetts Green Computing Center in Holyoke/MA, the premier computing center used by several large universities in the state (UMass-Amherst, BU, MIT, Harvard, and Northeastern). The software is developed in collaboration with RedHat. MGHPCC houses the ATLAS Tier 2 Cluster NET2, a collaboration between UMass-Amherst and BU. This proposal would allow NET2 to be the first large-scale application of the OCT/ORCI project. The trainees project would demonstrate this technology: install, configure and test OpenStack/Ironic on testbed NET2 nodes; configure the testbed nodes for NET2 production via static VLAN; test provisioning and use of testbed nodes as part of NET2. NET2 would be the first WLCG cluster operating in an ESI environment, strongly benefiting from the collaborative projects in the MGHPCC and shared operation of the resources. Project will build upon Training module 4.

### 5.3 Topical Area: Hardware-Software Co-design

**Project: Custom Algorithm IP Development for FPGAs**

*University mentors: S. Dasu, V. Martinez Outschoorn, S. Willocq; Lab mentor: N. Tran*

Custom FPGA boards often sport 100s of multi-gigabit transceivers, e.g., the APx boards built by Wisconsin have over 120 25-Gbps capable transceivers. The APx group has also produced a firmware framework shell that encapsulates the details of handling these streams of data and make available to inner algorithm core in a straight-forward arrays. Such arrays of data can be processed using C++ code instrumented with appropriate Vivado HLS pragma directives in an efficient way with very low latency at speeds of 360 MHz. Development of C++ code and selection of directives is crucial to fit within the limitations of the FPGA resources and the allowed latency. Several projects are already underway for LHC Run-3 in CMS. Opportunities for several IP developments exist for trigger and data acquisition system upgrades for Run-4 and beyond. Training Module 2 and hackathons will prepare trainees for developing IPs for some of the core CMS upgrade projects.

Another project aiming at the development of future trigger systems using Vivado HLS is a collaboration between UMass-Amherst Physics, EE, and Computer Science departments. Students and engineers are implementing firmware logic blocks to carry out portions of the muon track reconstruction inside the overall project firmware. Several projects are foreseen targeting muon track reconstruction applications: reconstruction of combined Micromegas and sTGC track segments, combining the two detector technologies present in the ATLAS New Small Wheel, and standalone reconstruction of Monitored Drift Tube track-segments including timing information.

## 6 Program Process

The overall program organization will be steered by a program committee that has one faculty member representing each of the three universities. Additionally, Dr. Julie Yun (Associate Dean for Diversity and Inclusion, School of Engineering and Applied Sciences, Princeton), and Dr. Vanessa Gonzalez-Perez (Assistant Dean for Diversity Initiatives in the Natural Sciences, Graduate School, Princeton), and Devon Wilson (Associate Dean for Diversity, Equity and Inclusion, College of Letters and Science at UW Madison ) will advise the team with regard to advertising and recruiting in order to attract a diverse cohort of trainees. They will make recommendations for shared practices among all three participating universities to ensure coherent messaging intended to reach, and appeal to, students from all backgrounds, including groups that have been historically excluded or marginalized in STEM fields. Dr. Wilson, Dr. Yun, and Dr. Gonzalez-Perez are committed to equity and access among graduate students, and have a track record of success improving support mechanisms for graduate students and trainees in STEM fields. The program committee, chaired by T. Bose (UW Madison) with members J. Olsen (Princeton) and S. Willocq (UMass-Amherst), will meet bi-weekly to oversee the training program taking into account the following components:

**Advertisement:** The traineeship program will be widely advertised at each of the three universities during graduate student recruitment events ("open-houses"), orientation sessions at the start of

Tables 4- 5 show an example 2-year schedule for a student starting with the proposed program typically in their 2nd year of graduate school after having taken core courses and after successfully passing their qualifying exam. As part of this training program, students will take the necessary advanced courses and obtain research credits (for the training modules) that help them satisfy the relevant requirements at each of the universities (e.g. the distributed minor requirement at UW-Madison). Additionally, students will give an end-of-semester presentation in front of all the student trainees and submit a paper-style report that will form part of their dissertation write-up.

| Semester 1 | Semester 2 | Summer Session 1 |
|---|---|---|
| Advanced CS Course 1 | Advanced CS Course 3 | TM 3 and TM 4 |
| Advanced CS Course 2 | TM 1 and TM 2 | R&D Project |
| TM 0 | Intro to R&D Project | |

Table 4: Year 1 schedule for a typical student trainee showing the sequence of advanced Computer Science (CS) courses, Training Modules (TM) and R&D project. TM1-4 have 1/2-course credit.

| Semester 3 | Semester 4 | Summer Session 2 |
|---|---|---|
| Special Topics/HEP Course | Special Topics/HEP Course | R&D Project |
| R&D Project | R&D Project | |

Table 5: Year 2 schedule for a typical student trainee including special topics courses in advanced software and computing as well as particle physics.