November 2026

Volume 17 Issue 2

JoCsE

Journal Of Computational Science Education

Promoting the Use of Computational Science Through Education

ISSN 2153-4136 (online)

Contents

Volume 17, Issue 2

Introduction to Volume 17 Issue 2 1
David Joiner, Editor

The Computational and Data Science Exchange (CDSExchange): A Model for Building Faculty Capacity in Computational and Data Science Education 2
Katharine Cahill, Karen Rios-Soto, Tandabany Dinadayalane, Rachel Vincent-Finley, Linda Akli, Ahlam Tannouri, Zhuoying Wang, and Lisa Garbrecht

Adapting for Adoption: Tailoring HPC Training for Resource-Constrained Communities 6
Bryan Johnston, James Carson, Charlie Dey, Jennifer M. Schopf , Christopher J.Whalen, and Grace Kebirungi

Introducing HPC-Enabled Computational Research Training in Africa: A Pilot Study 10
James Afful, Titus Nyarko Nde, Onyeka Onyenemezu, Brian Kyanjo , Benjamin Atribawuni Asaaga, and Joseph Oppong Akowuah

Expanding the National AI Research Resource: Building Capacity and Community through Sustainable Research Pathways 20
Mary Ann Leung, Connie L. McNeely, and Danielle Jacques

T^3-CIDERS Train-the-Trainers: Strengthening CI Skills for Cybersecurity Research and Education through Iterative Improvements in Second Cohort 26
Wirawan Purwanto, Mohan Yang, Peng Jiang, Dorothy X. Parry, Belle Li, Kristin Herman, Nolan Lovett, Shanan Chappell Moots, Masha Sosonkina, and Hongyi Wu

In Search of a Better Search: Using RAG on Cornell Virtual Workshop Content 34
Zilu Wang, Susan Mehringer, and Christopher J. Cameron

A Documentation Catalog for the AI Ecosystem: Design of the ICICLE AI Training Catalog 39
Amit Vyas, Carlos Guzman, Neelima Savardekar, Mary Thomas, Michelle Miller, Alfonzo Morales, Joe Stubbs, Christian Garcia, and Beth Plale

Understanding Computational Science and Engineering (CSE) and Domain Science Skills Development in National Laboratory Postgraduate Internships 46
Morgan Fong, Hilary Egen, Marc Day, Kristin Potter, and Michael J. Martin

Leveraging HPC in University Courses: A Cross-Disciplinary Workforce Development Model 57
F. Alex Feltus, Jon C. Calhoun, Rong Ge, Jill Gemmill, Carlos Toxtli Hernandez, Bing Li, Todd Allen Price, Justin Talbot, and Minjae Woo

Low-Cost Embedded Computer Vision and Neural Network-Enabled Autonomous Navigation for a Four-Wheeled Mobile Robot Using an ESP32 and Arduino Framework 65
Abhiram Kaakarla

Introduction to Volume 17, Issue 2

David Joiner, Editor
Kean University
djoiner@kean.edu

Issue 17-2 of the Journal of Computational Science Education features nine articles, including six papers presented at the SEHET26 Workshop on Strategies for Enhancing HPC Education and Training at PEARC26 in Minneapolis, MN. A common theme of the SEHET26 papers is building lasting capacity for computational science and HPC, both by reaching underserved communities and by making existing training materials easier to find.

Cahill et al. describe the CDS-Exchange, a faculty community of practice helping lesser-resourced institutions launch computational and data science offerings. Johnston et al. discuss adapting TACC's HPC training for African sites receiving repurposed Stampede2 hardware. Leung et al. present the Sustainable Research Pathways program as a model for expanding access to the NSF National AI Research Resource (NAIRR). Purwanto et al. describe how evaluation of the first cohort of T3-CIDERS, a train-the-trainer program in CI skills for cybersecurity, shaped the redesign of its second cohort. Wang et al. present a retrieval-augmented generation chatbot for searching Cornell Virtual Workshop content, and Vyas et al. describe the design of the ICICLE AI Training Catalog.

This issue also includes three additional articles. Fong et al. evaluate postgraduate computational science internships at the National Laboratory of the Rockies, finding the largest gains in technical rather than non-technical skills. Feltus et al. describe how faculty across many disciplines at Clemson University integrated the Palmetto2 supercomputer into their courses. Finally, JOCSE 17-2 features a student paper by Kaakarla, a high school student who built a low-cost autonomous robot using embedded neural networks and computer vision on an ESP32 microcontroller.

We encourage you to submit your work to the Journal of Computational Science Education. Computational science is an increasingly important interdisciplinary field, offering insights into complex systems, accelerating discovery, and helping to solve diverse problems. We welcome high-quality papers describing instructional materials, successful projects, or research on instructional efficacy. Whether you are faculty or a student, your contributions are valuable to advancing computational science education. Additionally, if you have expertise in computational science, consider volunteering as a reviewer to support our peer review process. Together, we can share successes and inspire others to develop and adopt computational science in education!

Sincerely, Dave Joiner

The Computational and Data Science Exchange (CDS-Exchange): A Model for Building Faculty Capacity in Computational and Data Science Education

Katharine Cahill
New Jersey Institute of Technology
kate.cahill@njit.edu

Karen Rios-Soto
University of Puerto Rico Mayaguez
karen.rios3@upr.edu

Tandabany Dinadayalane
Clark Atlanta University
dtandabany@clark.edu

Rachel Vincent-Finley
Southern University Baton Rouge
rachel.finley@sus.edu

Linda Akli
Texas Advanced Computing Center
lakli@tacc.utexas.edu

Ahlam Tannouri
Morgan State University
ahlam.tannouri@morgan.edu

Zhuoying Wang
Texas Advanced Computing Center
zwang@tacc.utexas.edu

Lisa Garbrecht
Texas Advanced Computing Center
lgarbrecht@tacc.utexas.edu

ABSTRACT

Small and lesser-resourced institutions often face challenges launching or expanding undergraduate Computational and Data Science (CDS) course offerings. CDS-Exchange addresses this challenge by building a faculty community of practice across institutions. Core components of the project include: webinar series, regional workshops, fellows program, and annual symposium. Diverse activities result in broad outcomes, such as multi-institution participation, strong repeat engagement, and positive self-reported gains in understanding and teaching relevance.

KEYWORDS

Computational and Data Science Education, Faculty Community of Practice, Collective Impact, Capacity Building

1 INTRODUCTION

The demand for workforce expertise in Computational and Data Sciences (CDS) continues to grow rapidly [2, 3, 5, 6]. However, institutions with limited local capacity often struggle to develop and implement related curricula, such as new minors or certificates. Because CDS inherently requires the integration of mathematics, computer science, and domain-specific science and engineering, creating comprehensive academic programs poses significant logistical challenges. Universities are traditionally organized into discipline-oriented departments with siloed budget models, creating a structural obstacle to rapid interdisciplinary implementation. Furthermore, existing curriculum structures exacerbate this issue: upper-level courses are typically tailored for discipline majors and restricted by numerous prerequisites, while introductory courses frequently lack computational modules and remain slow to adapt. Consequently, newly established CDS programs face substantial hurdles in recruiting students due to rigid schedule constraints and complex prerequisite pathways.

To address these pervasive challenges, the CDS-Exchange was established as a National Science Foundation (NSF)-supported initiative aimed at expanding access to CDS education through shared instructional resources and collaborative faculty development. Led by the New Jersey Institute of Technology (NJIT) in partnership with the Texas Advanced Computing Center (TACC), the Univer-sity of Puerto Rico at Mayagüez (UPRM), Clark Atlanta Univer-sity (CAU), Morgan State University, and Southern University, the project leverages inter-institutional cooperation to build capacity where local resources are constrained. This paper provides a con-cise overview of the CDS-Exchange model and details its major activities during the first two project years. By highlighting emerg-ing evidence of the project's early impact, this work emphasizes practical lessons and effective strategies that are highly relevant to the broader high-performance computing (HPC) and computational education communities.

2 PROJECT MODEL

CDS-Exchange is a faculty-focused project that uses several modes of engagement to connect faculty to a broader community of CDS educators. By encouraging development and sharing of the CDS curriculum to add to existing courses, we intend to have a greater impact on students' exposure to these topics. Rather than relying solely on the creation of entirely new, resource-intensive academic programs, the model encourages the collaborative development and sharing of CDS curriculum modules that can be integrated into existing courses, thereby maximizing student exposure to these topics. This approach evolved from the initial C^{2}Exchange pilot project (2018–2022), where a foundational group of project members successfully collaborated to share materials and expertise [1].

A critical insight gained from the pilot program is the diverse utility of shared resources and the importance of professional community support. Exchange materials proved highly adaptable, successfully utilized as independent study projects, integrated as specific modules within existing classes, or adopted entirely as new courses in department catalogs. Furthermore, the pilot highlighted

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/1

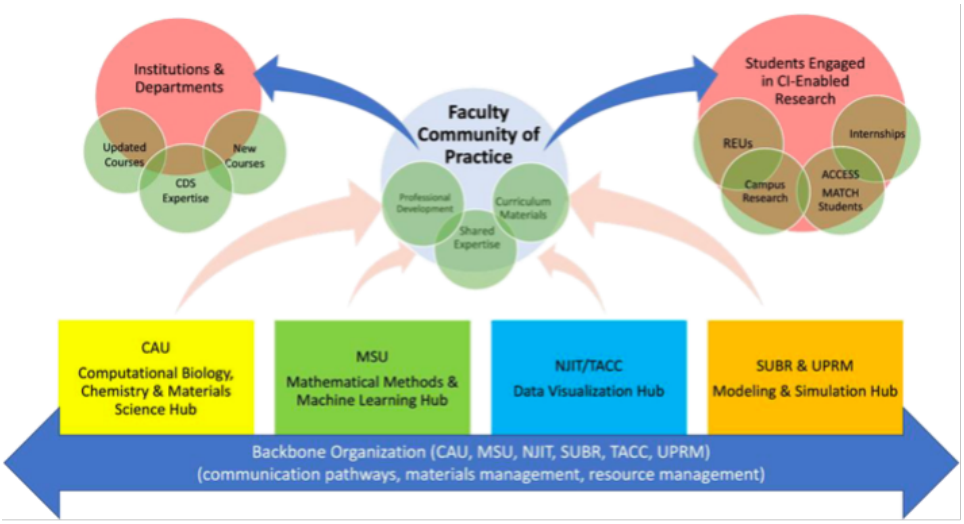


Figure 1: Structure and goals of the CDS-Exchange Project based on the Collective Impact model

Table 1: List of Regional Workshops

Institution	Date	Topic
UPRM	December 2024	Modeling & Simulation
UPRM	May 2025	Statistical Techniques
UPRM	December 2025	Chemistry & Biology
SUBR	June 2025	Modeling & Simulation
Morgan State	May 2026	Modules for Pre-Calculus

that isolated faculty members–who are often the sole computationally focused instructors in their departments–greatly benefit from the professional development and peer network that the exchange provides. Ultimately, this cross-institutional sharing model accelerates the implementation of CDS curricula, strengthens university partnerships, and ensures faculty are well-supported in preparing a larger cohort of students for a workforce where these specialized skills are in high demand.

2.1 Goals

Goals of CDS-Exchange:

- Share existing courses and modules with a larger faculty community
- Create new materials that fill gaps in CDS curriculum and are relevant to this community
- Facilitate course or module development by the community
- Provide opportunities for professional development and networking for faculty in CDS
- Encourage faculty research collaboration

The goals of the CDS-Exchange project were designed in accord with the National Science Foundation's (NSF) model of collective impact, which focuses on building "collaborative infrastructure" to achieve systemic change [4]. These goals work collectively to enhance Computational and Data Sciences (CDS) education through shared vision, partnerships, leadership, and scalability. Specifically, the goals of sharing existing courses and creating new materials to fill curriculum gaps provide a platform for collaborative action, effectively engaging a larger faculty community in a shared vision for CDS education. Furthermore, providing continuous opportunities for professional development and networking naturally fosters distributed leadership and increases communication among CDS educators. Ultimately, our focus on facilitating ongoing, community-driven course and module development establishes the necessary capacity for the expansion, sustainability, and scale of CDS curricula across participating institutions.

2.2 Activities

These are the project's four core mechanisms:

- Webinars: Building Bridges in CDS Curriculum Webinar series 2024 & 2025 recurring online sessions on topics such as computational thinking, data visualization, modeling and simulation, computational chemistry, computational biology and bioinformatics, and data storytelling.
- Regional workshops: in-person, hands-on events tailored to local communities such as Puerto Rico and Southern University campuses.
- Community Development Fellows: a cohort model supporting faculty curriculum development, with the first cohort focused on data visualization integration. The second cohort is focused on scientific computing tools.
- Annual symposium: a national convening that connects webinar participants, fellows, and regional workshop attendees around curriculum development, educational tools, and institutional implementation.

Table 1 provides an overview of the regional workshops offered so far. We have completed one full year of the fellows program, and are in the middle of the second fellows year with a new cohort. We've hosted two symposia held at Clark Atlanta University, the 1st symposium in May 2025 had 25 registered participants, and the 2nd symposium in April 2026 had 53 registered participants.

2.3 Infrastructure and Community Building

To ensure long-term sustainability and foster continuous collaboration, the CDS-Exchange project relies on a robust project infrastructure designed to transform individual activities into a cohesive, ongoing community. This infrastructure integrates a centralized website, active mailing lists, and shared repositories that allow faculty to seamlessly access, utilize, and contribute curriculum materials. Additionally, archived webinar recordings provide asynchronous professional development, while regional information networks facilitate localized support and peer engagement. By maintaining these interconnected communication and resource platforms, the project successfully moves away from a model of isolated events, instead providing a persistent digital and professional ecosystem that supports CDS educators year-round.

3 EMERGING OUTCOMES

3.1 Reach

ar series included 75 participants from 41 institutions, including HBCUs, high-Hispanic-enrollment institutions, a Tribal university, a community college, and a non-U.S. institution. In the second reporting year, webinar participation remained strong, with 57 attendees across the year, 18 non-R1 institutions represented, 16 R1 institutions represented, and continued engagement from nonprofit academic organizations and industry.

3.2 Engagement

The evaluation data demonstrates that the CDS-Exchange project model operates effectively as a community-building strategy by fostering repeat participation and cross-event movement among faculty. Between October 2024 and May 2025, a series of five targeted webinars accumulated a total of 91 participants while maintaining a steady 44% average evaluation survey response rate. Highlighting this community traction, 62 percent of webinar participants in the first year attended more than one session, a trend of repeat engagement that persisted into the second year alongside notable cross-event progression as several workshop participants subsequently attended the annual symposium. These patterns reinforce a key project insight: truly sustained faculty development requires multiple, interconnected modes of engagement rather than fragmented, one-off events.

This multi-layered engagement strategy has successfully cultivated deep cohort dedication and a robust pipeline for future participation. Within the program, fellowship cohort members demonstrated high ongoing involvement, with 91% of respondents agreeing or strongly agreeing (36% agree, 55% strongly agree) that they actively engaged in most of the project's scheduled activities throughout the year. Simultaneously, the project's regional training workshops generated a strong demand pipeline for the formal Fellows program, capturing the interest of 100% of Winter UPRM attendees, 82% of Spring UPRM attendees, and 100% of Summer SUBR attendees. This strong "community of practice pull" is further evidenced by a widespread desire to remain embedded in the project's broader ecosystem, with 100% of all regional workshop participants and 88% of webinar respondents expressing explicit interest in active, long-term participation within the wider CDS-Exchange network.

3.3 Perceived Value

Webinar respondents (88%) expressed interest in joining a community of practice, and two-thirds or more across five webinars reported increased understanding of the content. Regional workshop respondents also reported high gains in knowledge and skills for using CDS-Exchange materials, and all respondents to the 2025 fellows program and annual symposium surveys rated those experiences as valuable and/or useful. The fellows program was reported to be highly impactful among participating faculty. Every surveyed fellow reported overall satisfaction with their experience, while all concurrently affirmed that their personal investment of time and resources into the program was entirely worthwhile. This exceptional level of satisfaction is deeply rooted in the program's success in driving instructional empowerment and capacity building. Specifically, 100% of participants noted that the provided resources offered significant, direct support to their teaching (80% strongly agree, 20% agree). Furthermore, the program effectively fostered long-term educational growth by ensuring that 100% of the fellows agreed that the initiative successfully strengthened their individual instructional capacity while simultaneously elevating their home institution's core capability to deliver high-quality Computational and Data Sciences (CDS) education (70% strongly agree, 30% agree).

4 LESSONS LEARNED

Based on the first two years of survey data, we have identified several lessons learned.

- Multiple entry points matter. Webinars create a low-barrier first contact, while workshops, fellows activities, and the symposium create progressively deeper engagement.
- Regional adaptation matters. UPRM workshops show how the same project model can be tailored to local disciplinary interests and regional institutional networks, including Puerto Rico and the Caribbean context.
- Community infrastructure matters. Mailing lists, archived recordings, websites, and shared curriculum materials help convert short-term events into an ongoing educator network.
- Cross-disciplinary framing matters. Participation from mathematics, biology, chemistry, engineering, statistics, business, and public administration suggests that CDS/HPC education grows when framed as broadly applicable rather than limited to computer science alone.

5 REPRODUCIBILITY AND NEXT STEPS

The CDS-Exchange model demonstrates high reproducibility and transferability due to the exceptional clarity of its operational processes and structural design. Because its core elements are clearly defined, the framework can be systematically adapted by other academic communities or disciplines seeking to build collaborative educational networks. These structured components include recurring webinars for broad outreach, local workshop partnerships for localized faculty training, cohort-based faculty support to foster continuous mentorship, an annual national convening for community alignment, and shared online resources to ensure open access to curriculum materials. This clear, modular blueprint demystifies the complexities of cross-institutional collaboration, making it a

highly transferable strategy for expanding specialized educational capacities in resource-constrained environments.

Looking ahead, the project's next steps will focus on sustaining and expanding this structured process through continued webinars, additional regional workshops, and the recruitment of a third fellows cohort. Furthermore, the initiative plans the continued strengthening of the curriculum repository and project partnerships to deepen inter-institutional cooperation and secure long-term resource sustainability. Ultimately, the CDS-Exchange experience suggests that a layered model combining online engagement, regional faculty development, cohort-based support, and national convening can broaden participation in CDS and HPC education while building a sustainable community of practice.

REFERENCES

[1] Katharine Cahill, Linda Akli, Tandabany Dinadayalane, Ana Gonzalez, Raphael Isokpehi, Asamoah Nkwanta, Rachel Vincent-Finley, Lorna Rivera, and Ahlam Tannouri. 2022. Building a computational and data science workforce. *The Journal of Computational Science Education* 13, 1 (2022), 27–31. doi:10.22369/issn.2153-4136/13/1/5

[2] Burcin Campbell. 2026. Tracking the rise of AI academic programs: Methods, metrics, and what the data tell us. *CERP Bulletin* April 28 (2026).

[3] Andrea Danyluk and Scott Buck. 2019. Artificial intelligence competencies for data science undergraduate curricula. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33(01). 9746–9747. doi:0.1609/aaai.v33i01.33019746

[4] National Science Foundation. 2022. NSF's Eddie Bernice Johnson INCLUDES initiative. https://nsf-gov-resources.nsf.gov/files/INCLUDESOverviewOnePager-r.pdf

[5] National Center for Science and Engineering Statistics. 2021. *Women, minorities, and persons with disabilities in science and engineering: 2021.* Special Report NSF 21-321.

[6] U.S. Bureau of Labor Statistics. 2026. Occupational outlook handbook. https://www.bls.gov/ooh/

Adapting for Adoption: Tailoring HPC Training for Resource-Constrained Communities

Bryan Johnston
University of Cape Town
Inter-University Institute for Data Intensive Astronomy
bryan@idia.ac.za

James Carson
TACC, USA
jcarson@tacc.utexas.edu

Charlie Dey
TACC, USA
charlie@tacc.utexas.edu

Jennifer M. Schopf
TACC, USA
jms@tacc.utexas.edu

Christopher J Whalen
RDCT, USA
cwhalen@researchdata.us

Grace Kebirungi
IDI, Uganda
gkebirungi@idi.co.ug

ABSTRACT

Successful High-Performance Computing (HPC) adoption in resource constrained environments requires more than hardware deployment; it requires training models that are explicitly designed for local realities rather than transplanted from well-resourced contexts. This position paper argues that leadership-class HPC curricula must be re-engineered into pragmatic, context-aware training pathways if emerging HPC communities - particularly in Africa - are to achieve sustainable operational capability. Drawing on the adaptation of the Texas Advanced Computing Center's (TACC)[1] leadership-class training into an onboarding program for African HPC sites, we illustrate how global best practices can be reframed into actionable modules aligned with infrastructural, staffing, and operational constraints. We contend that such contextualisation is not optional but foundational for long-term HPC sustainability, and we outline principles for designing training that empowers institutions to deploy and operate HPC systems sustainably and effectively within resource-limited environments.

KEYWORDS

HPC training, Resource-Constrained Environments, Leadership-Class Expertise, African HPC, Workforce Development

1 INTRODUCTION

High-Performance Computing (HPC) adoption in emerging or resource-constrained environments (RCEs) cannot rely on hardware redistribution alone. Sustainable operation requires training models that translate leadership-class best practices into locally applicable skills. Yet, most global HPC training materials are designed for mature, well-resourced centres and assume the presence of stable infrastructure, adequate staffing, and access to established communities of practice. When transplanted directly into RCEs, these models fail to address the operational realities that shape sustainability and effective implementation of advanced research computing.

The Texas Advanced Computing Center (TACC) has long extended the life and impact of flagship HPC systems through its Legacy Computing Program[2], which loans retired hardware to broaden access to advanced computing. The HPC Ecosystems Project in Africa complements this hardware pipeline by focusing on repurposing the hardware, training the workforce, capacity building, and long-term operational support for emerging HPC sites [5]. Together, these initiatives highlight a central challenge: access to hardware does not automatically translate into operational capability.

This paper argues that leadership-class HPC training must be re-engineered for RCEs through pragmatic, context-aware onboarding frameworks. Preparing African recipients of Stampede2[3] hardware required a substantial redesign of TACC's existing curriculum, originally developed for experienced administrators in mature US-centric environments. The adapted program addressed operational challenges such as limited staffing, intermittent connectivity, and isolated technical communities, while also providing practical guidance for deploying and operating the Xeon Phi Knights Landing (KNL)[4] architecture within modern software stacks [3].

Position Statement

We contend that contextualised training is foundational - not optional - for meaningful HPC adoption in RCEs. By reframing leadership-class concepts into practical modules aligned with local constraints, we outline principles for designing training that supports both immediate onboarding and long-term institutional growth.

Defining Leadership-Class Training

Leadership-class, in the context of this position paper, refers to training, operational practices, and architectural expertise developed by an experienced large-scale, production-grade HPC centre in a mature computing environment. This denotes a depth of expertise and operational maturity. Leadership-class training, therefore, encompasses advanced system design principles, policy frameworks, and operational strategies used by established HPC centres with long-standing experience, stable infrastructure, and well-developed user communities.

[1] https://tacc.utexas.edu/

[2] TACC - https://tacc.utexas.edu/partnerships/legacy-computing-program/
[3] Stampede2 (Top500.org) – https://www.top500.org/system/179045/
[4] Intel Product Finder - https://www.intel.com/content/www/us/en/ark.html

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/2

2 AN OVERVIEW OF TRAINING RESOURCES GAPS

Although extensive documentation and training materials exist globally for HPC system deployment[5], they overwhelmingly reflect the assumptions of established centres in the Global North. These resources presume reliable power supply, deep layers of staffing, mature user communities, and ready access to peer support - conditions that rarely exist in resource-constrained environments. As a result, the materials do not address the realities faced by emerging HPC sites, where both technical staff and users may be encountering HPC for the first time.

The OpenHPC Virtual Lab is an example of hands-on training that accounts for resource constraints [4]. It provides scalable, accessible system administrator training without requiring advanced infrastructure. However, its focus remains primarily technical and does not address higher-level concerns such as governance, policy design, or sustainability planning. Most global training materials share this limitation: they emphasise operations but overlook the broader institutional ecosystem required for long-term success.

TACC's leadership-class curriculum illustrates this gap. It was developed for administrators already embedded in established HPC environments, whereas implementers of legacy hardware in RCEs often operate in isolation, with limited staffing, limited expertise, and no established user community. Effective training for these sites must therefore extend beyond technical instruction to include leadership principles, policy frameworks, and strategies for cultivating a sustainable HPC culture.

To address these gaps, we combined TACC's leadership-class training expertise with more than a decade of on-the-ground experience from the HPC Ecosystems Project in deploying HPC systems to RCEs in Africa. This integration enabled the development of training materials that retain the strategic depth of leadership-class instruction while grounding it in the operational realities of emerging African HPC sites. The resulting framework, described in the next section, re-engineers leadership-class concepts into pragmatic, locally tuned onboarding for RCEs.

3 TRAINING DESIGN

Early lessons from the HPC Ecosystems Project revealed a structural challenge: distributing decommissioned hardware does not automatically create operational capability. This is particularly true for architectures such as Intel's deprecated Xeon Phi Knights Landing (KNL), whose memory modes, NUMA behaviour, and performance characteristics require nuanced operational knowledge that is no longer widely documented. Without targeted onboarding, sites receiving such systems face a steep learning curve. This can undermine the intended impact of hardware distribution.

TACC's long history of delivering leadership-class training - through programs such as the Summer Institute[6] and advanced operations workshops - provides a strong foundation of strategic and architectural expertise [1, 2]. However, these curricula assume conditions typical of established HPC ecosystems. These assumptions do not hold in most RCEs, where administrators may be operating an HPC system for the first time, and often in isolation.

To bridge this gap, we adapted TACC's leadership-class curriculum into a practical onboarding program tailored specifically for RCEs. The goal was not to simplify the material, but to translate high-level system design principles into usable guidance aligned with local constraints. This required re-engineering the training around three core principles:

- **Contextualisation over transplantation** — leadership-class concepts must be reframed for environments with intermittent power, constrained staffing, and limited access to vendor or community support.
- **Operational pragmatism over architectural idealism** — training must prioritise reliability, reproducibility, and sustainability within resource-constrained environments rather than cutting-edge optimisation.
- **Capacity building over instruction** — administrators must be empowered not only to operate the system, but to teach users, enforce policy, and cultivate a local HPC culture.

The resulting program retained the strategic depth of the original TACC 5-day workshop ("Designing and Administering Large-scale Systems")[7] while restructuring it around the needs of RCEs:

- **Original Curriculum:** An advanced 5-day workshop for leadership-class HPC operations.
- **Adaptation Goal:** Translate high-level design principles into practical, context-aware guidance.
- **Audience:** System administrators with Linux and networking experience but limited exposure to operational HPC environments.
- **Outcome:** Participants gained the ability to design, deploy, and operate Stampede2 KNL systems with production-grade reliability while embedding strategic operational policies to ensure the integration of sustainability and user-support principles into daily operations.

4 PRINCIPLES FOR CONTEXT-AWARE HPC TRAINING

The adaptations described above reveal a set of recurring design considerations that extend beyond this specific training program. These patterns form a generalisable framework for developing HPC training that is operationally realistic, contextually-aware, and aligned with the constraints of RCEs. The following principles distill the lessons learned from adapting leadership-class training for RCEs and provides a foundation for designing future training models that support sustainable adoption:

- Design for operational reality, not idealised assumptions.
- Embed policy into onboarding to reduce support burden.
- Prioritise reproducibility and documentation over optimisation.
- Translate architectural concepts into actionable heuristics.
- Build local capacity by training administrators to train users.
- Plan for sustainability from the outset, even when growth is uncertain.

[5] Example 1: sighpceducation.hosting.acm.org/resources/hpcresources/;
Example 2: sighpceducation.hosting.acm.org/resources/systems/

[6] Summer Institute - https://www.rdworldonline.com/a-new-paradigm-for-training-in-advanced-computing/

[7] Archived advert from 2020 - https://web.archive.org/web/20200925225118/https:/www.tacc.utexas.edu/education/institutes/large-scale-systems

Table 1: Training Program Overview

Day	Content
Day 1	**Foundations of scientific system architecture:** capability vs. capacity computing, workload bottlenecks, and understanding how policy functions as an architectural mechanism.
Day 2	**User education, policy, and HPC user experience:** onboarding as policy enforcement, effective help-request practices, and reducing support load through structured user education.
Day 3	**Slurm, KNL, and hardware-aware scheduling:** configuring NUMA, MCDRAM, and scheduler policies for real-world KNL workloads.
Day 4	**Designing cluster policy:** fairshare, quotas, and policy-encoded scheduling rules to ensure equitable access and reduce long-term administrative overhead.
Days 5–7	**Containers, data lifecycle, security, and sustainability:** reproducible science, data-lifecycle policies, security practices, and sustainable workflows to reduce administrator burnout.
Day 8	**Capstone hackathon:** integrating architecture, policy, storage, and monitoring into a demonstrable HPC environment, reinforcing capacity building through hands-on synthesis.

5 AGENDA OVERVIEW (ADAPTED FOR RCES)

The adapted curriculum preserved the strategic depth of the original leadership-class workshop while restructuring each day around the operational realities of RCEs. The agenda emphasised actionable skills, policy-driven operations, and reproducible practices. The agenda reflected in Table 1 operationalises the principles of contextualisation, pragmatism, and capacity building: demonstrating how leadership-class concepts can be reframed for emerging HPC communities.

6 KEY ADAPTATIONS

Adapting a leadership-class curriculum for resource-constrained environments required more than simplifying technical content; it demanded a fundamental reframing of what effective HPC training looks like outside mature ecosystems. The adapted program shifted emphasis from architectural design toward pragmatic operational strategies that reflect the realities of African HPC sites. User experience was treated as a system-design problem, embedding policy directly into onboarding so that administrators could reduce support load and enforce good practices from the outset. Advanced hardware topics (such as KNL memory modes, NUMA behaviour, and MCDRAM configuration) were distilled into actionable scheduling heuristics that complemented the KNL architectural theory. The training prioritised community documentation and reproducibility over cutting-edge optimisation, recognising that sustainability in RCEs depends on clarity, consistency, and shared operational knowledge. Finally, capacity building was embedded as a core objective: administrators were trained not only to operate the system but to teach users, cultivate local expertise, and seed a self-sustaining HPC culture.

Summary of Key Adaptations

- Shifted emphasis from leadership-class design to practicalities for resource-constrained environments.
- Reframed user experience as a system design problem, embedding policy into onboarding.
- Simplified advanced hardware topics (KNL, NUMA, MCDRAM) into practical scheduling practices.
- Prioritised community documentation and reproducibility over optimisation.
- Embedded capacity-building goals: empowering admins to teach users, not just operate systems.

These adaptations reflect the principles outlined in Section 3 and demonstrate how leadership-class training can be effectively re-engineered to meet the realities of RCEs.

7 OUTCOMES

The adapted training model produced outcomes that reinforce the central argument of this position paper: context-aware training is essential for sustainable HPC adoption in resource-constrained environments. Participants reported increased confidence in deploying and managing Stampede2 systems, particularly in navigating KNL-specific operational challenges that are poorly documented in global training materials nowadays. By formalising troubleshooting paradigms and reframing user onboarding as a policy mechanism, the program reduced friction between administrators and users - a critically important factor in environments where staffing is limited and support burdens can quickly become overwhelming. Perhaps most significantly, the training seeded a community of practice rooted in the long-standing experience of the HPC Ecosystems Project, enabling sites to share operational knowledge, documentation, and lived experience despite limited vendor support. These preliminary outcomes suggest that leadership-class training, when re-engineered for local realities, has the potential to empower emerging HPC centres to operate legacy systems sustainably and contribute meaningfully to regional HPC growth.

Summary of Outcomes

- Participants gained confidence in deploying and managing repurposed Stampede2 and GPU systems.
- Training reduced friction between users and administrators through structured troubleshooting paradigms.
- The adapted curriculum seeded a community of practice inspired by the HPC Ecosystems Project, ensuring sustainability despite limited vendor documentation.

These outcomes suggest that the principles guiding the adapted training model hold promise for similar contexts.

8 FUTURE WORK

Building on this preliminary wireframe, we intend to run a larger cohort to gather more definitive quantitative and qualitative feedback. This will allow us to assess whether the adapted training model demonstrates promise beyond its preliminary pilot, and to strengthen our understanding of the model's efficacy and further refine the curriculum based on evidence from a broader and more diverse participant group.

9 CONCLUSION

Adapting leadership-class HPC training for resource-constrained environments suggests that sustainable capacity building requires both technical and cultural transformation. By embedding policy into user education and reframing operations around pragmatic, context-aware practices, the adapted program enabled new adopters to operate legacy KNL systems with confidence and autonomy. More importantly, it highlighted the potential value in re-engineering global HPC training models to reflect the realities of emerging centres rather than the assumptions of mature ones. Future work will expand shared documentation, deepen regional collaboration, and evaluate the long-term impact of this training framework across multiple African HPC sites. We encourage the broader HPC community to consider adopting these principles as a framework for designing training models that meaningfully support emerging and resource-constrained centres.

ACKNOWLEDGMENTS

The authors would like to thank the many individuals whose expertise, support, and operational efforts contributed to the success of the Building Machines for Science workshop.

We gratefully acknowledge Mitchell Collins, Joe Allen, Erik Ferlanti, Lena Duplechin Seymour, and Bob Garza for their contributions in teaching, mentoring participants, and supporting workshop activities throughout the program.

We also thank Nathaniel Mendoza and Nic Lewis for designing, configuring, and maintaining the training infrastructure used during the workshop.

This workshop was heavily inspired by the original Designing and Administering Large Scale Systems Institute. We would like to recognize Nick Thorne, Cyrus Proctor, and Je'aime Powell for their foundational work in developing the original institute model and curriculum that informed this training effort.

Finally, we would like to acknowledge Dan Stanzione for his continued leadership and support of workforce development, research computing education, and scientific cyberinfrastructure initiatives at Texas Advanced Computing Center.

Some editing of grammar and phrasing was performed using Microsoft CoPilot. NIH Grant G11TW013072.

REFERENCES

[1] William J. Allen, Kelsey M. Beavers, Erik Ferlanti, Lorenzo Concia, Joshua Urrutia, Ernesto A. B. F. Lima, John M. Fonner, Felix Zuo, H. E. Duplechin Seymour, Ari B. Kahn, Joe Stubbs, Anagha Jamthe, Stephanie N. Baker, Tabish Khan, and James P. Carson. 2025. A Model for Teaching Machine Learning, Deep Learning, and Research Computing to Domain Scientists on HPC Resources. In *Proceedings of the SC '25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC Workshops '25)*. Association for Computing Machinery, New York, NY, USA, 401–408. https://doi.org/10.1145/3731599.3767380

[2] Erik Ferlanti, William J. Allen, Ernesto A. B. F. Lima, Yinzhi Wang, and John M. Fonner. 2023. Perspectives and Experiences Supporting Containers for Research Computing at the Texas Advanced Computing Center. In *Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis (SC-W '23)*. Association for Computing Machinery, New York, NY, USA, 155–164. https://doi.org/10.1145/3624062.3624587

[3] Bryan Johnston, Charles Crosby, Quinn G. Reynolds, Jennifer M. Schopf, and Christopher J. Whalen. 2025. Sticking (with) the Landing A modern case for Knights Landing in Resource-Constrained Environments. *PEARC 2025 - Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration* (7 2025). https://doi.org/10.1145/3708035.3736075;PAGE:STRING:ARTICLE/CHAPTER

[4] Bryan Johnston, Lara Timm, Eugene de Beste, and Mabatho Hashatsi. 2024. Let's Get Our Heads Out of the Clouds: A scalable and sustainable approach to HPC Training Labs for Resource Constrained Environments and anyone else stuck in the Clouds. *Journal of Computational Science Education* 15, 1 (March 2024), 64–71. https://doi.org/10.22369/issn.2153-4136/15/1/13 Presented at Supercomputing SC23.

[5] Bryan Johnston, Lara Timm, David Macleod, and John Poole. 2024. Ten Years of the HPC Ecosystems Project: Transforming HPC in Africa for the Past Decade. In *Practice and Experience in Advanced Research Computing (PEARC '24)*. ACM, 1–11. https://doi.org/10.1145/3626203.3670537

Introducing HPC-Enabled Computational Research Training in Africa: A Pilot Study

James Afful
Iowa State University

Titus Nyarko Nde
Boise State University

Onyeka Onyenemezu
Iowa State University

Brian Kyanjo
Georgia Institute of Technology

Benjamin Atribawuni Asaaga
Kwame Nkrumah University of Science and Technology

Joseph Oppong Akowuah
Kwame Nkrumah University of Science and Technology

ABSTRACT

Access to computational training and research-computing mentorship remains uneven across global scientific communities. This paper reports a pilot study of the Center for Computational Research and Education (CCRE), a training initiative designed to introduce students and early-career researchers in Africa to computational research through a staged model of open recruitment, structured preparation, mentored project work, and reflective evaluation. The pilot was advertised through LinkedIn, WhatsApp, and Instagram for more than two months, and 24 applicants registered. Twelve applicants were selected for a mandatory four-module Google Classroom preparatory course; six completed all required modules and advanced to the mentored group-project phase. Evaluation used pre- and post-program survey summaries from the final cohort. The pre-program survey had six responses, and the post-program survey had five responses. Baseline results showed strong motivation to build practical high-performance-computing (HPC) skills, connect with a research-computing community, learn collaborative software practices, and explore computational approaches in participants' domains. They also showed substantial structural barriers: most respondents reported no prior HPC access and no or limited formal training in HPC, parallel computing, or research computing. Post-program respondents gave an average overall rating of 4.20 out of 5, and all five indicated that they would definitely recommend the program. Respondents nevertheless continued to report barriers related to HPC access, local mentorship, and computing cost. The CCRE pilot is presented as an educational-practice model whose principal contribution is a documented and critically evaluated pathway for introducing computational research training under constrained infrastructure and mentorship conditions.

KEYWORDS

Computational Training, HPC Education, Africa, Research Computing, Pilot Study, Project-based Learning, Mentorship, Cyberinfrastructure Workforce Development

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/3

1 INTRODUCTION

Computational methods are now central to research in engineering, the physical sciences, the life sciences, climate science, public health, data science, and artificial intelligence. Modern scientific work increasingly requires the ability to use command-line environments, version control, remote computing systems, batch schedulers, parallel workflows, data pipelines, and reproducible computational practices. In many research domains, these skills shape the scale of questions that can be asked and the kinds of evidence that can be produced. Access to computational training, however, remains uneven. High-performance-computing (HPC) and research-computing education require more than hardware. Students need structured learning pathways, mentors, examples of domain-specific computational research, access to computing environments, and peer communities that make computational research feel attainable. Existing work on HPC education identifies persistent challenges in curriculum design, educator preparation, student access, and the integration of HPC with contemporary areas such as artificial intelligence and data science [20]. Reviews of scientific-computing, HPC, and data-science education similarly show growing demand for research-computing instruction and the need for more systematic training pathways [19, 24]. These challenges are intensified where local HPC infrastructure, formal courses, and trained instructors are limited.

The African context requires particular attention. A systematic review of computing-education research in Africa found that the region remains underrepresented in the literature and that published work is unevenly distributed across countries and topics, with programming being the most common topic (43% of 68 studies) [22]. At the same time, African and Africa-oriented HPC initiatives have demonstrated that capacity building is possible when infrastructure, training, mentorship, and community formation are treated together rather than separately [1, 14, 15]. Work on sustainable HPC infrastructure in resource-limited settings further emphasizes that durable computational capacity depends on governance, maintenance, training, local ownership, and long-term support, not simply on acquiring computing resources [12].

This paper reports a pilot study of the Center for Computational Research and Education (CCRE), an initiative designed to introduce students and early-career researchers in African contexts to computational research. The program was intentionally small and was not designed to produce large-scale statistical evidence of learning gains. It was designed to test a practical model for introducing computational training, identify barriers faced by learners, and generate lessons for future cohorts.

The central research and practice questions were:

(1) What kinds of learners were reached by an open, social-media-based recruitment strategy for an Africa-oriented computational training program?
(2) What baseline barriers to HPC and research-computing participation were reported by the final pilot cohort?
(3) How did participants evaluate the program after completing the mentored-project phase?
(4) Which barriers appeared to remain after the pilot?
(5) What design lessons can be drawn for future computational-training programs in African contexts?

The primary contribution of this paper is the documentation and preliminary evaluation of an implementable educational-practice model for introducing African students and early-career researchers to computational research.

Specifically, the study contributes: (i) a staged training pathway connecting open recruitment, application review, structured preparation, and mentored computational projects; (ii) descriptive evidence concerning participant motivations, access barriers, satisfaction, and perceived instructional value; and (iii) practical implementation lessons for future computational training initiatives operating under infrastructure, connectivity, and mentorship constraints.

The study does not claim that the structural barriers identified are newly discovered, nor does it establish that the training intervention produced measurable technical proficiency gains. Rather, it provides an early feasibility and educational-practice account that can inform the design and systematic evaluation of larger future cohorts.

2 RELATED WORK

2.1 HPC and Research-Computing Education

HPC education lies at the intersection of computer science, domain science, systems knowledge, and practical workflow training. Raj et al. describe HPC education as a field facing curricular, institutional, and pedagogical challenges, including the need to define competencies, support educators, and connect HPC learning to artificial intelligence, data science, and contemporary computing education [20]. Ponce et al. review scientific-computing, HPC, and data-science training programs and argue that demand for advanced research-computing instruction has grown substantially, with learners increasingly seeking data-oriented and interdisciplinary computational skills [19]. Van Zon et al. similarly document growth and diversification in scientific-computing training delivered through a supercomputing center, including participation from nontraditional computing disciplines [24].

These studies support the premise that computational training should not be restricted to computer-science majors. Scientists and engineers increasingly need practical research-computing fluency. Reviews of HPC foundations for scientists emphasize that domain researchers often require practical HPC knowledge even when their formal training is outside computer science [13]. The CCRE pilot follows this broader direction by targeting learners across engineering, the physical sciences, computational biology, data-oriented fields, and related areas.

2.2 Cyberinfrastructure Workforce Development and Modular Training

Research-computing education is also part of cyberinfrastructure workforce development. The National Science Foundation's cyberinfrastructure learning and workforce-development efforts frame training as essential to broad use of advanced computational resources across the research community [17]. Landscape studies of research computing emphasize that access varies widely across institutions and that less-resourced communities need support to participate in advanced computational work [5]. Microcredential and modular training approaches have been proposed as one mechanism for widening access to computing education, especially for learners who need targeted competencies rather than complete degree programs [6].

The CCRE pilot used a mandatory four-module Google Classroom sequence before the mentored-project phase. In this pilot, the module sequence functioned as both a readiness-building mechanism and a participation filter. This design is consistent with broader movements toward modular, accessible, and scaffolded computational training. Unlike formal microcredential programs, however, the pilot did not issue standardized credentials or conduct externally validated skill assessments. That distinction is important when interpreting the results.

2.3 African Computing Education and HPC Capacity Building

The literature on computing education in Africa is growing but remains uneven. Sanusi and Deriba systematically review general computing education research, rather than HPC-specific instruction, and describe an uneven geographical and topical distribution, including substantial emphasis on programming [22]. Their review is therefore used here to characterize the broader education-research landscape, not as direct evidence of HPC training outcomes or as proof that research-computing education is absent. More directly relevant African examples include HPC ecosystem-development and student-cluster training initiatives [14, 15]. Recent regional work also examines practical ICT-training design and delivery barriers in universities in South Africa, Cameroon, and Nigeria [4]. Although that work is not HPC-specific, it provides relevant evidence about instructional implementation in resource-constrained environments.

Africa-oriented HPC capacity-building efforts provide important precedents. Abiona et al. discuss the ICTP experience in HPC infrastructure setup and training in Africa, emphasizing that capacity building involves developing people and instructor communities, not only deploying machines [1]. The HPC Ecosystems Project provides another model by combining infrastructure deployment, formal training workshops, and community development across multiple countries [15]. The South African Student Cluster Competition further demonstrates how selection, training, mentorship, and authentic technical challenges can support high-level student participation in HPC [14].

These efforts indicate that African HPC capacity building should be understood as an ecosystem problem. CCRE contributes to this literature by documenting a small, project-based, Africa-oriented

pilot that began with open recruitment and proceeded through structured preparation into mentored computational project work.

Recent African research-training initiatives provide additional points of comparison for the CCRE model. BioStruct-Africa describes a community-driven framework integrating computational structural biology, AlphaFold-enabled research, sustained mentorship, and long-term workforce development [2]. Its proposed training targets represent program ambitions rather than independently demonstrated educational outcomes.

A capacity-strengthening initiative involving six East African Community partner states similarly describes practical technical instruction, including introductory HPC exposure, together with training-of-trainers and follow-up support [18].

Although these initiatives address different scientific domains and institutional settings, they illustrate complementary approaches to computational capacity building. Their emphasis on practical research engagement, local instructor development, and sustained institutional support reinforces the need to connect short-term training with longer-term research capacity.

The CCRE pilot adds an implementation-level account of an open-recruitment, staged-preparation, and mentored-project pathway. Its contribution is therefore the documentation of a potentially adaptable educational practice rather than evidence of superior learning outcomes or large-scale program effectiveness.

2.4 Project-Based Learning, Active Learning, and Mentorship

The CCRE pilot included foundational HPC training sessions and a final phase centered on mentored group projects. This design aligns with evidence that active learning improves outcomes in science, technology, engineering, and mathematics compared with lecture-only approaches [11]. Project-based and hands-on computational training are especially important in HPC education because many key competencies are procedural and contextual: learners must understand concepts and learn how to work in real computational environments.

Mentorship is also central. Studies of STEM mentorship show that high-quality mentoring and research experiences can support science identity, efficacy, values, and integration into scientific communities, especially for students from underrepresented groups [3, 9, 21]. For CCRE, mentorship should therefore be interpreted not only as technical troubleshooting but also as a mechanism for supporting research identity, belonging, and persistence in computational research.

2.5 Training Evaluation and Pilot-Study Reporting

Training programs are often evaluated using models that distinguish satisfaction from learning, behavior, and longer-term results. The Kirkpatrick model, for example, separates evaluation into reaction, learning, behavior, and results [16]. This distinction is important for the present pilot. The available CCRE data support claims about reaction, perceived value, baseline barriers, and intended future use.

Pilot studies are best used to examine feasibility, acceptability, process, implementation, and measurement procedures before larger-scale evaluation [8, 23]. The CCRE pilot should therefore be interpreted as an early program evaluation and educational-practice report. Its value lies in documenting the model, identifying barriers, and refining future implementation rather than proving definitive effectiveness.

3 PROGRAM CONTEXT

The Center for Computational Research and Education was developed to provide structured exposure to computational research, HPC, artificial-intelligence-enabled workflows, and collaborative scientific computing. The pilot focused on learners motivated to engage with computational research but who often lacked local access to HPC systems, formal training, mentors, or peers working in the area.

Recruitment relied on open advertisement through LinkedIn, WhatsApp, and Instagram over more than two months. Social-media recruitment was intentional because it allowed the program to reach motivated learners beyond a single classroom or institution. The application review record documented 24 applicants.

The program aimed to select a smaller group for deeper preparation and mentorship. Selection considered motivation, application fit, programming experience, prior HPC exposure, teamwork experience, availability, academic stage, gender representation, and institutional or external balance. Twelve applicants were selected for the mandatory Google Classroom preparatory stage. Six completed all four required modules and advanced to the final pilot cohort. Figure 1 summarizes the pathway.

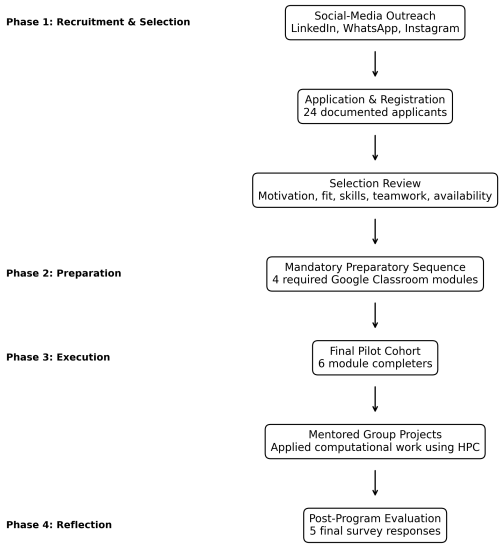


Figure 1: Staged pathway used in the CCRE pilot.

4 PROGRAM DESIGN

The CCRE pilot was organized around four design principles:

(1) **Access:** widen entry into computational research by recruiting beyond traditional local classroom channels.
(2) **Preparation:** provide a structured preparatory sequence before intensive project work.
(3) **Mentored practice:** move participants from exposure to application through group projects and mentor support.
(4) **Community and identity:** help participants see themselves as part of a research-computing community.

These principles are consistent with literature on HPC education, cyberinfrastructure workforce development, active learning, and mentorship [5, 9, 11, 19, 20]. Table 1 maps program components to intended outcomes and available evidence. This mapping connects the evaluation items to program objectives rather than treating the post-program instrument as a generic satisfaction survey.

5 EVALUATION METHODS

The pilot evaluation used pre- and post-program survey summaries. The pre-program survey was completed by six respondents corresponding to the final cohort that advanced into the pilot program. The post-program survey was completed by five respondents. Survey items addressed academic background, disciplinary background, prior HPC access, prior training, motivations, barriers, confidence in computational skills, research identity, satisfaction with program components, perceived impact, internet and connectivity constraints, intended future use of skills, and willingness to recommend the program.

The evaluation was descriptive. Because the sample was small and the available summaries did not confirm paired individual-level responses, inferential statistical tests were not conducted. Counts and percentages are reported where the survey summaries provide sufficient information. Results are interpreted as evidence about feasibility, acceptability, perceived value, and remaining barriers rather than definitive evidence of technical mastery.

The analysis drew on three program documents: a pre-program survey summary with six responses, a post-program survey summary with five responses, and an internal application-evaluation document describing review criteria and potential participants. The internal document contained personally identifiable applicant information. This paper therefore reports only aggregate or procedural information. Applicant names, personal motivations, and identifiable background details are not reported.

The evaluation drew on four complementary sources of evidence, summarized below.

- **Pre-program survey ($n = 6$):** The survey documented participants' academic and disciplinary backgrounds, prior access to HPC resources, previous formal training, motivations for participation, and perceived barriers to computational research. Self-reported technical confidence covered Linux command-line navigation, remote HPC login, batch-job scripting, job scheduling, parallel or distributed computing, and Git version control. Research identity and participants' expectations were also assessed.
- **Post-program survey ($n = 5$):** The survey examined overall satisfaction, perceived usefulness of training components, internet connectivity challenges, technical confidence, research identity, intended future use of computational skills, remaining barriers, overall program rating, and willingness to recommend the program. These responses reflect participants' perceptions and intentions rather than independently verified learning outcomes.
- **Application-evaluation records:** Internal program documentation recorded application-review criteria, participant selection, and progression through the preparatory phase. These records supported description of the recruitment and completion pathway but did not establish the reasons individual applicants failed to complete the required modules.
- **Mentored group-project descriptions:** The program documented two interdisciplinary computational projects: a machine-learning surrogate for reactor thermal-hydraulic analysis and a brain-tumor MRI classification workflow. These descriptions illustrate the nature of project-based computational activities but do not provide independently audited evidence of individual technical proficiency.

The evaluation was descriptive. Because the final cohort comprised only six participants, with five post-program respondents, the findings should not be generalized to the broader population of African students or early-career researchers. Furthermore, individual-level pre- and post-program survey responses were not confirmed as paired. Consequently, differences between the two survey summaries cannot be interpreted as measured within-person learning gains. No inferential statistical tests or causal-effect estimates were performed.

6 RESULTS

6.1 Recruitment and Participation Funnel

The program advertised through LinkedIn, WhatsApp, and Instagram over more than two months. The application-review record documented 24 applicants. Twelve were selected for the Google Classroom preparatory stage, and six completed all four required modules and formed the final pilot cohort.

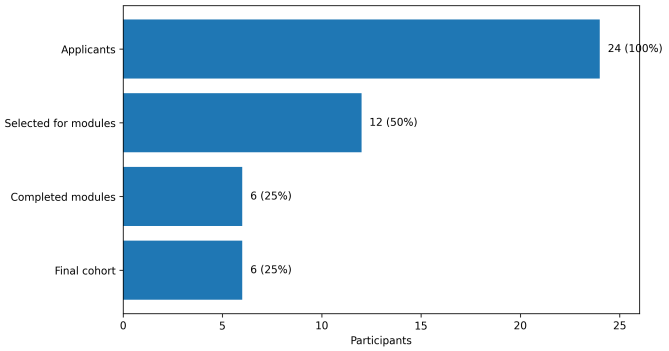


Figure 2: Recruitment, preparation, and participation funnel. Percentages are relative to the 24 documented applicants.

The program generated broader interest than it could intensively support, and progression into the final cohort required completion of preparatory work. As shown in Figure 2, 24 applicants were recorded, 12 were selected for the mandatory preparatory stage,

Table 1: CCRE training components, intended outcomes, and available evidence.

Component	Intended outcome	Available evidence
Social-media recruitment	Reach motivated learners beyond a single institution or formal course	Recruitment record; pre-survey source-of-awareness item
Application review	Select applicants with strong motivation, availability, and readiness for project-based training	Internal selection document
Google Classroom modules	Establish baseline preparation and identify participants able to complete required asynchronous work	Module completion; final cohort size
Foundational technical sessions	Build confidence in Linux, remote computing, job scheduling, scripting, and version control	Pre- and post-program confidence items
Mentored group projects	Move from passive exposure to applied computational work	Post-survey impact item; reactor-safety Streamlit tool; MRI classification and Grad-CAM tool
Office hours and troubleshooting	Reduce technical barriers and support persistence	Post-survey satisfaction item
Guest research sessions	Expose participants to computational-research pathways and community	Post-survey satisfaction item
Final presentation and closeout	Develop research communication and accountability	Post-survey satisfaction item; project presentations

and six completed all four modules and advanced to the mentored-project phase. Thus, one quarter of documented applicants entered the final pilot cohort.

The available records do not establish why some selected applicants did not complete the required modules. Future cohorts should track module-level completion, time burden, connectivity constraints, and reasons for withdrawal to distinguish barriers to participation from other causes of attrition.

6.2 Application Review and Baseline Characteristics

Table 2 summarizes the dimensions considered during the application-review process. The internal evaluation document indicates that review criteria included motivation, application fit, programming experience, prior HPC exposure, teamwork experience, availability, gender representation, and academic or career stage. These criteria highlight a design tension for future computational capacity-building programs. Intensive mentorship requires selecting participants with the readiness and availability to persist, whereas the broader capacity-building objective also requires expanding opportunities for learners with limited prior exposure to HPC.

Table 3 summarizes academic level, disciplinary background, prior HPC access, and prior formal training among the six pre-program respondents.

The final cohort was interdisciplinary and mostly lacked prior access to HPC systems. Five of six respondents reported no previous access to an HPC cluster or supercomputing resource, while one reported access through an external program or grant. Three reported no formal training, two reported partial exposure, and one reported an online course or workshop. No respondent reported completing a full university course in HPC, parallel computing, or research computing.

Participants reported strong motivation across technical, community, and professional dimensions. Table 4 summarizes their selected goals.

These motivations show that participants sought technical instruction, community, collaborative software practice, and pathways into computational research. This supports the decision to design CCRE as a mentored cohort rather than a one-time workshop. Figure 3 summarizes the barriers reported by respondents before and after the program. The pre-program survey showed substantial barriers to HPC learning. All six respondents reported a lack of local instructors or mentors with HPC expertise. Five reported lack of access to HPC systems and the cost of cloud or commercial computing resources. Four reported a lack of a structured curriculum and a lack of nearby peers working in the area.

These findings are consistent with literature emphasizing that computational capacity requires training, infrastructure, mentorship, and communities of practice [1, 5, 12]. Because the pre-program survey included six respondents and the post-program survey included five respondents, the two distributions should be interpreted as separate descriptive summaries rather than paired measurements of individual change.

6.3 Post-Program Satisfaction and Perceived Value

The post-program survey received five responses. As summarized in Table 5, respondents gave the program an average overall rating of 4.20 out of 5. Four respondents assigned a rating of 4, while one assigned a rating of 5. All five respondents indicated that they would definitely recommend the program to a colleague or peer.

These findings suggest favorable participant reactions within the small responding cohort. However, satisfaction and recommendation responses should not be interpreted as objective evidence of technical skill acquisition or long-term research capacity development.

These results support a claim of high acceptability and perceived value but do not establish technical mastery. In the Kirkpatrick framework, the data primarily represent reaction and perceived usefulness, not independently measured learning, behavior change, or long-term institutional impact [16]. Table 6 presents respondents' assessments of the program components that contributed most strongly to their perceived learning experience.

Two of the five respondents selected working on the team project using real HPC resources as the component with the greatest impact. Foundational technical sessions, peer collaboration and team

Table 2: Application review criteria used in the CCRE pilot.

Criterion	Role in selection
Motivation and fit	Primary criterion for identifying applicants aligned with computational-research training
Programming experience	Readiness indicator for technical participation
Prior HPC exposure	Background indicator; lack of exposure did not automatically exclude applicants
Teamwork experience	Indicator of readiness for mentored group projects
Availability	Feasibility criterion for participation in a time-bound pilot
Gender representation	Broadening-participation consideration during review
Academic or career stage	Tie-breaker, with attention to undergraduate and master's-level applicants
Institutional or external balance	Consideration of institution-linked and external applicants

Table 3: Baseline characteristics of pre-program survey respondents ($n = 6$).

Domain	Category	Count
Academic level	Undergraduate student	3
	Master's student	1
	PhD student	0
	Early-career researcher or postdoctoral scholar	2
	Other	0
Disciplinary background	Engineering	3
	Physical sciences	1
	Computational biology or bioinformatics	1
	Other	1
	Computer science or software engineering	0
	Mathematics or statistics	0
	Data science, artificial intelligence, or machine learning	0
Prior HPC access	No prior access	5
	External program or grant	1
	University-provided access	0
	Cloud-platform access	0
Prior formal training	No formal training	3
	Partial exposure	2
	Online course or workshop	1
	Full university course	0

Table 4: Goals for joining the CCRE pilot ($n = 6$).

Goal	Count
Build practical HPC skills for research	6
Connect with a research-computing community	6
Learn collaborative software practices	6
Explore computational approaches in the participant's domain	6
Strengthen a CV or graduate-school application	5
Other	0

Table 5: Post-program satisfaction and recommendation results ($n = 5$).

Measure	Result
Post-program survey respondents	5
Average overall program rating	4.20/5
Respondents giving a rating of 4	4
Respondents giving a rating of 5	1
Respondents who would definitely recommend the program	5/5

dynamics, and mentor feedback and guidance were each selected by one respondent. Guest-researcher talks were not selected as the single most impactful component by any respondent. These responses provide preliminary insight into which activities participants valued most, but the very small sample prevents robust comparisons among instructional components.

This result supports one of the pilot's central design claims: computational training should move beyond passive exposure into mentored practice. The finding aligns with active-learning evidence in STEM education and HPC-education literature emphasizing hands-on practice [7, 10, 11].

Internet connectivity remained a relevant participation constraint. As shown in Table 7, three of the five post-program respondents reported moderate internet issues that caused them to

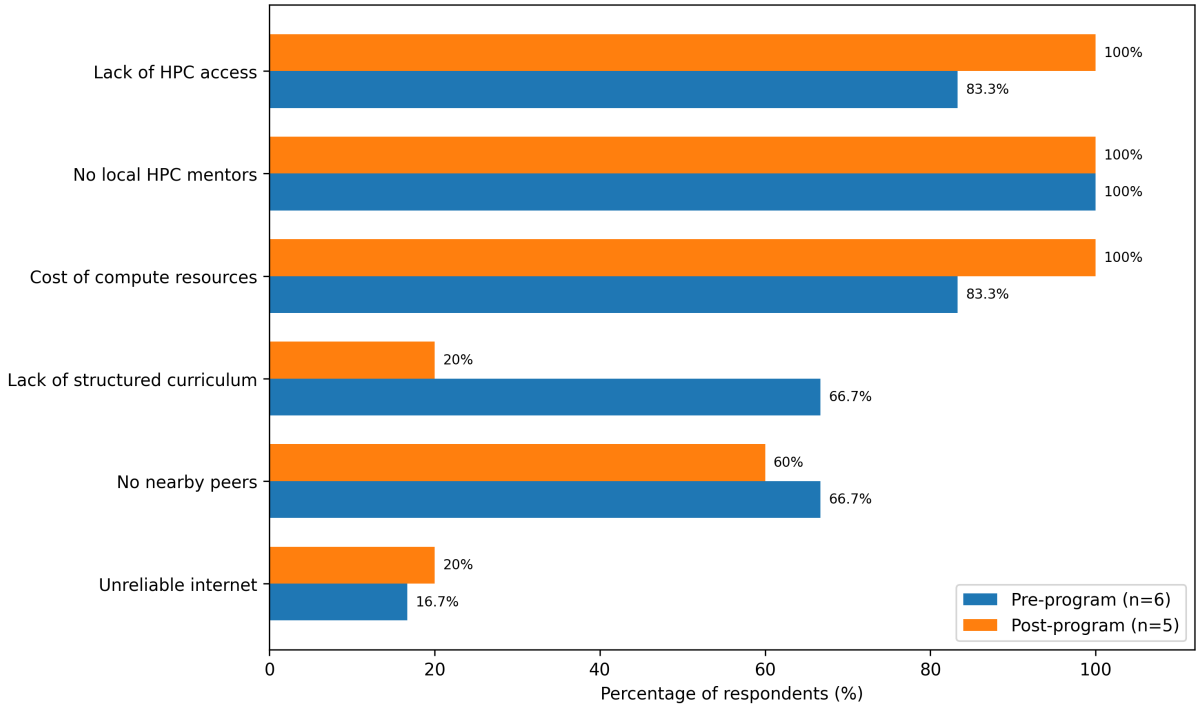


Figure 3: Reported barriers to HPC learning in the pre- and post-program surveys. Because the surveys had different denominators and responses were not confirmed as paired, the values should be interpreted descriptively rather than as individual-level change.

Table 6: Program aspect reported as having the greatest impact on learning ($n = 5$).

Aspect	Count
Team project using real HPC resources	2
Foundational technical sessions	1
Peer collaboration and team dynamics	1
Mentor feedback and guidance	1
Guest-researcher talks	0

Table 7: Reported effect of internet connectivity on participation ($n = 5$).

Connectivity effect	Count
Moderate issues; missed some sessions or content	3
Minor issues; participation not significantly affected	1
No issues	1
Significant issues; participation substantially limited	0

Table 8: Intended future use of CCRE skills ($n = 5$).

Intended use	Count
Include the experience in graduate-school applications	5
Apply to future HPC programs or internships	5
Apply HPC methods to current research or coursework	4
Share knowledge with peers or teach others	4
Contribute to HPC capacity at the participant's institution	2
Pursue a role in research computing or data science	1

miss some sessions or content. One respondent reported minor connectivity issues without substantial participation effects, while one reported no connectivity problems. No respondent reported severe connectivity difficulties that substantially prevented participation.

This connectivity-impact measure is distinct from the general barriers checklist reported in Figure 3. The two questions therefore capture related but different aspects of participants' experiences.

Connectivity did not prevent completion among the post-survey respondents, but it affected participation for a majority. Future cohorts should therefore provide asynchronous backups, downloadable materials, low-bandwidth options, and structured catch-up mechanisms.

Table 8 summarizes how respondents expected to use what they learned.

These results indicate strong intended transfer, but they remain statements of intent. A stronger future evaluation should track whether participants subsequently use HPC methods in coursework or research, apply to programs, mentor peers, contribute to local capacity, or produce computational-research outputs.

Table 9 compares pre- and post-program barrier reports. Because the pre-survey had six respondents and the post-survey had five,

both counts and percentages are shown. Because responses were not confirmed as paired, differences should not be interpreted as measured individual-level change.

The proportion reporting a lack of structured curriculum was lower in the post-program summary, while access, mentorship, and cost remained highly prevalent. These descriptive comparisons suggest that the program may have helped address the need for a structured learning path, but the data do not establish causality or individual-level change. The findings clarify the scale of the problem: training programs can create exposure and structure, but durable computational capacity requires sustained infrastructure access, local mentor development, and affordable computing pathways.

6.4 Applied Project Outcomes

The six participants were divided into two groups, each tasked with executing an end-to-end computational project using HPC resources and machine-learning workflows.

The first group developed a machine-learning surrogate model for steady-state thermal-hydraulic safety analysis of a 10 MW Material Testing Reactor. Using an HPC-generated database of 1,000 samples from the PLTEMP/ANL v4.3 simulation code, the team evaluated Random Forest, Gradient Boosting, and feedforward neural-network models. The group identified Gradient Boosting as the most accurate of the evaluated surrogate approaches and deployed the model through an interactive Streamlit graphical interface intended to support rapid safety assessments without rerunning the computationally expensive source simulations.

The second group developed an automated brain-tumour classification tool using magnetic-resonance images. The group used two-phase transfer learning to fine-tune EfficientNetB3, VGG16, and DenseNet121 convolutional-neural-network architectures. The resulting ensemble was deployed through a web application with Grad-CAM heatmaps that visualized image regions influencing the model's classifications.

These group artifacts show that, with mentor support, the cohort participated in end-to-end computational work extending from foundational workflow preparation to model training, comparison, and web deployment. The evidence does not independently establish each participant's technical mastery or individual contribution.

7 DISCUSSION

7.1 A Staged Model for Computational Training

The CCRE pilot provides preliminary evidence of the feasibility and acceptability of a staged model for introducing computational training in African contexts. The model begins with open outreach, proceeds through application review, uses mandatory asynchronous preparation to establish baseline readiness, and advances participants who complete the preparation into mentored group projects. This structure differs from a one-time workshop.

The recruitment funnel may reflect time constraints, module difficulty, internet access, competing obligations, or other factors, but the available data do not identify the reasons. Future programs should collect process data on module engagement, missed sessions, reasons for noncompletion, and participant time burden.

The post-program survey suggests that applied project work was an important component of the experience. Working on a team project using real HPC resources was the most frequently selected high-impact component, although it was chosen by only two of the five respondents. This finding is consistent with evidence that active learning can improve STEM outcomes and that computational skills are best learned through practice [11]. In HPC education, learners need to interact with computing environments, scripts, errors, queues, dependencies, filesystems, and collaboration tools that cannot be learned fully through lectures alone.

Future cohorts should preserve and strengthen the group-project phase. The completed artifacts show that the cohort, with mentor support, participated in domain-specific computational work involving a thermal-hydraulic surrogate model with Streamlit and an MRI-classification ensemble with Grad-CAM visualization. Future evaluations should assess such artifacts through a structured rubric and should distinguish team-level products from individual learning outcomes.

The survey summaries also show that mentorship is not peripheral. All six pre-program respondents reported a lack of local instructors or mentors with HPC expertise, and all five post-program respondents identified the same barrier. This finding is consistent with mentorship literature showing that mentoring can support research identity, belonging, and integration into scientific communities [3, 9, 21].

For CCRE, mentorship should be treated as infrastructure. A program that depends on a small number of expert mentors will be difficult to scale. Future cohorts could use a tiered mentorship model in which alumni serve as peer mentors for subsequent cohorts. Over time, this could help transform CCRE from a training program into a distributed community of practice.

The most important structural finding is that barriers related to HPC access, local mentorship, and computing cost remained highly prevalent after the program. The pilot created perceived value while revealing limits that a short training intervention cannot resolve. Sustainable HPC capacity in resource-limited settings depends on access, maintenance, governance, training, and local ownership [12]. Future CCRE work should therefore combine training with partnerships for compute access, institutional support, and mentor development.

7.2 A Practical Framework for Future Cohorts

Based on the pilot results, four pillars are proposed for future computational-training efforts in African contexts:

(1) **Access:** provide learners with exposure to real computational environments and workflows.
(2) **Preparation:** use structured modules to establish baseline readiness before intensive project work.
(3) **Mentored practice:** organize learners into projects with technical and research mentorship.
(4) **Sustainability:** build alumni mentorship, institutional partnerships, and longer-term compute access.

This framework is grounded in both the pilot data and literature on HPC education, cyberinfrastructure workforce development, active learning, and mentorship [5, 9, 11, 20].

Table 9: Reported barriers before and after the CCRE pilot. Percentages use the relevant survey denominator.

Barrier	Pre count	Pre %	Post count	Post %
Lack of access to HPC systems	5/6	83.3%	5/5	100.0%
No local instructors or mentors with HPC expertise	6/6	100.0%	5/5	100.0%
Cost of cloud or commercial computing resources	5/6	83.3%	5/5	100.0%
Lack of a structured curriculum or learning path	4/6	66.7%	1/5	20.0%
No peers working in this area nearby	4/6	66.7%	3/5	60.0%
Unreliable internet connection	1/6	16.7%	1/5	20.0%
No significant barriers	0/6	0.0%	0/5	0.0%

8 LIMITATIONS

This pilot has several important limitations.

First, the sample size was small: the pre-program survey had six responses, and the post-program survey had five. Results should therefore be interpreted descriptively rather than inferentially.

Second, the available survey summaries do not confirm paired individual-level responses. Without anonymous identifiers linking pre- and post-program responses, individual-level change cannot be measured reliably.

Third, the evaluation relies heavily on self-report. Participants reported satisfaction, perceived value, confidence, and intended future use, but the pilot did not include an independent assessment of technical skill.

Fourth, the survey summaries do not provide raw item-level data for all Likert-scale constructs. Exact means and changes for each confidence or identity item should be calculated only from verified raw survey exports.

Fifth, attrition occurred between registration, selection, module completion, and final participation. The current data document the funnel but do not explain why selected participants did or did not complete the preparatory modules.

Sixth, the final cohort was selected rather than randomly sampled. This was appropriate for a pilot program but limits generalizability.

Seventh, long-term outcomes are unknown. The post-program survey captures intended future use, not whether participants subsequently used HPC in research, applied to programs, published work, mentored peers, or contributed to institutional capacity.

Eighth, the small cohort creates privacy risks. Reporting should avoid identifiable combinations of country, institution, academic stage, field, and personal motivation unless explicit consent permits their disclosure.

9 RECOMMENDATIONS FOR FUTURE IMPLEMENTATION

The pilot suggests five concrete improvements:

(1) **Use anonymous participant identifiers.** Link future pre- and post-program surveys through anonymous identifiers so that individual-level change can be analyzed without compromising privacy.
(2) **Track module analytics.** Record module starts, completion, quiz scores, time to completion, and reasons for noncompletion where possible.
(3) **Provide low-bandwidth alternatives.** Record sessions, provide downloadable slides and materials, and design assignments that can be completed under intermittent connectivity.
(4) **Develop alumni pathways.** Invite participants who complete the program to mentor future cohorts, lead focused sessions, or maintain project repositories.
(5) **Secure institutional and compute partnerships.** Pursue partnerships with universities, national laboratories, cloud-credit programs, HPC centers, and research networks.

10 CONCLUSION

This paper presented a pilot study of the Center for Computational Research and Education, an initiative designed to introduce students and early-career researchers in African contexts to computational research, HPC, artificial-intelligence-enabled workflows, and collaborative scientific computing. The pilot used a staged model consisting of open social-media recruitment, application review, mandatory Google Classroom preparation, selective progression into a final cohort, mentored group projects, and post-program evaluation.

Final-cohort respondents were highly motivated but had limited prior HPC access and formal training. Post-program respondents rated the program highly and unanimously indicated that they would recommend it. The team-project component using real HPC resources was the most frequently selected high-impact aspect. At the same time, respondents continued to report barriers related to HPC access, local mentorship, and cost.

The main lesson is that computational training in African contexts should be designed as an ecosystem rather than a one-time workshop. Training can provide structure, exposure, perceived value, and community, but durable capacity requires sustained computing access, mentor development, institutional partnerships, and pathways for alumni to support future learners. The CCRE pilot is therefore best understood as a practical and critically evaluated model for developing a future phase of computational-research training.

DATA AVAILABILITY

The analysis is based on program-level survey summaries and an internal application-evaluation document. Because the underlying materials may contain personally identifiable information, raw survey and application materials are not publicly released. De-identified aggregate data used in the tables can be made available upon reasonable request, subject to ethical and privacy constraints.

ACKNOWLEDGMENTS

The authors thank the participants of the CCRE pilot program for their engagement, persistence, and feedback. The authors acknowledge the mentors, speakers, and collaborators who supported the program. The authors thank Doreen Anyeley Afful for serving as the administrative lead. The authors also thank the Boise State University HPC team for providing computing resources and office-hour support used by the trainee groups. The authors acknowledge the use of the open-source Brain Tumor MRI Dataset by Masoud Nickparvar, which supported the biomedical-classification project.

CONFLICT OF INTEREST STATEMENT

The authors declare no competing interests.

REFERENCES

[1] Olatunde Abiona, Clement Onime, Stefano Cozzini, and Sebsibe Hailemariam. 2011. Capacity Building for HPC Infrastructure Setup in Africa: The ICTP Experience. In *2011 IST-Africa Conference Proceedings*. IEEE, 1–8. https://doi.org/10.1109/ISTAFRICA.2011.6107383

[2] Justice O. Amofa, Juergen George, Noella A. Okumu, et al. 2026. BioStruct-Africa's scalable framework for AlphaFold-enabled research training and sustainable workforce development in Africa. *Communications Biology* 9 (2026), 321. https://doi.org/10.1038/s42003-026-09711-x

[3] Kaitlyn Atkins, Bryan M. Dougan, Michelle S. Dromgold-Sermen, Hannah Potter, Viji Sathy, and A. T. Panter. 2020. "Looking at Myself in the Future": How Mentoring Shapes Scientific Identity for STEM Students from Underrepresented Groups. *International Journal of STEM Education* 7, 1 (2020), 42. https://doi.org/10.1186/s40594-020-00242-3

[4] Maria Assunta Cappelli and Abdeljalil Akkari. 2025. Academic ICTs training in South Africa, Cameroon, and Nigeria. Strategies for ICT training course design. *Education and Information Technologies* 30 (2025), 10463–10510. https://doi.org/10.1007/s10639-024-13200-1

[5] Dhruva K. Chakravorty, Sarah K. Janes, James V. Howell, Lisa M. Perez, Amy Schultz, Marie Goldie, Austin L. Gamble, Rajiv Malkan, Honggao Liu, Daniel Mireles, Yuanqi Jing, Zhenhua He, and Tim Cockerill. 2022. Expanding the Reach of Research Computing: A Landscape Study: Pathways Bringing Research Computing to Smaller Universities and Community Colleges. In *Practice and Experience in Advanced Research Computing (PEARC '22)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3491418.3535169

[6] Dhruva K. Chakravorty, Richard Lawrence, Zhenhua He, Wesley Brashear, Honggao Liu, Andrew J. Palughi, Lisa M. Perez, Xin Yang, Jacob Pavelka, Ritika Mendjoge, Marinus Pennings, Randy McDonald, Gerry Pedraza, and Sunay V. Palsole. 2023. Access to Computing Education Using Micro-credentials for Cyberinfrastructure. *The Journal of Computational Science Education* 14, 2 (2023), 23–27. https://doi.org/10.22369/issn.2153-4136/14/2/5

[7] Paweł Czarnul, Mariusz Matuszek, and Adam Krzywaniak. 2024. Teaching High-Performance Computing Systems: A Case Study with Parallel Programming APIs: MPI, OpenMP and CUDA. In *Computational Science – ICCS 2024*, Leonardo Franco, Clélia de Mulatier, Maciej Paszynski, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M. A. Sloot (Eds.). Springer Nature Switzerland, Cham, 398–412. https://doi.org/10.1007/978-3-031-63783-4_29

[8] Sandra M. Eldridge, Claire L. Chan, Michael J. Campbell, Christine M. Bond, Sally Hopewell, Lehana Thabane, and Gillian A. Lancaster. 2016. CONSORT 2010 Statement: Extension to Randomised Pilot and Feasibility Trials. *BMJ* 355 (2016), i5239. https://doi.org/10.1136/bmj.i5239

[9] Mica Estrada, Paul R. Hernandez, and P. Wesley Schultz. 2018. A Longitudinal Study of How Quality Mentorship and Research Experience Integrate Underrepresented Minorities into STEM Careers. *CBE—Life Sciences Education* 17, 1 (2018), ar9. https://doi.org/10.1187/cbe.17-04-0066

[10] Kristen Finch, Ryan Beck, Xiaosong Li, Nam Pho, and Xiao Zhu. 2025. Computational Skills Training for Undergraduate Researchers in Molecular Engineering. *The Journal of Computational Science Education* 16, 1 (2025), 50–56. https://doi.org/10.22369/issn.2153-4136/16/1/10

[11] Scott Freeman, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active Learning Increases Student Performance in Science, Engineering, and Mathematics. *Proceedings of the National Academy of Sciences* 111, 23 (2014), 8410–8415. https://doi.org/10.1073/pnas.1319030111

[12] Ronald Galiwango, Christopher J. Whalen, Grace Kebirungi, Mugume T. Atwine, Rodgers Kimera, Alfred Ssekagiri, Timothy W. Kimbowa, Edward Lukyamuzi, Mike Nsubuga, and eLwazi ODSP. 2025. Ten Simple Rules for Building and Maintaining Sustainable High-Performance Computing Infrastructure for Research in Resource-Limited Settings. *PLOS Computational Biology* 21, 9 (2025), e1013481. https://doi.org/10.1371/journal.pcbi.1013481

[13] Pablo García-Risueño and Pablo E. Ibáñez. 2012. A Review of High Performance Computing Foundations for Scientists. arXiv:cs.DC/1205.5177

[14] Bryan Johnston, Nick Thorne, Matthew Cawood, Eugene de Beste, David Macleod, and John Poole. 2026. A Retrospective on South Africa's Student Cluster Competition and Its Model for Inclusive HPC Outreach and Training (2012–2020). *The Journal of Computational Science Education* 17, 1 (2026), 42–49. https://doi.org/10.22369/issn.2153-4136/17/1/6

[15] Bryan Johnston, Lara Timm, David MacLeod, and John Poole. 2024. Ten Years of the HPC Ecosystems Project: Transforming HPC in Africa for the Past Decade. In *Practice and Experience in Advanced Research Computing (PEARC '24)*. Association for Computing Machinery, New York, NY, USA, Article 35, 8 pages. https://doi.org/10.1145/3626203.3670537

[16] Donald L. Kirkpatrick. 1994. *Evaluating Training Programs: The Four Levels*. Berrett-Koehler Publishers, San Francisco, CA.

[17] National Science Foundation. 2017. *Blueprint for Cyberinfrastructure Learning and Workforce Development*. Technical Report. National Science Foundation. Cyberinfrastructure Learning and Workforce Development Blueprint.

[18] Julien A. Nguinkal, Florian Gehre, Hakim I. Lagu, Emmanuel Achol, Eric Nzeyimana, John N. Kiiru, Joseph Nyandwi, Gregory W. Dumo, Nyambura Moremi, Susan N. Nabadda, Isabelle Mukagatare, Andrea Molina, Jürgen May, and Muna Affara. 2025. A capacity strengthening model toward self-reliant and sustainable one-health workforce in six East African Community Partner States. *Frontiers in Public Health* 13 (2025), 1636817. https://doi.org/10.3389/fpubh.2025.1636817

[19] Marcelo Ponce, Erik Spence, Ramses van Zon, and Daniel Gruner. 2019. Scientific Computing, High-Performance Computing and Data Science in Higher Education. *The Journal of Computational Science Education* 10, 1 (2019), 24–31. https://doi.org/10.22369/issn.2153-4136/10/1/5

[20] Rajendra K. Raj, Carol J. Romanowski, John Impagliazzo, Sherif G. Aly, Brett A. Becker, Juan Chen, Sheikh Ghafoor, Nasser Giacaman, Steven I. Gordon, Cruz Izu, Shahram Rahimi, Michael P. Robson, and Neena Thota. 2020. High Performance Computing Education: Current Challenges and Future Directions. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '20)*. Association for Computing Machinery, New York, NY, USA, 51–74. https://doi.org/10.1145/3437800.3439203

[21] Rachael D. Robnett, Peter A. Nelson, Eileen L. Zurbriggen, Faye J. Crosby, and Martin M. Chemers. 2018. Research Mentoring and Scientist Identity: Insights from Undergraduates and Their Mentors. *International Journal of STEM Education* 5, 1 (2018), 41. https://doi.org/10.1186/s40594-018-0139-y

[22] Ismaila Temitayo Sanusi and Fitsum Gizachew Deriba. 2024. What Do We Know About Computing Education in Africa? A Systematic Review of Computing Education Research Literature. *SSRN Electronic Journal* (2024). https://doi.org/10.2139/ssrn.4826205 Also available as arXiv:2406.11849.

[23] Lehana Thabane, Jinhui Ma, Rong Chu, Ji Cheng, Afisi Ismaila, Lorena P. Rios, Reid Robson, Marroon Thabane, Lora Giangregorio, and Charles H. Goldsmith. 2010. A Tutorial on Pilot Studies: The What, Why and How. *BMC Medical Research Methodology* 10 (2010), 1. https://doi.org/10.1186/1471-2288-10-1

[24] Ramses van Zon, Marcelo Ponce, Erik Spence, and Daniel Gruner. 2019. Trends in Demand, Growth, and Breadth in Scientific Computing Training Delivered by a High-Performance Computing Center. *The Journal of Computational Science Education* 10, 1 (2019), 53–60. https://doi.org/10.22369/issn.2153-4136/10/1/9

Expanding the National AI Research Resource: Building Capacity and Community through Sustainable Research Pathways

Mary Ann Leung
Sustainable Horizons Institute
mleung@shinstitute.org

Connie L. McNeely
George Mason University
cmcneely@gmu.edu

Danielle Jacques
Independent
danielle@jacquescs.com

ABSTRACT

Focused on facilitating artificial intelligence (AI) research through the U.S. National Science Foundation (NSF) National AI Research Resource (NAIRR), the Sustainable Research Pathways (SRP) program provides an effective model for building an innovative, interactive, and wide-ranging community of research-competent computing sciences professionals. Aimed at expanding access to NAIRR for higher education faculty and students, SRP provides networking opportunities and cultivates collaborations with related researchers using structured, evidence-based approaches built on over 10 years of experience. More specifically, SRP incorporates three principal programmatic and interactive training strategies: 1) broadly recruit students and faculty/student teams from a wide range of backgrounds and institutions, 2) establish active collaborative research relationships through an iterative and successively more intensive matching process, and 3) catalyze the development of relevant competencies and of systemic and participatory research capacities. Accordingly, SRP operates to pave the way for further expansion of NAIRR.

SRP is aimed at developing skilled talent poised to address the pressing challenges facing the AI workforce. It has the potential to develop into a new and effective paradigm for expanding access to NAIRR while making significant progress in AI accessibility and providing a broader spectrum of perspectives on the development of AI models and methods. The program builds strong connections among the existing NAIRR researchers, institutions, and individual participants across the U.S., especially those who would not typically engage in NSF-sponsored AI research.

SRP expands NAIRR by democratizing, increasing the variety of AI talent, and advancing safe, secure, and trustworthy AI. The program explores changes to the approaches and practices of AI research through collaborations between NAIRR projects and new and emerging researchers–a crucial activity for future AI efforts.

KEYWORDS

Workforce Development, AI, Talent Development, U.S. NSF NAIRR, Trustworthy AI

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/4

1 INTRODUCTION

A critical challenge marking today's complex artificial intelligence (AI) ecosystem and landscape has been gaps in access to resources for conducting research and training the next generation of researchers. In the United States (U.S.), recognizing this issue, the National Science Foundation (NSF) National Artificial Intelligence Research Resource (NAIRR) Pilot was launched in January 2024 to provide a shared national research infrastructure designed to bridge these gaps [10]. The espoused goal of the NAIRR Pilot was to spur innovation, increase variety of talent, improve capacity, and advance safe, secure, and trustworthy AI in research and society. To that end, as described in this article, an evidence-based program and approach has been adopted in pursuit of this goal in keeping with the broader issue of AI access. More specifically, the Sustainable Research Pathways (SRP) program is examined here as a model for expanding access to NAIRR by matching existing NAIRR researchers with faculty and students at accredited U.S. academic institutions of higher education who are new to NAIRR. Its principal operative approach is to establish research collaborations through an experiential practicum along with community building activities and participation and presentation of research in a national professional forum.

AI has had a dramatic impact on scientific research with ubiquitous effects. However, with it comes a set of complexities. The ethical and fair development and engagement of AI is a topic of much debate as a socio-technological issue, especially since some algorithms have been found to mimic and exacerbate implicit unfairness and inequities in society [2]. Other problems have surfaced with algorithms due to directive assumptions and lack of data, resulting in the exclusion of some individuals from employment recruitment efforts. While the reasons behind such outcomes are complicated and under some debate, they suggest the importance of developing an AI research ecosystem comprised of individuals compelled and versed in the intricacies of related issues to ensure a broad spectrum of perspectives and considerations—which also speaks to a more democratized approach to AI disposition and engagement.

Democratizing, increasing variety of talent, improving capacity, and advancing safe, secure, and trustworthy AI require including and meaningfully engaging a broad spectrum of talent in the AI research enterprise. While a variety of efforts have been and are underway, the current cyberinfrastructure workforce is largely stagnant, reflecting a variety of complex socio-historical, political, cultural, and economic issues [11, 13]. Typically, as in other fields, AI researchers use their own networks to recruit collaborators just in time for project awards, creating a self-perpetuating stationary pool of talent. The NAIRR Pilot represents a significant opportunity

to expand and establish the national AI research enterprise utilizing a range of approaches in cultivating the AI workforce of the future.

It is in this sense that SRP for NAIRR (SRP-NAIRR) represents a crucial activity for the future of AI efforts in the U.S. It has been designed to address the perspectives and practices governing AI research throughout participating NAIRR projects. The program builds strong connections among existing NAIRR researchers and new and emerging researchers across the country, especially those who had not typically engaged in NSF AI research. The program is based on a highly successful Sustainable Research Pathways program established in 2014 to expand the U.S. Department of Energy (DOE) National Laboratory workforce [4, 5]. The goal is to broaden access to NAIRR resources by providing training opportunities and building meaningful connections between NAIRR researchers and new and emerging AI researchers and institutions.

2 SUSTAINABLE RESEARCH PATHWAYS

Sustainable Research Pathways (SRP) evidence-based techniques are the foundation for the SRP-NAIRR program. SRP-NAIRR focuses on three main affective strategies: 1) broadly recruit faculty/student teams and independent students from a variety of backgrounds and institution types, 2) establish meaningful collaborations through an iterative and successively more intensive matching process, and 3) foster broadly participatory and innovative AI ecosystems.

The following components are fundamental SRP activities: 1) recruit existing NAIRR project principal investigators (PIs) and faculty-student teams and independent students not currently engage in NAIRR and from accredited U.S. academic institutions of higher education; 2) facilitate a virtual matching workshop to identify collaboration partners; 3) provide a 10-week summer experiential research practicum; 4) organize post-summer Cultivating Ubiquitous Learning Through InnoVATive and Engaging (CULTIVATE) Conversations seminars; and 5) fund participants to attend the Building Engagement (BE) program at the Society of Industrial and Applied Mathematics (SIAM) Computational Science and Engineering conference where they present their summer work and begin to develop their professional standing.

The overall program design utilizes an integrated baseline intervention model grounded in techniques for engaging people in science, technology, engineering, and mathematics (STEM) [6], as posited in, for example, the Tinto model of student engagement and others [13–18]. SRP emphasizes five key focus areas, as shown in Figure 1, all of which contribute to building and engaging epistemic and professional community [6, 7].

As is apparent, the SRP is an integrated intervention approach, drawing support and direction in research and practice [14]. Among others, exemplary programs include the Meyerhoff Scholars [16], Minority Engineering Program (MEP) [15, 17, 18], and The Mathematics Workshop [7] The Meyerhoff Scholars program focuses on knowledge and skills, motivation and support, monitoring and advising, and academic and social integration. Elements of the MEP, which has been replicated on more than 100 campuses, include strong pre-college and community college outreach, concentrated efforts in working with freshman and sophomores, and emphasis on cooperative learning, clustering of students into learning communities, study centers, professional development with role

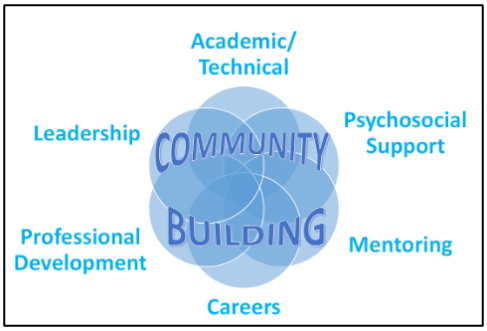


Figure 1: SRP Baseline Intervention Model

models, academic support, and personal counseling. The Mathematics Workshop, with more than 40 years of success, emphasizes group learning and community building. SRP adopts and combines a number of like features in its integrated approach, centered on building community, academic preparation, and strengthening self-efficacy.

In an overall sense, the SRP model has two principal defining features. 1) It includes components that address community building integrated through a mutually sustaining combination of academic preparation, psychosocial support, career exploration and preparation, and professional and leadership development, as detailed in Table 1. While each of these components contribute to breaking barriers to participation, their combination and integration strengthen the overall effect. Moreover, 2) SRP activities are realized in mainstream professional scientific communities in ways that balance critical mass and full engagement within and by the broader community. The point here is to strengthen the sense of community and belonging across all participants. The focus is not on "fixing the participant"; rather, the aim is to catalyze transformation and participation within the broader community.

Table 1: SRP Baseline Intervention Model Components

Component	Description
Academic Preparation	Students receive appropriate preparation for the rigors of STEM academic requirements, including expanded familiarity and engagement with "the language of science."
Psycho-Social Support	The individual develops a positive and active relationship with the scientific culture and belonging in the community.
Career Development	It is essential for persistence that the individual develops a vision of eventual acquisition of a rewarding and productive career.
Professional Development	Individuals must acquire both technical and non-technical skills necessary in professional life, including an understanding of the unspoken rules and behaviors of practice.
Leadership Development	Skills and attitudes must be developed to conduct research, lead people, and direct projects to thrive in STEM.

2.1 Recruiting

2.1.1 NAIRR Project Leader Applicants. SRP-NAIRR builds upon a program originally designed to match DOE national laboratory staff researchers with faculty and students who were unlikely to engage with the DOE laboratories otherwise [6]. Extending this idea to NAIRR led to recruiting existing NAIRR project leaders to train other faculty and students, not then engaged with NAIRR, on their projects and the use of NAIRR resources. NAIRR project leaders were recruited through direct electronic mail messages and communications through NAIRR community forums.

A total of 38 NAIRR project leaders responded to the call for SRP participation, representing NAIRR projects: Classroom (5), Pilot (26), and Resource allocation (7). Faculty principal investigator institutions ranged from Carnegie classification Research 1: Very High Research Spending and Doctoral Production, Research 2: High Research Spending and Doctoral Production, and Professions-focused Baccalaureate Small, as shown in Figure 2.

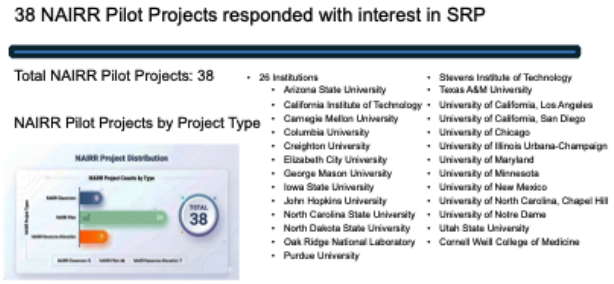


Figure 2: Call for Participation: NAIRR Pilot Projects

2.1.2 Faculty/Student Team Applicants. The SRP model utilizes faculty/student team cohorts to lower possible barriers for students at smaller and less research-intensive institutions who may be reticent to engage with high research organizations. The students enter into the research experience with a faculty member with whom they are already familiar, a factor which can provide certain levels of psychological assurance and confidence [3, 9]. This relationship can play a critical role in developing a sense of belonging in a research community.

As a programmatic issue, belonging is a process that requires intention. Encompassed in the SRP aims, belonging is a dynamic relational process co-produced through interpersonal interactions and can be influenced by leadership behaviors [1]. In that regard, faculty/student teams or cohorts have been identified as effective in expanding STEM participation and improving self-efficacy through building a sense of epistemic belonging [8].

This approach also provides the advantage of including a research trained faculty member on the team who is already familiar with the student(s) and can play a transitional and bridging role, collaborating with the project leader, and assisting in supervising the student(s).

Thirty-nine faculty/student teams applied for SRP which included 58 students to accompany the faculty. The teams represented 19 states, 31 institutions, and 7 department types, as shown in Figure 3.

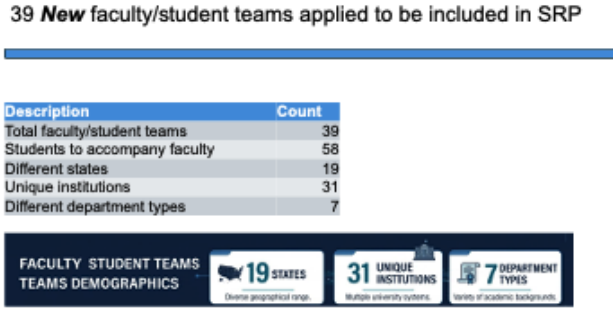


Figure 3: Call for Participation: Faculty/Student Teams

2.1.3 Independent Student Applicants. Independent students were recruited using electronic mail and social media posts through the Sustainable Horizons Institute mailing list and social media sites, as well as through partner organizations reaching over 12,000 individuals. One hundred forty-two students applied on their own, representing 33 states, 69 academic institutions, and 52 different majors. Student applicant institutions ranged from community colleges, predominantly undergraduate teaching institutions, and doctoral degree-granting institutions, as shown in Figure 4.

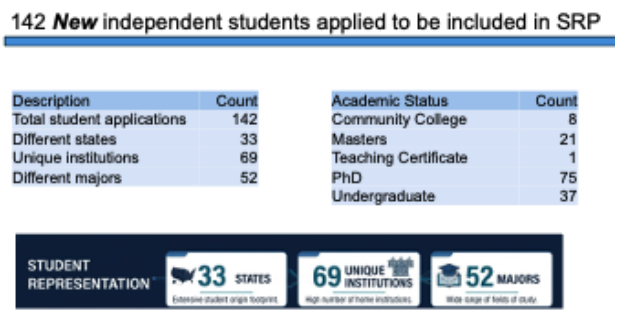


Figure 4: Call for Participation: Independent Students

2.2 Virtual Matching Workshop

The virtual matching workshop is a core component of SRP-NAIRR, occurring over four days with sessions each day ranging from one to two hours in length and providing successively more intensive opportunities for participants to interact and explore possible collaborations. The workshop includes a welcome, lightning talks, exploratory sessions for informal followup, and one-on-one meetings to consider possible projects, culminating in the designation of summer research preferences. The 2025 workshop was co-located with matching for DOE national laboratories and the High Performance Software Foundation (HPSF). Participation by track is shown in Table 2, with a total of 171 participants in the matching workshop.

SRP employs a deliberate two-way matching design. Potential matches are identified when the faculty/student and the NAIRR researcher both select each other. This approach creates some competition for matches while also empowering individuals. NAIRR

Table 2: Virtual Matching Workshop Participation by Track

Track	Number of Participants
Project PI: NAIRR	20
Project PI: DOE Lab	18
Project PI: HPSF	5
Faculty	24
Students of Faculty	28
Independent Students	76
Total	171

researchers must compete for students, and students must compete for projects; all the while, no one is assigned a match in which they have not expressed an interest. Two-way matching begins the formation of deeper, meaningful collaborative relationships.

A total of 45 faculty and students were matched with NAIRR projects including 8 faculty with 12 students and 25 independent students.

2.3 Research Experiences

Carrying out research tasks allows for increased mastery such that the research experience itself represents an opportunity to develop science self-efficacy [8]. Funded NAIRR projects fall into four main categories: Healthcare and Biomedical AI, Smart Infrastructure and Transportation, Energy and Climate Systems, and Foundational AI, Safety, and Education. After matches were identified, participants developed brief project plan proposals based on their interests relative to projects being conducted in these areas, describing the team members (project leaders, faculty, and/or students), motivation, goals, approach/methodology, expected outcomes, computing resource requirements, participation mode, collaboration plan, and period of performance. The project plan proposals were reviewed for support by an external review committee comprised of cyberinfrastructure researchers and related professionals who evaluated the technical merit and the potential developmental and professional impact on the faculty and students.

A total of 40 faculty and students were funded to collaborate with 14 NAIRR project PIs including 21 independent students, 7 faculty, and 12 students of faculty. The student academic status was 63.6% doctoral with master and undergraduate students representing 18.2% each. Thirty-four percent of students and 57% of faculty were first generation scholars and 14% of students and 17% of faculty were from Established Program to Stimulate Competitive Research (EPSCoR) states, as shown in Figure 5. (EPSCoR states in the U.S. are those that have been identified by the government as historically receiving less federal research and development funding.)

Virtual office hours, question and answer sessions, and a kick-off event prepare participants for the summer research experience. The participants complete progress reports throughout the summer. A mid-summer event provides a professional development activity allowing participants to share, practice, and receive feedback on their research progress. At the end of the summer, sessions are held for participants to practice presenting lightning talks and to receive

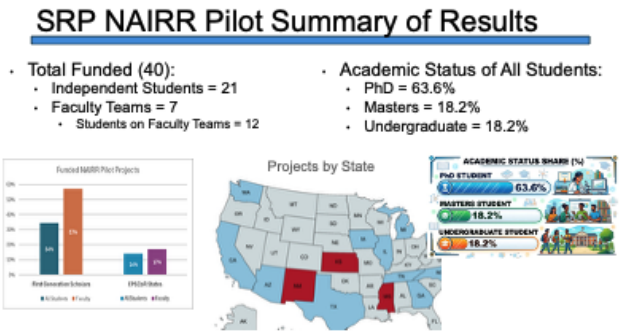


Figure 5: NAIRR Summer Research Matches

feedback and suggestions for improvements to their presentations. During an End of Summer Celebration event, participants share their final results and engage in interactive discussions.

2.4 CULTIVATE Conversations seminars

Community building is a central theme for SRP and begins in the matching workshop and continues throughout the summer and the following fall and spring. Monthly Catalyzing Ubiquitous Learning Through InnoVATive and Engaging (CULTIVATE) Conversations seminars stimulate learning, maintain regular contact with the community, and extend summer relationships. The sessions include introductions to the activity and topic, interactive facilitated conversations, and discussions of what was learned and reflections on their experiences. The activities are coordinated by SRP staff and all faculty, students, and NAIRR researchers are invited. Fall 2026 CULTIVATE Conversations seminars include 1) Taking the Next Steps in Utilizing NAIRR Assets and 2) Next Steps in Your Academic Journey.

2.5 Research Presentations

After the summer experience, faculty and students have the opportunity to present their research results at the Society of Industrial and Applied Mathematics (SIAM) Computational Science and Engineering professional conference, through the Building Engagement (BE) program [12]. This kind of activity is critical for helping students develop their professional standing and confidence in a mainstream professional society. They experience critical mass in the BE groups, while also participating in what may be their first conference experience. BE offers activities aimed at extending and deepening students' connections with NAIRR researchers.

BE aims to include students early in their academic careers, including community college and early undergraduates. For many participants, BE is their first time attending a cyberinfrastructure professional conference, or any scientific conference for that matter. To make the experience more accessible, a variety of activities are offered to demystify the often very dense technical material and create a welcoming environment. A small number of community volunteer faculty, graduate students, and professionals serve as guides or leaders for focused activities that engage students in conference venues, such as poster sessions, technical presentations, and networking, career, professional, and leadership development.

Pre-event sessions and participation in conference sessions serve to strengthen the learning including technical discussions and tutorials. Training and mentoring are other important components included in BE activities as described in Table 3.

Table 3: BE Activities

Activity	Description
Welcome and Question-Answer Session	Virtual welcome session six weeks prior to the start of the conference, overview of activities, introduction to the guided affinity groups.
Orientation	Introduction of participants to the program and other BE members. Conference agenda and program events are reviewed. Includes structured networking activities with other participants, conference organizers, SIAM leadership, and BE committee. Held onsite prior to conference student orientation and reception.
Guided Affinity Groups	Learning groups designed to help students get more out of conference sessions. Led by volunteer professional society members, these groups explore conference topics from an entry level perspective. Affinity groups meet virtually one month prior to the conference, meet daily each morning before the conference sessions start, and attend conference sessions together. Each group presents what was learned from the experience during a BE Wrap up session at the end of the week.
Mentoring	Mentoring engages students with mentors to encourage career development and professional incorporation. Volunteer mentors are recruited through the conference registration system and then paired with mentees based on questionnaires. Guided affinity group leaders also serve as mentors and often recruit their colleagues to visit the morning meetings and serve as additional mentors.
Presentation Tips and Lightning Talks	Session to learn effective presentation tech-niques and tips. Participants share their work in a "lightning round" session and address questions from the audience and receive constructive feedback.
Technical Posters	BE participants are encouraged to present posters in the conference poster session. Assistance with preparations is provided in the Lightning Talks session.
Wrap Up Session	Participants present what they learned during the week and through the Guided Affinity Group, and engage in general networking.

2.6 Individual Accomplishments

Regarding accomplishments, note that SRP faculty alumni, which includes those from non-research intensive institutions (e.g., Bronx Community College, Texas State University, and University of South Dakota), received NSF Early Career and DOE awards in the Reaching a New Energy Sciences Workforce, Early Career, and Funding for Accelerated, Inclusive Research programs, as well the Presidential Award for Excellence in Science, Mathematics, and Engineering Mentoring. Also, student participants have been hired full time at several DOE laboratories; decided to pursue graduate degrees; won best paper and poster awards; been awarded the NSF Graduate Research Fellowship and DOE Computational Science and Stockpile Stewardship Graduate Fellowships; presented their research at venues such as the Supercomputing Conference; and participated in the Oak Ridge National Lab Artificial Intelligence Workshop and the Argonne Training Program on Extreme-Scale Computing.

3 SUMMARY AND CONCLUSION

Sustainable Research Pathways is an evidenced-based program designed to establish effective ways of cultivating AI talent and participation. More to the point, specifically in the U.S., it is poised to address pressing challenges facing the country. The approach and features of the SRP program constitute mechanisms for bridging gaps in access to NAIRR such that it potentially can develop into a new paradigm for expanding STEM access and participation more generally. While making significant progress in broadening access and making AI accessible to all, SRP-NAIRR also has the potential to bring a wide spectrum of perspectives to the development of AI models and methods. SRP offers a path to building strong connections among NAIRR researchers and institutions with individuals across the U.S., including those who typically would not have access nor opportunities to engage in related research.

The SRP approach to expanding NAIRR capacity offers an effective programmatic means to strengthen opportunities to democratize, increase the variety of talent, improve capacity, and advance safe, secure, and trustworthy AI. The program stimulates changing perspectives and practices in AI research through participating NAIRR projects and new and emerging researchers–a crucial activity for AI efforts today and in the future.

ACKNOWLEDGEMENTS

Our thanks to the U.S. National Science Foundation for support through Awards Number (FAIN): 2515632 and 2609667.

REFERENCES

[1] M. Antoniadou and M. Crowder (Eds.). 2026. *Introduction to Belonging in the Workplace.* Edward Elgar Publishing, Northampton, MA.
[2] B. Christian. 2020. *The Alignment Problem: Machine Learning and Human Values.* W. W. Norton & Company.
[3] H. A. Kobulnicky and D. A. Dale. 2016. A Community Mentoring Model for STEM Undergraduate Research Experiences. *Journal of College Science Teaching* 45, 6 (2016), 17–23.
[4] M. A. Leung. 2018. Developing Sustainable Methods for Broadening Participation: Transforming Mainstream Science and Technology Communities through Normalization of Inclusion. *American Behavioral Scientist* 62, 5 (2018).
[5] M. A. Leung, S. Crivelli, and D. L. Brown. 2019. Sustainable Research Pathways: Building Connections across Communities to Diversify the National Laboratory Workforce. In *Collaborative Network for Engineering and Computing Diversity (CoNECD).* Washington, DC.
[6] M. A. Leung and C. L. McNeely. 2015. Opening Doors to Communities of Practice: Programmatic Interventions for Inclusion in the Computing Sciences. In *Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT).*
[7] M. A. Leung and C. L. McNeely. 2015. Opening Doors to Educational Attainment: A Programmatic Model for a More Inclusive Engineering Community. (2015). Submitted.
[8] M. M. C. Musgrove, M. E. Ko, J. N. Schinske, and L. A. Corwin. 2024. Broadening Participation in Biology Education Research. *Life Sciences Education* 23, 1 (2024).
[9] National Academies. 2019. *The Science of Effective Mentorship in STEMM.* National Academies Press, Washington, DC.
[10] National Artificial Intelligence Research Resource. 2023. Strengthening and Democratizing the U.S. Artificial Intelligence Innovation Ecosystem: An Implementation Plan for a National Artificial Intelligence Research Resource.

[11] Office of Advanced Cyberinfrastructure. 2021. *Blueprint for Cyberinfrastructure Learning and Workforce Development.* Technical Report. National Science Foundation, Washington, DC.

[12] Sustainable Horizons Institute. 2024. Building Engagement Program. https://shinstitute.org/the-building-engagement-be-program/ Accessed: 04 12 2025.

[13] V. Tinto. 1987. *Leaving College: Rethinking the Causes and Cures of Student Attrition.* University of Chicago Press, Chicago, IL.

[14] K. Topping. 1996. The Effectiveness of Peer Tutoring in Furthering Higher Education: A Typology and Review of the Literature. *Higher Education* 32, 3 (1996), 321–345.

[15] A. Torres. 2000. Rethinking the Model. In *Access Denied: Race, Ethnicity, and the Scientific Enterprise.* Oxford University Press, New York, 219–221.

[16] J. C. H. Trippi. 1991. Counseling Effects on African American College Student Graduation. *Journal of College Student Development* 32, 4 (1991), 342–349.

[17] R. Wilson. 2000. Barriers to Minority Success in College Science, Mathematics, and Engineering Programs. In *Access Denied: Race, Ethnicity, and the Scientific Enterprise.* Oxford University Press, New York, 193–206.

[18] L. Wolf-Wendel. 1998. Models of Excellence: The Baccalaureate Origins of Successful European American Women, African American Women and Latinas. *Journal of Higher Education* 69 (1998), 141–186.

T^3-CIDERS Train-the-Trainers: Strengthening CI Skills for Cybersecurity Research and Education through Iterative Improvements in Second Cohort

Wirawan Purwanto
Old Dominion University
wpurwant@odu.edu

Mohan Yang
Texas A&M University
mohanyang@tamu.edu

Peng Jiang
University of Nebraska Omaha
pjiang@unomaha.edu

Dorothy X. Parry
Old Dominion University
dparr005@odu.edu

Belle Li
Purdue University
li4808@purdue.edu

Kristin Herman
Millersville University
kristin.herman@millersville.edu

Nolan Lovett
Old Dominion University
nlove002@odu.edu

Shanan Chappell Moots
Old Dominion University
schappel@odu.edu

Masha Sosonkina
Old Dominion University
msosonki@odu.edu

Hongyi Wu
The University of Arizona
mhwu@arizona.edu

ABSTRACT

T^3-CIDERS is a multi-year train-the-trainer program that develops university faculty, researchers, and students as future trainers (FTs) that are equipped to integrate advanced cyberinfrastructure (CI) skills, including high-performance computing (HPC), machine learning, big data analytics, and parallel programming, into cybersecurity research and education. This paper describes the iterative redesign of T^3-CIDERS from its first cohort (2024) to its second cohort (2025–2026), motivated by systematic evidence collected through pre/post surveys, daily reflections, observations, and interviews. Challenges identified in Cohort 1, such as unequal participant preparedness, participant fatigue from a dense schedule, limited interactivity, and insufficient community building drove a set of evidence-based program improvements. For Cohort 2, the pre-training phase was substantially expanded to cover all six technical modules and a foundational pedagogy module. The in-person institute was restructured for more substantial interactivity and engagement, adopting a rhythm of mini-lectures, hands-on CI learning, guided discussions, and studio-based lesson development. Community-building activities were strengthened through social events during the institute, structured mentorship meetings, and a new community webinar series. Early findings from Cohort 2, drawing from reflections and direct feedback from the participants, show increased participation in the program beyond the institute, as well as confidence and eagerness to teach CI to others. This paper offers a reproducible model of iterative, evaluation-driven redesign for HPC education and train-the-trainer programs.

KEYWORDS

Cyberinfrastructure, Cybersecurity, Research, Train-the-trainer, Pedagogy, HPC, Parallel Computing, Big Data, Machine Learning, Non-degree Training, Hands-on

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/5

1 INTRODUCTION

Cybersecurity is a rapidly expanding field, crucial not only in information technology but across our tech-driven world [15]. Cyberattacks have become increasingly common and sophisticated in today's digital landscape, sparing no individual or organization from potential threats [12, 13]. Cybersecurity research and practice increasingly depend on advanced cyberinfrastructure (CI) capabilities—encompassing big data analytics, artificial intelligence (AI), high-performance computing (HPC), and parallel computing—to detect threats, train large-scale models, and process massive datasets at speed. This escalating challenge is compounded by a critical shortage in the cybersecurity workforce. There is a shortage of 4–5 million cybersecurity professionals globally according a 2024 ISC2 study [3]. Despite the growing importance of these CI tools in the state-of-the-art cybersecurity research and practice, there has been a persistent and systemic gap existing between CI availability and CI readiness among the cybersecurity workforce. One notable example is that traditional cybersecurity curricula at many institutions do not incorporate meaningful training in these CI methods. Closing this gap requires not merely more training resources or events, but a scalable strategy for producing qualified trainers who can bring CI skills into their own academic communities.

T^3-CIDERS, short for "Train-The-Trainer Approach to Fostering CI- and Data-Enabled Research in CyberSecurity" [14], is a train-the-trainer program to increase the adoption of advanced cyberinfrastructure (CI) and data skills into the fabric of research and education in cybersecurity and cyber-related disciplines. Rather than training individual learners directly at scale, T^3-CIDERS trains cohorts of "future trainers" (FTs)—faculty, researchers, and students—who are then equipped and obligated to conduct locally tailored CI training at their own institutions. This multiplier effect positions each cohort as a *node* in a growing network of CI-competent cybersecurity educators. T^3-CIDERS promotes the acquisition and propagation of expertise in CI technologies such as HPC, big data analytics, and machine learning/AI among the participants and their respective communities. The program empowers them to conduct more effective cybersecurity research. It aims to develop trainers in a community of next-generation cybersecurity researchers and educators who are proficient in CI technologies.

This paper focuses on the iterative redesign of T^3-CIDERS between its first cohort (2024) and its second cohort (2025–2026). Following the model of evidence-based program improvement [1], the design team systematically collected evaluation data from Cohort 1, identified specific challenges and pain points, and translated them into concrete structural changes for Cohort 2. We describe both cohorts, detail the redesign decisions and their rationale, and present early evaluation findings from Cohort 2.

2 OVERVIEW OF THE T^3-CIDERS PROGRAM

2.1 Vision and Goal

The overarching goal of T^3-CIDERS is to accelerate state-of-the-art research and development in cybersecurity and cyber-related fields by (1) preparing competent trainers to broaden the utilization of advanced CI in these disciplines, and (2) fostering a CI-enabled cybersecurity research community of practice [10, 11]. As a train-the-trainer program, T^3-CIDERS is designed to be synergistic with the teaching and research activities of its participants. Over time, the accumulation of FT cohorts across institutions would produce a diverse, self-sustaining community of practice of CI-competent cybersecurity researchers and educators.

2.2 Target Audience

FTs are recruited from university faculty, researchers, students, and practitioners with backgrounds in cybersecurity and cyber-related fields. Basic computer programming proficiency is a prerequisite for joining the program. Having prior experience with CI topics (HPC, AI, etc.) is beneficial but not required, as the program provides foundational instructions in these areas. FTs are organized into "FT units" pairing one faculty or researcher with one or two students, to encourage collaborative lesson development and peer learning.

2.3 Program Structure

T^3-CIDERS engages FTs through a year-long program with six interconnected phases [10]:

- Phase 1 – Pre-training: A self-paced, virtual training providing CI skills baseline and foundational pedagogical concepts, enabling all FTs to enter the in-person institute with shared foundational knowledge.
- Phase 2 – Summer/Winter Institute: An intensive, in-person, multi-day training combining hands-on CI technical instruction with pedagogy and instructional design practice, culminating in FT teaching demonstrations.
- Phase 3 – Learning Engagement: Ongoing virtual meetups, research webinars, and peer learning activities sustained throughout the academic year.
- Phase 4 – Local Training: FT units are required to conduct locally tailored CI training at their home institutions, with support from the T^3-CIDERS team.
- Phase 5 – K-12 Outreach: Optional, encouraged activities to promote CI and cybersecurity awareness among K-12 students.
- Phase 6 – "Module-X": Community-driven development of new CI training modules addressing emerging research needs, informed by FT's and community's input.

The first two phases serve as a condensed "boot camp" to build CI and pedagogical competencies. The T^3-CIDERS program leverages the six hands-on CI modules [7] previouly developed within the "DeapSECURE" training project [2, 6, 8, 9]. The remaining phases represent the sustained outcomes: local dissemination, community growth, and curriculum expansion.

The T^3-CIDERS project is designed for multiple successive cohorts. Cohort 1 completed the 2024 Summer Institute at Old Dominion University (ODU). Cohort 2 completed the 2026 Winter Institute at the University of Arizona. A third hybrid cohort is being planned. Beyond the funded period, an online adaptation (T^3-Online) will open the program to the broader public as an open educational resource (OER).

3 COHORT 1 (YEAR 2024): DESIGN, IMPLEMENTATION, AND EVALUATION

The T3-CIDERS design team adapted the Successive Approximation Model (SAM) [1] as its guiding instructional design process, consisting of preparation, iterative design, and iterative development phases [10]. A distinctive feature of the design process was the use of AI-generated learner personas to inform inclusive design decisions in the absence of direct access to the first cohort of participants [5]. Beyond that, the design process also involved multiple rounds of content and context analyses, including task analyses and learning objective development.

3.1 Recruitment, Pre-training, and Summer Institute

The Cohort 1 call for participation was disseminated in Spring 2024 through Virginia's research computing professional network (MARIA—Mid-Atlantic Research Infrastructure Alliance), the Commonwealth Cyber Initiative, and cybersecurity researcher networks. A total of 20 FTs participated in the 2024 Summer Institute: 8 faculty and researchers and 12 students (graduate and undergraduate) [10]. Participants represented multiple states (New York, North Carolina, Texas, Virginia, and Washington), and a range of institution types. The majority of FTs indicated primary research or study areas in

cybersecurity, followed by AI/ML, computer science, data science, and electrical and computer engineering.

Pre-training was delivered in Spring 2024 as a four-week self-paced virtual course on Canvas, covering two DeapSECURE modules: *Introduction to HPC* and *Dealing with Big Data*—each with lecture materials, readings, case studies, and hands-on Jupyter notebook activities. The 2024 Summer Institute was held July 29–August 2 at ODU in Norfolk, Virginia. The five-day program covered CI technical topics (Machine Learning, Deep Learning, Cryptography, and Parallel Programming), foundations of education and instructional design, lesson planning studio sessions, expert panels (cybersecurity practitioners and K-12 educators), and a capstone teaching demonstration by each FT unit.

3.2 Post-Institute Local Training and Module-X

Following the Summer Institute, the team continued to engage FTs virtually through newsletters, announcements via Discord, and optional meetings with the team.

3.2.1 Local Training. Many of the FT units from Cohort 1 successfully delivered their local training activities following the Summer Institute attendance, as showcased in Table 1. The total number of impacted learners across all the local training events is conservatively estimated at about 75.

Table 1: Local training and outreach activities conducted by Cohort 1 FT units.

Event Description	Type	Date(s)	Num. Learners
CTRL+ALT+ELITE – Level Up Your Skills With the Command Line	High school outreach workshop	April 2025	16
Federated Learning hands-on lesson	Graduate course lab	April 2025	9
Generative AI Capture the Flag	Virtual contest	Unknown	Unknown
Introductory workshops on AI/ML	Hands-on workshops	November 2024	25
Transformer Model Architecture	Seminar	April 2025	Unknown
Intro to HPC & Intro to Machine Learning	Hands-on workshops	Fall 2024	Unknown
Tutorial on Linux Command Line	Video tutorial	Summer 2025	N/A

3.2.2 Module-X: Federated Learning. The "Module-X" component of T^3-CIDERS encourages FTs to develop and contribute additional training materials on emerging CI and/or cybersecurity topics. One of the FT units developed such a new module in close collaboration with the T^3-CIDERS team. After completing the core CI training during the 2024 Summer Institute, this FT group developed a small hands-on lab on the topic of *federated learning* (FL) and delivered this lab in a graduate-level machine learning class (Spring 2025). Afterwards, they generalized and extended their lab materials to create a hands-on introductory FL lesson. FL is a branch of privacy-preserving machine learning, which is well aligned with their research interests. FL enables collaborative model training across decentralized clients without sharing the raw data, making it highly relevant for privacy-sensitive cybersecurity applications. The module was designed as a scaffolded, hands-on workshop that balances conceptual understanding with practical experimentation. The module was developed over the Fall 2025 semester and refined into a deployable instructional unit, which was demonstrated during the second cohort's Winter Institute.

Instructional Design: The FL module follows a progressive structure, guiding learners from foundational concepts to more advanced topics. It begins with the core FL workflow, including local model training and server-side aggregation, and then introduces challenges such as data heterogeneity and system variability. Subsequent components explore optimization strategies and security considerations, enabling participants to understand both performance and robustness aspects of FL.

Hands-on Implementation: To support experiential learning, the FL module is implemented through Jupyter notebook-based exercises executed on a HPC platform. Participants interact with real training pipelines, modify configurations, and observe the impact of factors such as non-IID (independent and identically distributed) data and algorithmic choices on model convergence and accuracy.

3.3 Evaluation

T^3-CIDERS evaluation employed a mixed-methods design aligned with Kirkpatrick's Four-Level Evaluation approach [4] including pre-surveys, daily reflections, post-institute surveys, observations, and post-cohort interviews. Initial results from Cohort 1 pre/post surveys (20 and 16 respondents, respectively) showed encouraging improvements in FT self-assessed CI knowledge, motivation, and confidence to teach. Qualitative feedback from daily reflections highlighted participant enthusiasm and creative planning for local training activities. However, the design team's retrospective analysis, through review of planning documents, implementation records, and weekly meeting notes, along with deeper qualitative feedback from interviews surfaced a set of specific challenges that would guide the redesign for the next cohort:

- *Unequal Preparedness*: FTs arrived with widely varying CI backgrounds; pre-training (HPC + Big Data) did not fully bridge the gap for participants with minimal prior CI experience, especially for advanced topics at the Institute.
- *Fatigue and Pacing*: The dense five-day schedule with heavy lecture content led to participant fatigue and reduced content absorption, particularly in the second half of the week.
- *Limited Interactivity*: Content was often delivered in lecture format with insufficient time for discussion, peer learning, and active application. Studio sessions, while valuable, needed greater scaffolding.
- *Student Disengagement Post-Institute*: Nearly half of student FTs became disengaged after the Summer Institute, resulting in low participation in local teaching activities and ongoing learning engagements.
- *Insufficient Community Building*: Participants reported insufficient opportunities to build interpersonal connections and research networks during the Institute; community engagement post-Institute was limited.

- *Operational Bottlenecks*: Centralized team responsibilities, late announcements, and Discord platform underutilization created logistical friction. Lesson plan completion was deferred post-Institute, creating an ongoing "assignment burden".

4 COHORT 2 REDESIGN AND IMPLEMENTATION

The transition from Cohort 1 to Cohort 2 was guided by a set of evidence-based redesign principles: (1) responsive design—using Cohort 1 evaluation to inform meaningful improvements while avoiding wholesale program overhaul; (2) balancing depth and accessibility—expanding pre-training so the in-person institute can engage more deeply; (3) building sustainable trainer communities through mentorship and structured community platforms; and (4) operationalizing pedagogical improvements so that training design itself models best pedagogical practices.

4.1 Operational Improvements

Cohort 2 recruitment began significantly earlier, enabling wider outreach, a larger candidate pool, and a more deliberate selection process. The team employed a structured voting process among PIs to evaluate applicants, ultimately selecting 20 FTs from a large pool of candidates. Operationally, the team distributed responsibilities more explicitly through role specifications, and shifted primary community communication from Discord to email (which proved more reliable for participants). Centralized master planning documents and a shared Google Drive workspace improved coordination.

4.2 Pre-training Expansion

One of the most substantial structural changes for Cohort 2 was the significant expansion of the pre-training phase. Whereas Cohort 1 pre-training covered only two modules (HPC and Big Data) with full hands-on activities, Cohort 2 pre-training expanded to cover all six core DeapSECURE CI topics and added a dedicated pedagogy module, for an estimated total of approximately 16 hours of content. Table 2 summarizes the revised pre-training curriculum.

Table 2: Cohort 2 Pre-training Modules

Module	Topic	Type	Hours
1	Big Data (Jupyter, Pandas)	Lecture video*	1
	Jupyter & Pandas	Hands-on	4
2	HPC (Unix shell, SLURM)	Lecture video*	1
	Shell & SLURM job scheduler	Hands-on	4
3	Educational Foundation	Lecture video*	2
4	Machine Learning	Lecture video*	1
5	Neural Networks	Lecture video*	1
6	Cryptography	Lecture video*	1
7	Parallel Programming	Lecture video*	1

The rationale for this expansion was to ensure that all FTs arrive at the in-person Winter Institute with at least a conceptual familiarity with each CI topic, regardless of their background. Modules 1 and 2 retain full hands-on activities; Modules 4–7 provide lecture-based overview videos and carefully curated reading materials without the hands-on activities to avoid overwhelming participants before the institute, while still broadening their exposure. The new pedagogy module (Module 3) provides foundational concepts in instructional design and adult learning, enabling FTs to connect pedagogical theory to practice more readily during the institute's studio sessions. Complex readings and external platform materials used in Cohort 1 were removed or simplified. Each module includes reflection questions; FTs are asked to respond with brief video podcasts or written reflections, promoting engagement and metacognitive processing before the institute.

4.3 Winter Institute (2026): Redesign and Implementation

The T3-CIDERS Winter Institute (WI), held from January 5–9, 2026 at the University of Arizona, served as a central program for Cohort 2. Building on evaluation findings and observations from the inaugural 2024 Summer Institute, the 2026 WI was strategically redesigned to address key challenges, including uneven participant preparedness, limited engagement, and insufficient time for hands-on and instructional development activities. The key changes include: (1) moving lecture-based contents to the pre-training phase; (2) increased time for hands-on activities and lesson development; (3) increased social activities; and (4) reduction of the institute time from five to four days.

4.3.1 Strategic Shift to Asynchronous Pre-Training. A major modification was the relocation of all the technical lectures into the expanded pre-training phase. Compared to the 2024 Summer Institute, where lectures and hands-on activities were co-located during the event, the 2026 WI moved foundational instructions to an asynchronous format prior to the institute. Participants completed approximately 16 hours of pre-training, which included recorded lecture-based overviews of four additional core CI modules. By front-loading more contents in the expanded pre-training, participants arrived at the WI with a more consistent baseline of knowledge. This shift also allowed the institute to be shortened from five to four days without sacrificing quality. This strategy significantly reduced the need for lecture-heavy sessions during the institute and enabled a stronger focus on more interactive and high-impact learning activities, as well as FTs' own lesson development rather than broad content coverage.

4.3.2 On-Site Program Structure. With lectures largely moved to pre-training, the on-site institute adopted a refined daily structure that emphasized hands-on practice and pedagogical development (see Table 3 for the detailed timetable of the contents):

- *Morning sessions* were dedicated to intensive, hands-on technical CI training aligned with the four core CI modules now added into pre-training: machine learning, neural networks, cryptography (including homomorphic encryption), and parallel programming. Mini-lectures were added as a refresher on the key points, as well as guided discussions. Rather than introducing new concepts through lectures, these sessions focused on experiential learning, including tool usage, problem-solving, and guided exploration of CI workflows. This approach allowed participants to deepen their understanding through active engagement.

- *Afternoon sessions* shifted toward pedagogical training and instructional design. Participants were introduced to key teaching strategies through concise lectures, followed by collaborative discussions and structured "studio time." During these sessions, FT units worked to develop their own training materials using the provided templates, including learning objectives, instructional activities, and evaluation plans.

Table 3: Cohort 2 Winter Institute Contents

Category	Day 1	Day 2	Day 3	Day 4
CI discussion	Mini-lecture (tech hands-on orientation); Group discussions (CI application for research)			
CI hands-on	Machine learning	Neural networks	Cryptography	Parallel programming
Pedagogy lecture & hands-on	Education basic, objectives, proposal	Backward design; lesson planning; Module-X demo	Instructional design	Activity showcase (1)
Studio time	Training proposal	Lesson planning; evaluation plan	Activity design	Activity showcase (2)

A notable addition to the 2026 WI was the inclusion of a "Module-X" session, delivered by a former FT from the 2024 cohort. (The design and development of this module have been discussed in the previous section.) This FT presented the newly built federated learning module on Tuesday afternoon (day 2). Beyond its technical content, this new module also serves as a proof-of-concept for sustainable curriculum evolution, demonstrating how participants from earlier cohorts can contribute new modules back into the program. Overall, the Module-X development highlights T^3-CIDERS' emphasis on modular, participant-driven curriculum design and reinforces its goal of equipping future trainers with both technical depth and instructional capacity.

To reinforce progress and reduce post-institute workload, participants were required to produce incremental deliverables throughout the week. This ensured that by the conclusion of the institute, most FT units had developed substantial portions of their lesson plans, minimizing the need to complete assignments after returning to their home institutions.

4.3.3 Community Building and Engagement. To strengthen participant engagement and address challenges observed in the 2024 cohort, the 2026 WI emphasized collaborative and interactive learning throughout the program. Structured development time enabled participants to iteratively design and refine their training materials in the studio during afternoon sessions with real-time feedback, peer collaboration, and continuous instructional support. We design the pedagogical activities to increase student FT's engagement by encouraging them to take a lead role during lesson design and planning, teaching demonstrations, as well as post-institute local training. In addition, the 2026 institute incorporated more social and networking activities aimed at fostering a community of practice.

4.3.4 Transition to Local Implementation. The Winter Institute concluded with a transition from centralized training to decentralized implementation. Equipped with near-complete lesson plans and strengthened technical and pedagogical skills, participants were prepared to carry out local training activities at their home institutions.

Preliminary observations suggest that this redesigned structure enabled earlier and more effective deployment of local teaching activities compared to the previous cohort. This outcome aligns with the core objective of T3-CIDERS: to empower participants to act as trainers and disseminate CI knowledge within their own academic and research communities.

5 THE SECOND COHORT INITIAL EXPERIENCE (2025–2026)

5.1 Participant Recruitment and Selection

The strengthened recruitment effort for Cohort 2025 yielded a substantially larger and more competitive applicant pool than the inaugural cohort. A total of 42 individuals from 15 institutions applied, of whom 25 were faculty or staff and 17 were students. Following a structured review process in which the principal investigators evaluated and voted on the applications, acceptance offers were extended to 23 individuals. Only 20 were able to attend the Winter Institute. The cohort was organized into ten FT units, with pairing decisions made primarily based on home institution or geographic proximity to facilitate practical collaboration in lesson planning and local training delivery. Most units consist of two FTs; one unit has three members, and one is a single member. Six of the ten units include at least one student member.

5.2 Cohort Demographics and Institutional Profile

The 20 participating FTs represent a broad cross-section of academic roles. Among non-student FTs, seven hold research tenure-track faculty positions (including one professor, one associate professor, and five assistant professors), three are teaching faculty or lecturers in non-tenure-track roles, one is a postdoctoral associate, and two are staff members. The seven student FTs include four Ph.D. students, two M.S. students, and one undergraduate student. In terms of gender, 14 FTs identified as male, five as female, and one preferred not to answer.

From the academic disciplines standpoint, the majority of FTs (nine) are affiliated with computer science departments, with three from electrical and computer engineering, and the remainder drawn from information systems, applied mathematics, cybersecurity, and research computing services. Seven FTs indicated research interests spanning both AI (or a subfield of AI) and cybersecurity—reflecting a stronger alignment between participants' technical backgrounds and the program's core curriculum than was seen in the first cohort.

The ten home institutions span ten states: Arizona, California, Colorado, Hawaii, Iowa, Indiana, Massachusetts, Nebraska, Oregon, and Virginia (Figure 3). Six institutions carry an R1 doctoral research classification, two are R2, and two are classified as research colleges and universities. Eight are large public universities; two are private institutions, one small and one medium-sized. Three of the ten

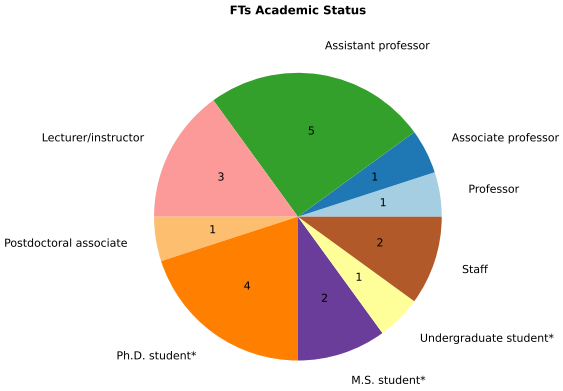


Figure 1: Academic status distribution of the 2025-2026 cohort.

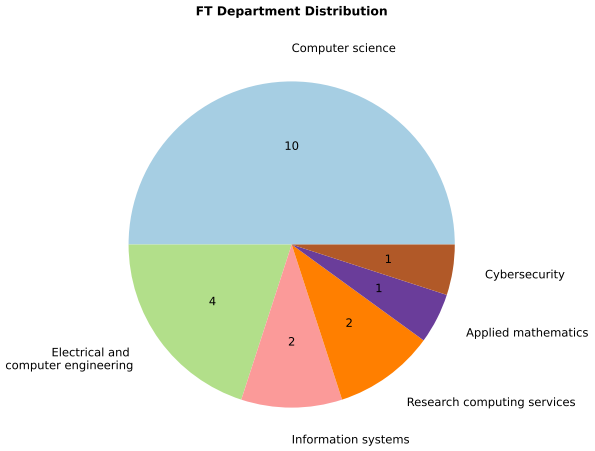


Figure 2: Academic departments indicated by the FTs (2025-2026 cohort).

institutions hold one or more minority-serving institution (MSI) designations — reflecting the program's commitment to broadening participation in CI-enabled cybersecurity research across diverse institutional contexts.

5.3 The 2026 Winter Institute

The 2026 Winter Institute was held at the University of Arizona in Tucson in January 2026, marking the first time the in-person institute was hosted outside of ODU. The four-day program reflected the structural and pedagogical redesigns described in Section 4: a compressed and better-paced schedule, a consistent daily rhythm alternating between technical and pedagogical content, and a deliberate emphasis on interactive discussion and hands-on activity over extended lecture. Each morning opened with a brief technical recap followed by guided discussion prompts that asked FTs to connect CI concepts to their own research or institutional contexts, before moving into hands-on lab work. Afternoons were devoted

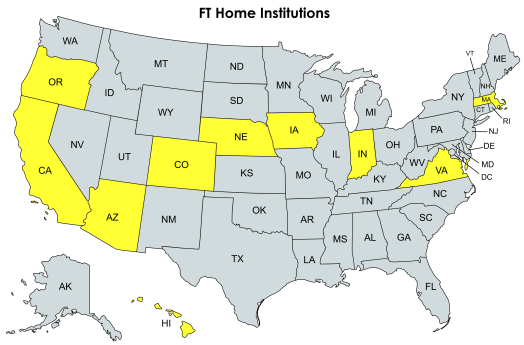


Figure 3: Geographical distribution of the Cohort 2 FTs' home institutions (created with mapchart.net).

to pedagogical sessions and structured studio time, during which FTs worked on their lesson plans using scaffolded templates — with the explicit goal of leaving the institute with substantially complete plans ready for local deployment. Social and community-building activities were woven throughout the four days, including: a welcome reception on the first evening, a cohort dinner on the fourth evening, organized tours of research facilities, and an optional hiking tour the day after the institute. These provided informal settings in which FTs could build personal connections alongside their technical and pedagogical work, thus facilitating a more cohesive and connected cohort that may extend the engagement beyond the institute.

5.4 Initial Reactions and Findings

Formal analysis of the pre- and post-institute survey data and interview transcripts for Cohort 2 is ongoing and will be reported comprehensively in a forthcoming publication. However, early indicators from participant reflections and unsolicited public feedback suggest a positive reception.

Several FTs shared their experiences on LinkedIn following the Winter Institute. Their posts point to three recurring themes: First, participants highlighted the value of the pedagogical content—specifically the theory-informed, evidence-based approach to instructional design—as something that offered new and practically applicable perspectives on curriculum development and course design. Second, they emphasized the utility of the T^3-CIDERS technical materials as resources directly transferable to their own CI- and AI-related courses. Third, participants consistently pointed to the professional networking and potential research collaboration opportunities afforded by the cohort experience as meaningful program outcomes in their own right.

These early signals are consistent with the program's intended impact at Kirkpatrick's first two levels—reaction and learning—and provide an encouraging precursor to the more systematic evaluation findings that are in process.

5.5 Remaining Cohort 2025 Activities

With the Winter Institute complete, FT units have moved into the local training and ongoing engagement phases of the cohort year. During this period, the T^3-CIDERS team is engaged in structured one-on-one mentorship meetings between FT units and project team members to provide individualized guidance for lesson plan refinement and local training preparation. As of the time of writing, five FT units have already delivered their local training activities—a notably higher early completion rate compared to Cohort 2024, which the project team attributes in large part to the redesigned studio sessions that enabled FTs to leave the institute with more fully developed lesson plans. The remaining units have confirmed plans for training delivery in the near term.

On the community engagement front, the inaugural T^3-CIDERS community webinar (April 1, 2026) drew participation from both cohort members and external attendees. Subsequent webinars are planned, with upcoming sessions to feature presentations by cohort members themselves—both deepening FTs' experience as presenters and extending the program's reach to a growing external audience. These activities collectively represent the ongoing learning engagement phase that sustains community momentum between the intensive institute experience and the longer-term community of practice that T^3-CIDERS aims to cultivate.

6 T^3-ONLINE: TOWARD A SCALABLE TRAIN-THE-TRAINER

The first two cohorts of T^3-CIDERS have demonstrated that an intensive, in-person train-the-trainer model can effectively build CI competencies and seed a community of practice. However, the cohort-based, in-person format inherently constrains program reach: each institute accommodates roughly twenty FTs, requires substantial travel and hosting resources, and is bounded by the institutional and regional networks through which recruitment is conducted. Building on the lessons learned from two in-person cohorts, the T^3-CIDERS team is developing an asynchronous, self-paced online adaptation of the program, known as "T^3-Online", to be released publicly as an Open Educational Resource (OER), thus extending the program's reach to faculty, researchers, students, industry practitioners, K-12 educators, and international participants who cannot be part of the in-person cohorts.

6.1 Design and Implementation of T^3-Online

The design of T^3-Online follows the same SAM-based iterative process used for the in-person program (see Fig. 4), proceeding through preparation, iterative design, and iterative development phases. The preparation phase draws directly on the rich evaluation dataset accumulated across Cohorts 1 and 2—demographic profiles, prior-experience data, and identified pain points—to construct empirically grounded learner personas and inform content chunking, cognitive load management, and scaffolding strategies suited to asynchronous delivery. The Cohort 2 pre-training curriculum and the federated learning Module-X serve as the primary content foundation, with materials re-mediated for self-paced formats where needed. Through its stages of development, the T^3-Online materials will be advanced from the *alpha* stage, to *beta*, and finally to *gold*.

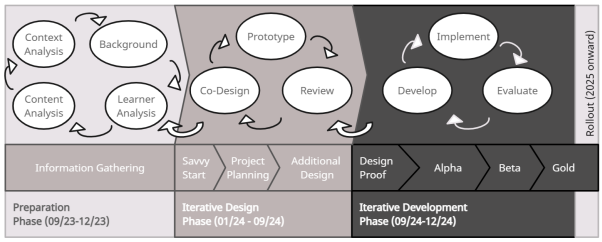


Figure 4: The iterative design and development process of T^3-CIDERS, which is roughly subdivided into three major phases following the Sucessive Approximation Method (SAM): Preparation, iterative design, and iterative development. T^3-Online will follow the same pattern, building upon the experiences of T^3-CIDERS cohorts.

The planned module structure follows a consistent architecture: short video lectures introducing core concepts; embedded or linked Jupyter notebooks, or access to an HPC sandbox environment for hands-on practice; self-check quizzes for formative assessment; reflection prompts (text or video) modeled on the Cohort 2's metacognitive activities; downloadable lesson-plan templates that learners progressively complete throughout the program; and a capstone teaching-demonstration submission that mirrors the culminating activity of the in-person institute.

Pedagogical scaffolding from the Cohort 2 redesign will be carried forward and adapted. The in-person cohort daily rhythm of mini-lectures, guided discussions, and extended studio time becomes a sequence of recorded mini-lectures with embedded discussion prompts, threaded discussion forums for asynchronous peer exchange, structured studio activities completed independently or in self-organized FT units, and scheduled live "office hours" providing synchronous touchpoints with the project team without requiring synchronous attendance from all learners.

6.2 Cohort 3: A Hybrid Pilot of T^3-Online

Cohort 3 will be conducted as a hybrid pilot that simultaneously serves two purposes: it delivers the train-the-trainer experience to a new group of FTs, and it provides the formative evaluation needed to advance T^3-Online from *beta* to the final *gold* stage.

The hybrid Cohort 3 experience will comprise three phases: (1) *Pre-institute phase*: FTs complete the full *beta* version of T^3-Online asynchronously over a defined window prior to the in-person event. (2) *Local-prep phase*: FT units draft lesson plans and training materials using T^3-Online's *beta* templates, with virtual mentorship checkpoints from the project team. (3) *In-person phase*: a condensed two-day institute focused almost entirely on FT teaching demonstrations, peer feedback, lesson-plan refinement, and community-building, rather than on content delivery.

The compression of the in-person institute from four days (Cohort 2) to two days is justified by the front-loading of technical and pedagogical contents into the online program. The remaining in-person time is reserved for the highest-bandwidth activities: live teaching practice, peer feedback, networking, and mentorship,

which Cohort 1 and 2 evaluation consistently identified as the most valuable elements of the program.

6.3 Rollout of T^3-Online

At the conclusion of the project's current funding period by U.S. National Science Foundation, T^3-Online will be released publicly as an Open Educational Resource, freely accessible to the cybersecurity and broader CI education communities. Web-based delivery is intended to maximize dissemination. Sustainability beyond the grant period will depend in part on identifying institutionally-supported personnel to maintain content currency and to facilitate periodic Module-X additions contributed by the growing FT alumni community.

7 CONCLUSION

T^3-CIDERS demonstrates that a rigorous, evaluation-driven approach to train-the-trainer program design can meaningfully advance the adoption of advanced cyberinfrastructure skills in cybersecurity research and education. Across its first two cohorts, the program has prepared forty future trainers drawn from diverse institutions nationwide, producing a growing body of locally tailored CI training activities that have collectively reached nearly two hundred learners at their home institutions. The program's core commitment to iterative improvement is demonstrated by translating the challenges identified from the first cohort into concrete redesigns of the pre-training curriculum, the in-person institute structure, the pedagogical approach, and the community engagement strategy. Early indicators from Cohort 2 suggest stronger pre-institute readiness, higher post-institute engagement with local training activities, and greater participant enthusiasm for teaching CI to others. The effort toward building a community-of-practice is also beginning to take shape, as cross-cohort connections form, a Module-X contribution, and a new webinar series opens the program to participants beyond enrolled cohorts. Looking ahead, the development of T^3-Online as an open educational resource will extend the program's reach to faculty, researchers, students, and practitioners beyond the current project funding's lifetime. The upcoming hybrid Cohort 3 model ensures that the online adaptation is validated against real learners before the public release. Together, these efforts constitute a scalable, reproducible, and community-sustained model for closing the CI skills gap in cybersecurity—one trainer, one institution, and one cohort at a time.

ACKNOWLEDGMENTS

The T^3-CIDERS program is collaboratively developed and delivered by Old Dominion University (ODU), the University of Arizona, the University of Nebraska Omaha, and Texas A&M University, with funding from the U.S. National Science Foundation CyberTraining program, grants #2320998 and 2320999. The authors graciously acknowledge the support from the University of Arizona for hosting the 2026 Winter Institute, and ODU Research and Cloud Computing group for the use of their Wahab HPC cluster for conducting the CI training activities. The Wahab cluster is supported in part by National Science Foundation's grant CNS-1828593, "MRI Acquisition: A Reconfigurable Computing Infrastructure Enabling Interdisciplinary and Collaborative Research in Hampton Roads". We also gratefully acknowledge the contributions of our teaching assistants to the success of this program: Jiawei Chen, Chun-lin Huang, Ted Birkland, and Junyi Jiang. Generative AI (Claude) has been used to synthesize the initial draft of the paper based on outlines, detailed notes, and other documents (which are not AI-generated) provided by the authors. The process is heavily constrained to produce texts that are grounded on these input materials. The generated texts have been edited and reviewed by the authors to ensure that this paper accurately represent the actual work of the authors.

REFERENCES

[1] Michael Allen and Richard Sites. 2012. *Leaving ADDIE for SAM: An agile model for developing the best learning experiences.* Association for Talent Development.

[2] Bahador Dodge, Jacob Strother, Rosby Asiamah, Karina Arcaute, Dr. Wirawan Purwanto, Dr. Masha Sosonkina, and Dr. Hongyi Wu. 2022. DeapSECURE Computational Training for Cybersecurity: Third Year Improvements and Impacts. http://www.modsimworld.org/papers/2022/MSVSCC_2022_InfrastructureSecurityMilitary.pdf

[3] ISC2. 2024. The State of the Cybersecurity Workforce. ISC2 Insights. https://www.isc2.org/Insights/2024/10/Cybersecurity-Workforce-INSIGHTS-October-2024 Accessed 2026-05-18.

[4] James D Kirkpatrick and Wendy Kayser Kirkpatrick. 2016. *Kirkpatrick's Four Levels of Training Evaluation.* Association for Talent Development.

[5] Nolan Lovett, Mohan Yang, Kristin Herman, Belle Li, Masha Sosonkina, Wirawan Purwanto, Peng Jiang, and Michael Hongyi Wu. 2025. Leveraging Large Language Models to Create Learner Personas for Training Design. *The Journal of Applied Instructional Design* 14, 3 (2025). doi:10.59668/2223.22718

[6] Wirawan Purwanto, Bahador Dodge, Karina Arcaute, Masha Sosonkina, and Hongyi Wu. 2024. DeapSECURE Computational Training for Cybersecurity Students: Progress Toward Widespread Community Adoption. *The Journal of Computational Science Education* 15 (2024), 2–9. Issue 1. https://doi.org/10.22369/issn.2153-4136/15/1/1

[7] Wirawan Purwanto, Issakar Doude, Yuming He, Jewel Ossom, Qiao Zhang, Liwuan Zhu, Jacob Strother, Rosby Asiamah, Bahador Dodge, Orion Cohen, Jiawei Chen, Chunyu Hu, Dorothy Parry, Peng Jiang, Masha Sosonkina, and Hongyi Wu. 2024. DeapSECURE Lesson Modules. https://deapsecure.gitlab.io/lessons/

[8] Wirawan Purwanto, Yuming He, Jewel Ossom, Qiao Zhang, Liuwan Zhu, Karina Arcaute, Masha Sosonkina, and Hongyi Wu. 2021. DeapSECURE Computational Training for Cybersecurity Students: Improvements, Mid-Stage Evaluation, and Lessons Learned. *The Journal of Computational Science Education* 12 (2021). Issue 2. https://doi.org/10.22369/issn.2153-4136/12/2/1

[9] Wirawan Purwanto, Hongyi Wu, Masha Sosonkina, and Karina Arcaute. 2019. DeapSECURE: Empowering Students for Data- and Compute-Intensive Research in Cybersecurity through Training. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)* (Chicago, IL, USA) *(PEARC '19)*. ACM, New York, NY, USA, Article 81, 8 pages. doi:10.1145/3332186.3332247

[10] Wirawan Purwanto, Mohan Yang, Peng Jiang, Shanan Chappell Moots, Masha Sosonkina, and Hongyi Wu. 2025. T3-CIDERS: Train-the-Trainer and Community Building to Increase Cyberinfrastructure Adoption in Cybersecurity Research and Education. In *Proceedings of the Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration* (Columbus, OH, USA) *(PEARC '25)*. ACM, New York, NY, USA, Article 21, 9 pages. doi:10.1145/3708035.3736021

[11] Wirawan Purwanto, Mohan Yang, Peng Jiang, Masha Sosonkina, and Hongyi Wu. 2025. T3-CIDERS: Fostering a Community of Practice in CI- and Data-Enabled Cybersecurity Research Through a Train-the-Trainer Program. *The Journal of Computational Science Education* 16, 1 (2025), 16–19. doi:10.22369/issn.2153-4136/16/1/4

[12] M. F. Safitra, M. Lubis, and H. Fakhrurroja. 2023. Counterattacking cyber threats: A framework for the future of cybersecurity. *Sustainability* 15, 18 (2023), 1–32. doi:10.3390/su151813369

[13] K. L. Snider, R. Shandler, S. Zandani, and D. Canetti. 2021. Cyberattacks, cyber threats, and attitudes toward cybersecurity policies. *Journal of Cybersecurity* 7, 1 (2021), 1–11. doi:10.1093/cybsec/tyab019

[14] T3-CIDERS Team. 2025. *T3-CIDERS Project Website.* Retrieved 2025-05-18 from https://sites.wp.odu.edu/t3-ciders/

[15] W.J. Triplett. 2023. Addressing Cybersecurity Challenges in Education. *International Journal of STEM Education for Sustainability* 3, 1 (2023), 47–67. doi:10.52889/ijses.v3i1.132

In Search of a Better Search: Using RAG on Cornell Virtual Workshop Content

Zilu Wang
Cornell University
Center for Advanced Computing
zw427@cornell.edu

Susan Mehringer
Cornell University
Center for Advanced Computing
shm7@cornell.edu

Christopher J. Cameron
Cornell University
Center for Advanced Computing
cjc73@cornell.edu

ABSTRACT

The Cornell Virtual Workshop (CVW) is a continually updated HPC training platform with about forty roadmaps and more than two thousand pages, organized in a three-level hierarchy of roadmaps, topics, and pages. The CVW's traditional keyword search works poorly for users who lack domain terminology or want navigational guidance rather than a specific page. We describe a RAG-based chatbot that routes each query to a retriever matched to the appropriate level of the content hierarchy, ensuring that navigational queries return information about CVW structure and factual queries return specific page content. The chatbot is built with LangGraph and uses a hybrid BM25 and FAISS retrieval pipeline with cross-encoder reranking. A fixed-path pipeline provides prompt-injection defense, query rewriting, post-generation URL verification, and graceful handling of out-of-scope queries.

KEYWORDS

RAG, LLM, HPC, AI chatbot, Hybrid Retrieval, Hierarchical Retrieval, Query Routing, HPC Training, Workforce Development

1 INTRODUCTION

The Cornell Virtual Workshop [4] (CVW) is an online asynchronous training platform for high-performance computing developed by the Cornell Center for Advanced Computing, currently containing about forty roadmaps, with about 300K page views per year. We help people find the right content by organizing material into roadmaps, topics, and pages; adding keyword tags to each topic; and providing two types of traditional searches. In our observation, the quality of results from traditional site-specific searches depends on using specific terminology and keywords that may be unfamiliar to new learners.

We developed a Retrieval-Augmented Generation (RAG) chatbot based on CVW content to answer natural language queries without requiring CVW-specific terminology. The chatbot is intended to direct learners to the CVW materials, not replace them, so it provides a single, accurate answer that includes both the answer and a link to the in-context information.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/6

We chose to use a RAG solution because it allowed us to use a general-purpose large language model (LLM) to answer queries using only CVW content and reduce AI hallucinations [6].

Our chatbot contributes the following:

- a hierarchy-aware RAG design that routes each query to a level-filtered retriever (roadmap, topic, or page), ensuring that navigational queries return structured overviews and factual queries return specific page content
- a fixed-path LangGraph pipeline with explicit nodes for prompt-injection defense, query rewriting, multi-level retrieval, post-generation URL verification, and graceful handling of vague and out-of-scope queries

2 RELATED WORK

Several recent efforts use LLMs to support HPC education and training. Within the HPC-ED project, Morsy et al. [8] demonstrate the potential of using LLMs as a quality assurance tool to review HPC training catalog metadata at scale.

Other works have focused on adding a retrieval layer between the LLM and HPC data. Miyashita et al. [7] extend RAG by allowing the system to execute a set of predefined commands and feed the output back as context for user queries.

Closest to our work, Bondapalli et al. [1] present AskHPC, a RAG-based chatbot for HPC users that uses facility user guides and scheduler and programming documentation as the corpus. In contrast, our chatbot is built around CVW, a hierarchically structured corpus, to route queries to roadmap-, topic-, and page-level retrievers.

3 SYSTEM DESIGN

CVW users ask questions at all three levels of its content hierarchy: some want a learning path, others a specific page, and others a topic overview. This section describes how we encode that structure in the chatbot's ingestion, retrieval, and routing components.

3.1 Corpus: Cornell Virtual Workshop

Cornell Virtual Workshop content is organized in a three-level hierarchy. At the top, a *roadmap* is a learning sequence on a single subject area, consisting of 5–10 topics listed in a suggested order. Each *topic* contains a set of individual *pages* that cover specific concepts, examples, or exercises. Each roadmap and topic includes structured metadata, such as a description, learning objectives, and prerequisites; roadmaps also specify system requirements. CVW pages follow the URL structure `cvw.cac.cornell.edu/<roadmap>/<topic>/<page>`.

3.2 Ingestion and Chunking

CVW pages are written as ASP.NET Razor pages with JSON, C#, and a database back end. The files used to build the corpus contain a mix of HTML markup and Razor directives. Before any file is processed, we use Python to strip out these elements, except for `<pre><code>` blocks, which are preserved verbatim. We do so by replacing each code block with a sentinel placeholder, removing the HTML tags and Razor directives, and then restoring the code blocks.

We then split pages with a recursive character splitter into 512-character chunks with 64-character overlap, treating each as an indexable document. We chose 512 characters because CVW pages are prose-heavy, and smaller chunks generally improve semantic retrieval when matching against user queries. Future work may evaluate other chunk sizes.

In addition to the chunked page documents, we construct a single *synthetic summary document* for each roadmap and each topic. A roadmap summary contains the roadmap's description, learning objectives, prerequisites, and system requirements, followed by a flat list of its topics with brief descriptions. A topic summary contains the topic's description, objectives, prerequisites, and keyword list, followed by a flat list of its pages by title. These documents are indexed unchunked because splitting them would sever the relationship between labeled fields and their values, and would distribute the child listing across separate chunks. If the retriever did not return all the chunks, the LLM could not accurately summarize roadmap and topic contents. The summaries are compact by design — typically a few hundred to roughly a thousand characters — so the cost of keeping them whole is low. Because each retriever is filtered to a single corpus level (roadmap, topic, or page), a generation node always receives either summary documents or page chunks.

Every chunk and summary document also carries structured metadata identifying its source, such as the roadmap and topic it belongs to, its CVW URL, and the topic description (or, for roadmap-level documents, the roadmap description).

3.3 Embeddings and Vector Store

In the embedding phase, all documents — both page chunks and synthetic summaries — are encoded using the `text-embedding-ada-002` embedding model from OpenAI. The resulting vectors are stored in a single FAISS (Facebook AI Similarity Search) index [3] shared across all three corpus levels. Level-filtered queries at retrieval time make this functionally equivalent to maintaining separate indexes per level, while simplifying index maintenance. The index is rebuilt from source only when a new rebuild token is set via an environment variable; otherwise it is loaded from disk on startup.

3.4 Pipeline

When a user sends a query, it passes through a pipeline implemented in LangGraph [5]. All queries follow a fixed sequence of nodes (Figure 1): Guard, Understand-and-Classify, one of Roadmap, Topic, Page, or Clarify, then URL-Verification, optionally Rate, then Log. Each node is a function that reads from and writes to a shared State object.

The initial Guard node strips invisible or non-printing Unicode characters from the input. It then checks the sanitized text with regular expressions against a set of common prompt-injection phrases, and also checks whether any base64 or hexadecimal runs in the query contain injection patterns when decoded. Queries shorter than 3 or longer than 2000 characters are also rejected. If any check fails, the pipeline exits immediately and returns a refusal without calling an LLM API.

The Understand node first rewrites the user query to produce a cleaned query that corrects grammatical or spelling errors and uses past conversations to form a complete question. For example, if a user first asks about Linux and then follows up with "How do I learn it?", the Understand node infers that "it" refers to Linux and produces a standalone query "How do I learn Linux?". The Understand node also generates a list of retrieval keywords, the main subjects of the query plus synonyms and related technical terms, that are likely to appear in relevant documents.

The Classify node routes the cleaned query to one of four nodes: Roadmap, Topic, Page, or Clarify. An LLM call at temperature 0 makes the routing decision on the cleaned query. The prompt defines the four routes: *roadmaps* for queries seeking a learning sequence or structured path; *topic* for queries that explicitly reference a CVW topic or ask for a topic-level overview; *page* for concrete how-to and reference questions, which is also the default when neither of the other labels applies; and *clarify* for out-of-scope, code-generation, and vague requests. Although originally implemented as two separate nodes, Understand and Classify were merged into a single LLM call to reduce round-trip latency.

The Clarify node handles queries that are out of scope or too vague to retrieve against. If the query is still within the HPC scope but vague, the node responds with a brief clarifying question; if the query is entirely out of scope, it returns a one-sentence refusal.

In the Roadmap, Topic, and Page nodes, the retrieval keywords generated by the Understand node serve as the search query; if no keywords are produced, the cleaned query is used instead. The query is passed to the corresponding retriever described in Section 3.5, and the top-n reranked documents are spliced into the answer generation prompt. We developed the routing prompt through iterative manual inspection of the 30-case internal example set described in Section 5. The selected node sends the system prompt, the top-n reranked documents, and the cleaned query to the LLM, which produces an answer. If no documents are retrieved, a fixed message states that the CVW does not have the requested topic, without calling the LLM.

Before returning the answer to the user, the URL-Verification node normalizes URLs and eliminates hallucinated URLs by comparing against a URL allowlist generated from the indexed CVW corpus. It strips invalid URLs and returns the final answer to the user.

After generating an answer, the Rate node optionally scores the response against a rubric, and the Log node stores the exchange in a local database.

3.5 Retrieval and Generation

As described in the pipeline section (Section 3.4), relevant queries are routed to one of three retrievers: roadmap, topic, or page. Each

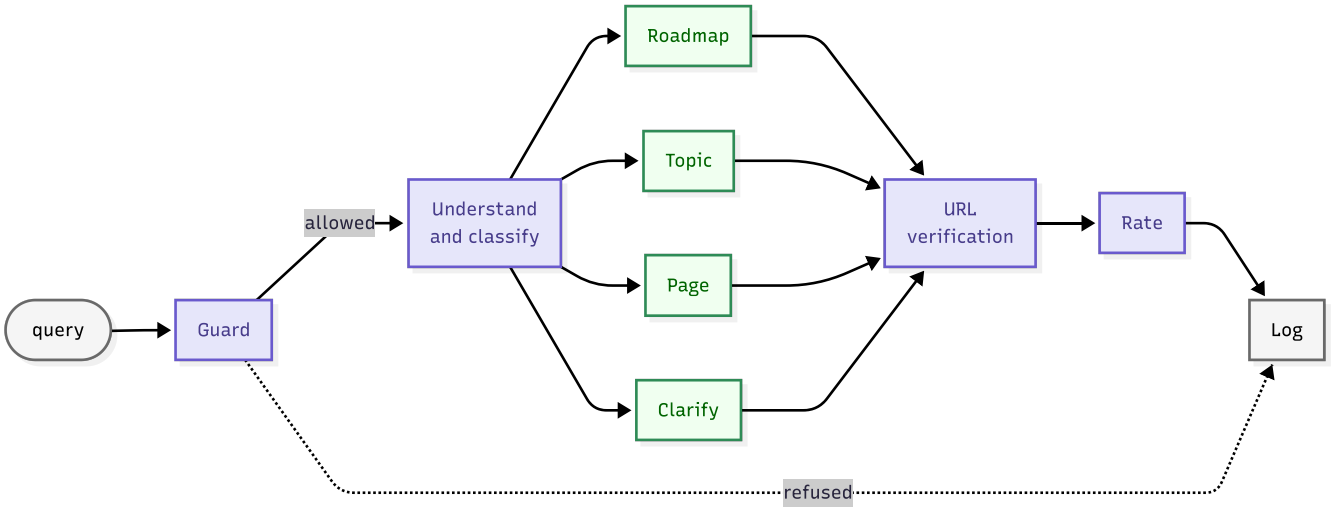


Figure 1: The CVW Chatbot pipeline. Queries come from users into Guard, where they are either allowed for further understanding and classification, or refused. Allowed queries are routed to one of four nodes, which all converge on URL verification before rating and logging.

retriever operates on the single FAISS index built in Section 3.3 and applies a `level` filter to ensure the returned documents remain within its scope. For example, the roadmap retriever returns only documents whose `level` field equals `"roadmap"`. Using multiple retrievers allows the chatbot to respond to roadmap- and topic-level queries because it hides details that would otherwise distract the LLM; the roadmap and topic retrievers operate on the synthetic summary documents, whereas the page retriever operates on the chunked page documents.

BM25 (Best Matching 25) [9] and FAISS are then run on the filtered documents, and a weighted ensemble combines their results, with BM25 at 0.4 and FAISS at 0.6. BM25 is a lexical ranking function that matches on exact terms, whereas FAISS retrieves documents whose embeddings are semantically closest to the query. The result is passed to FlashRank [2], which reranks the documents and returns the top n documents. For the roadmap retriever, we fetch 20 documents and rerank to the top 10; for the topic retriever, 30 and 20; for the page retriever, 30 and 20.

We group the reranked documents by their roadmap and topic, and prepend each group's metadata, such as description, objectives, prerequisites, and requirements, above the chunks. We then splice the grouped documents into a level-specific answer-generation prompt and pass it to OpenAI's GPT-5.4 at a temperature of 0.2. Each retrieved document includes a `Source URL:` line that we stored in metadata, which the generation prompt copies verbatim into the answer. A post-generation check then removes any URL not found in the indexed corpus, so every URL cited in a response points to a page in the indexed corpus.

3.6 Prompts

Each of the Understand, Classify, Roadmap, Topic, Page, Clarify, and Rate nodes includes a prompt that guides the LLM to perform the node-specific task. These prompts define much of the pipeline's behavior. Below, we highlight several prompt choices that shape the chatbot's answers.

- In the Roadmap, Topic, and Page nodes, the prompt instructs the LLM to append the URL from the source document's `Source URL:` to any answer it generates as a markdown-formatted URL with the page title as the link text. Together with the post-generation URL check, this ensures that a reference always exists to validate the answer.
- The prompt in the Page node also limits its answer to one or two sentences with CVW URLs, presented as a high-level teaser to encourage users to visit the link, rather than relying on the generated answer. We want to encourage users to discover and engage with the CVW material, rather than using the chatbot as a substitute.
- Several prompts also define out-of-scope behavior. Code-generation requests, for instance, are either routed to the Clarify node or refused outright when the query falls clearly outside CVW's HPC scope.

4 DEPLOYMENT

The chatbot is deployed as a search agent on the CVW website https://cvw.cac.cornell.edu. Depending on operating costs, we may impose access limits in the future by restricting access to logged-in users or imposing a per-session rate limit.

LLMs are accessed through the Cornell AI Gateway, which provides the GPT-5.4 model from OpenAI and applies Cornell-managed content safety guardrails, including prompt-attack and harmful-content filters applied to answer-generation calls. The chatbot's LLM usage is billed per token, and operators can monitor usage via the Cornell AI Gateway's LiteLLM dashboard. Using test data, we estimate that the cost will be approximately $0.05 per query.

The chatbot code runs on an on-premises CAC Red Cloud instance. The conversation history between users and the chatbot is

redacted of personally identifiable information before being stored in a database. It is used only for multi-turn conversations and development purposes.

We have observed limited user engagement. Initially, we restricted access to logged-in users and observed that users opened the agent chat page but did not interact with it or log in. We are reviewing the impact of both a login requirement and a privacy notice as potential barriers to adoption.

5 EVALUATION

To evaluate the chatbot, we constructed an internal test set of 30 cases, each targeting a specific case type: page-level queries, roadmap-level queries, topic-level queries, vague queries, out-of-scope queries, code requests, prompt injections, and content not covered by the CVW. We sampled these cases from the conversation log collected during development.

The Rate node evaluates chatbot responses on a 1–5 scale using the rubric below. The evaluator LLM first identifies which of three legitimate response types the query requires: (A) a cited answer for factual or learning-path queries within CVW's scope; (B) a clarifying question for vague or ambiguous in-scope queries; or (C) a one-sentence scope refusal for out-of-scope or adversarial queries. It then scores the match between the required type and the actual response, penalizing type mismatches and execution failures:

- **5 — Excellent.** Response type matches the query's need and execution is clean: a cited answer is accurate, brief, and includes a real CVW URL as a teaser; a clarifying question is focused and narrows intent; a scope refusal is one clean sentence with no hedging.
- **4 — Good.** Right type, minor issues: slight verbosity, a citation pointing to a related but not exact page, or a clarifying question broader than ideal.
- **3 — Mixed.** Right type but weak execution: tutorial-style prose rather than a teaser, a weak or generic citation, a hedged refusal that implies the bot might know elsewhere, or a graceful recovery from a routing error.
- **2 — Poor.** Wrong type for the query: a clarifying question when the query was directly answerable, answering an out-of-scope query as in-scope, or refusing an answerable in-scope query.
- **1 — Unacceptable.** Hallucinated CVW content, fabricated URLs, system prompt leakage, capitulation to prompt injection, or clearly unsafe output.

The bot generally handles in-scope queries well, producing concise answers with valid CVW URLs as citations:

> **User:** "What is the best way to transfer files to Vista?"
> **Bot:** "For Vista, CVW points to the standard secure transfer tools—**scp, sftp, and rsync**—as the main ways to move files from your local or remote machine, with the best choice depending on your file size and workflow. It also notes that transfers should follow TACC good-conduct guidelines.
> *Source:* Getting Started on Vista: Storing and Moving Data"

An example of roadmap recommendations:

> **User:** "I'm just getting started with HPC, what do I need to know?"
> **Bot:** "Here are a few good starting points for getting oriented to HPC and parallel computing:
> - Parallel Programming Concepts and High Performance Computing gives a broad introduction to HPC fundamentals, parallel terminology, and how to think about designing code for many-core systems.
> - Getting Started on Frontera fits if you want a practical overview of how an HPC system works, including environment setup, storage, compiling, and running jobs.
> - Data Transfer is useful for learning one of the most common HPC workflow tasks: moving code and data between your local machine, cloud storage, and remote systems.
>
> *Source:* Parallel Programming Concepts and High Performance Computing
> *Source:* Getting Started on Frontera
> *Source:* Data Transfer"

An example of an out-of-scope query:

> **User:** "Give me a recipe for burritos?"
> **Bot:** "That question is outside Cornell Virtual Workshop's scope — CVW focuses on HPC, parallel programming, MPI, CUDA, Slurm, Linux, Python performance, storage, and related computing topics."

The examples above are representative of the 30-case test set. In the test set, all injection attempts were refused. Twenty-four of the twenty-six queries with an expected route were routed correctly. The errors were vague queries routed to the roadmap retriever rather than the clarification node. The self-evaluation average is 3.6/5 for rateable responses. The LLM tended to evaluate itself more harshly than the human rater, and since the LLM-as-rater was not aware of the CVW content, it tended to give low ratings to refusals to answer out-of-scope questions that were plausible in-scope. A human review of the test query responses found that 25/30 responses were reasonably helpful; the human-identified error cases involved the LLM misunderstanding a question about the latest published material (2 cases), failing to fully cite the source (1 case), failing to return complete results (1 case), and failing to return results (1 case).

Once we have collected a meaningful volume of user queries, we will conduct a formal evaluation. Quantitatively, we will measure answer quality on a 1–5 scale using the LLM-as-judge method, and measure latency and cost per query as reported by the Cornell AI Gateway; we will measure routing accuracy separately by cross-checking answers with a human expert. Qualitatively, we will manually review low-rated answers identified by the Rate node, examine multi-turn conversations to assess behavior in a realistic setting, and use the results to identify pipeline weaknesses. The LLM-as-judge method is known to have several flaws, such as verbosity bias (LLMs favor longer responses) and self-enhancement bias (LLMs favor responses generated by the same model) [10]. A fully reliable evaluation will require manual review by a human expert.

6 FUTURE WORK

Future work falls into two areas. First, the chatbot is deployed but has seen limited user engagement to date. We plan to investigate barriers to adoption — including whether requiring a login or the presence of a privacy notice discourages use — and to improve chatbot response through prompt refinement based on real user queries. We plan to assess routing accuracy using a human-labeled test set as part of the post-deployment evaluation.

Second, while current configurations and parameters produce acceptable results, we chose the retrieval parameters and prompts heuristically rather than tuning them. We selected the BM25 weight (0.4) and FAISS weight (0.6) to prioritize semantic matches while still allowing keyword search. We chose the reranker model by comparing three FlashRank cross-encoders of different sizes (`TinyBERT-L-2`, `MiniLM-L-12`, and `MultiBERT-L-12`) on informal tests. The larger models introduced unacceptable response latency, so we selected `TinyBERT-L-2`. We set the `fetch_k` and `top_n` values based on the size of the CVW corpus at each level. A smaller candidate set is appropriate for the smaller roadmap corpus (about 40 documents). In comparison, the topic and page corpora are larger and benefit from a larger candidate set at the reranking stage. Future work will refine these configurations and parameters to reduce the cost per query and the response latency.

ACKNOWLEDGMENTS

This work was conducted with the support of the Cornell University Center for Advanced Computing and the Cornell AI Gateway.

REFERENCES

[1] Akhilesh Bondapalli, Huihuo Zheng, Oluwaseun T. Ajayi, Murat Keceli, Haritha Siddabathuni Som, Taylor Childers, Lisa Childers, Yasaman Ghadar, Michael Papka, Venkatram Vishwanath, and Rong Ge. 2025. AskHPC: A ChatBot for High Performance Computing User Support. In *Proceedings of the SC '25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC Workshops '25)*. Association for Computing Machinery, New York, NY, USA, 727–739. https://doi.org/10.1145/3731599.3767433

[2] Prithiviraj Damodaran. 2023. FlashRank, Lightest and Fastest 2nd Stage Reranker for search pipelines. https://doi.org/10.5281/zenodo.10426927

[3] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. (2024). arXiv:cs.LG/2401.08281

[4] Cornell Center for Advanced Computing. 2026. Cornell Virtual Workshop. Retrieved May 18, 2026 from https://cvw.cac.cornell.edu/

[5] LangChain AI. 2024. LangGraph. https://github.com/langchain-ai/langgraph. GitHub repository.

[6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:cs.CL/2005.11401 https://arxiv.org/abs/2005.11401

[7] Yusuke Miyashita, Patrick Kin Man Tung, and Johan Barthélemy. 2024. LLM as HPC Expert: Extending RAG Architecture for HPC Data. arXiv:cs.DC/2501.14733 https://arxiv.org/abs/2501.14733

[8] Habiba Morsy, Essence Toone, Charlie Dey, Zilu Wang, Mary Thomas, and David Joiner. 2026. HPC-ED: Testing Automated Agents to Assess the Quality of Training Resource Metadata. *The Journal of Computational Science Education* 17 (March 2026), 34–41. Issue 1. https://doi.org/10.22369/issn.2153-4136/17/1/5

[9] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3 (09 2009), 333–389. https://doi.org/10.1561/1500000019

[10] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. arXiv:cs.CL/2306.05685 https://arxiv.org/abs/2306.05685

A Documentation Catalog for the AI Ecosystem: Design of the ICICLE AI Training Catalog

Amit Vyas
Department of Computer Science, The Ohio State University
vyas.154@osu.edu

Carlos Guzman
Department of Computer Science, The Ohio State University
guzman.109@osu.edu

Neelima Savardekar
Department of Computer Science, The Ohio State University
savardekar.2@osu.edu

Mary Thomas
San Diego Supercomputer Center (SDSC), UC San Diego
mpthomas@ucsd.edu

Michelle Miller
University of Wisconsin–Madison
mmmille6@wisc.edu

Alfonso Morales
University of Wisconsin–Madison
morales1@wisc.edu

Joe Stubbs
Texas Advanced Computing Center (TACC), University of Texas at Austin
jstubbs@tacc.utexas.edu

Christian Garcia
Texas Advanced Computing Center (TACC), University of Texas at Austin
cgarcia@tacc.utexas.edu

Beth Plale
University of Oregon
plale@uoregon.edu

ABSTRACT

As artificial intelligence becomes increasingly important across scientific disciplines, there is a growing need for scalable platforms that support training, knowledge sharing, and workforce development. The ICICLE AI Institute, funded by the National Science Foundation, focuses on democratizing artificial intelligence and cyberinfrastructure through open education, community engagement, and shared software infrastructure.

This paper presents the design and architecture of the ICICLE Training Catalog and demonstrates how the platform supports broader community engagement and workforce development through practical use cases and user personas. The catalog aggregates educational materials, tutorials, and technical documentation from multiple institutions into a centralized and structured repository.

We describe the website design, documentation design principles, and early usage evidence from privacy-preserving Umami analytics. We also discuss how the Training Catalog supports workforce development and how it can be leveraged by different user groups, including students, researchers, educators, and research software engineers. Finally, we highlight lessons learned from developing a collaborative training platform across institutions and discuss future directions for expanding its impact.

KEYWORDS

Artificial Intelligence, Cyberinfrastructure, Open Source, Training Material, Research Software

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/7

1 INTRODUCTION

Artificial intelligence (AI) and cyberinfrastructure (CI) technologies are increasingly central to modern scientific discovery. The National Science Foundation-funded ICICLE AI Institute [1, 6] pursues new AI and CI technologies with the explicit goal of democratizing AI. The institute draws its challenges from three use-inspired domains—animal ecology, digital agriculture, and food access—and uses these challenges to develop software, tools, training materials, and workflows that make state-of-the-art AI technologies usable across scientific domains.

This work is accompanied by a growing need for accessible pathways through which researchers, educators, professionals, and students can learn to use AI tools and CI platforms effectively. Research computing communities commonly distribute training materials through learning management systems, workshop pages, single-project documentation sites, API portals, and institutional knowledge bases. Each of these channels is individually valuable, but each typically optimizes for a single audience or content type: a course site supports a scheduled or self-paced cohort, an API portal supports endpoint lookup, and a single-project documentation site supports users of a specific package. In multi-institutional AI and CI work, these separate channels can obscure how a software component, an educational module, an API, and a hosted service relate to one another.

To address this gap, ICICLE developed the ICICLE Training Catalog [2], which centralizes training resources produced by researchers, educators, and developers across the ICICLE ecosystem. By organizing these materials in a structured and searchable form, the platform broadens access to AI education and supports workforce development across institutions.

This paper describes the design and usage of the ICICLE Training Catalog. We present the platform's architecture, the conventions it uses to organize content, evidence of early adoption drawn from privacy-preserving analytics, and the ways in which it supports distinct user communities within the AI ecosystem.

2 WHAT IS ICICLE

ICICLE (Institute for Intelligent Cyberinfrastructure with Computational Learning in the Environment) [6] is a National Science Foundation (NSF) funded AI Institute focused on enabling the next generation of AI-powered cyberinfrastructure.

The institute brings together researchers, educators, and software engineers across universities, national laboratories, and computing centers. Its mission is to democratize artificial intelligence by making AI technologies accessible to scientists, educators, and students across disciplines.

ICICLE focuses on several key goals:

- Developing AI-powered cyberinfrastructure tools for scientific research
- Creating open educational resources for AI training
- Building reusable AI software components and workflows
- Supporting community-driven collaboration across institutions

To pursue these goals, the institute developed an open-source framework that makes materials findable for current users and approachable for new ones. We call the product the "ICICLE Training Catalog." The Training Catalog serves as a documentation source and as a pedagogical vehicle, connecting conceptual materials, hosted services, and reusable source code.

3 ICICLE TRAINING CATALOG

The ICICLE Training Catalog [2] is an open documentation platform that supports AI and CI education across a broad community of learners, educators, and researchers. It is designed as a bridge across the otherwise siloed channels that research computing communities use to distribute training material. The Training Catalog combines a documentation-site model for maintainability, the Diátaxis framework for consistent organization, metadata and thrust-based tags for cross-domain discovery, and privacy-preserving analytics for continuous improvement. Unlike a static list of workshop links or a single-project manual, it integrates software documentation, API specifications, educational materials, workshop resources, and source-code entry points into one searchable structure that supports workforce development in AI.

3.1 Building the Training Catalog

The Training Catalog is built using Docusaurus [5], a widely adopted open-source framework for creating documentation websites. Docusaurus enables content authoring in lightweight Markdown files, allowing contributors across institutions to easily create, maintain, and update documentation.

The architecture follows a static-site generation model with content version-controlled through repositories, enabling scalable and collaborative development. The platform integrates Umami [10], an open-source, privacy-preserving analytics solution deployed on TACC infrastructure via Tapis [9]. Umami captures aggregate engagement signals—page views, referrers, devices, browsers, operating systems, and navigation paths—without recording personally identifiable information. In Figure 1, the upper path depicts documentation ingestion and deployment; the Tapis Pods component (lower portion) hosts the Umami service and its PostgreSQL backend.

3.2 Structured Documentation with Diátaxis

To ensure consistency and usability, the ICICLE Training Catalog adopts the Diátaxis documentation framework [7], which organizes technical documentation into four user-centered categories:

- Tutorials
- How-to Guides
- Explanations
- References

Each documented resource includes a project overview together with one or more of these sections, enabling users to select content appropriate to their task (see Figure 2 for an example component page). The documentation also carries metadata such as last-updated timestamps, allowing users to assess the currency of the material.

Contributors can produce ICICLE-compatible README files through two complementary authoring pathways. The first is a repository-based template provided through the ICICLE GitHub workflow [4], which gives contributors a deterministic structure for project description, canonical tags, license information, references, acknowledgements, issue reporting, and Diátaxis sections. The second is an AI-assisted workflow in which contributors invoke the `icicle-readme-skill` [3] from AI-powered development environments such as Claude Code, Cursor, or Codex. In this mode, the skill guides the authoring or revision of a README so that it conforms to the ICICLE documentation standard while preserving project-specific technical content. Together, these pathways support both conventional template-driven authoring and interactive documentation development, lowering the barrier for contributors with different writing preferences and toolchains.

To operationalize this structure, a custom Python-based parser processes README files from individual projects and organizes them into Markdown files and directory structures compatible with Docusaurus. The parser extracts project metadata such as title, source repository, tags, last-updated information, and external service links; detects Diátaxis sections such as tutorials, how-to guides, explanations, and references; normalizes relative links and image paths; and emits Docusaurus-compatible pages and sidebar entries. When a README lacks required metadata or has ambiguous section headings, the parser reports warnings so maintainers can correct the source material before publication.

This automated pipeline ensures consistency across documentation while reducing manual effort. It also makes the Training Catalog easier to scale because contributors can continue to maintain project-level README files while the catalog build process produces a coherent public-facing site.

3.3 API Documentation

The platform's API documentation section presents OpenAPI v3 specifications for publicly available ICICLE endpoints, including Tapis-hosted services and project-specific APIs when component teams provide a specification. Integrated within Docusaurus, this section exposes endpoint descriptions, request and response schemas, authentication notes where applicable, and example calls. Examples are oriented toward common AI and cyberinfrastructure workflows, including command-line `curl`, Python, and JavaScript clients.

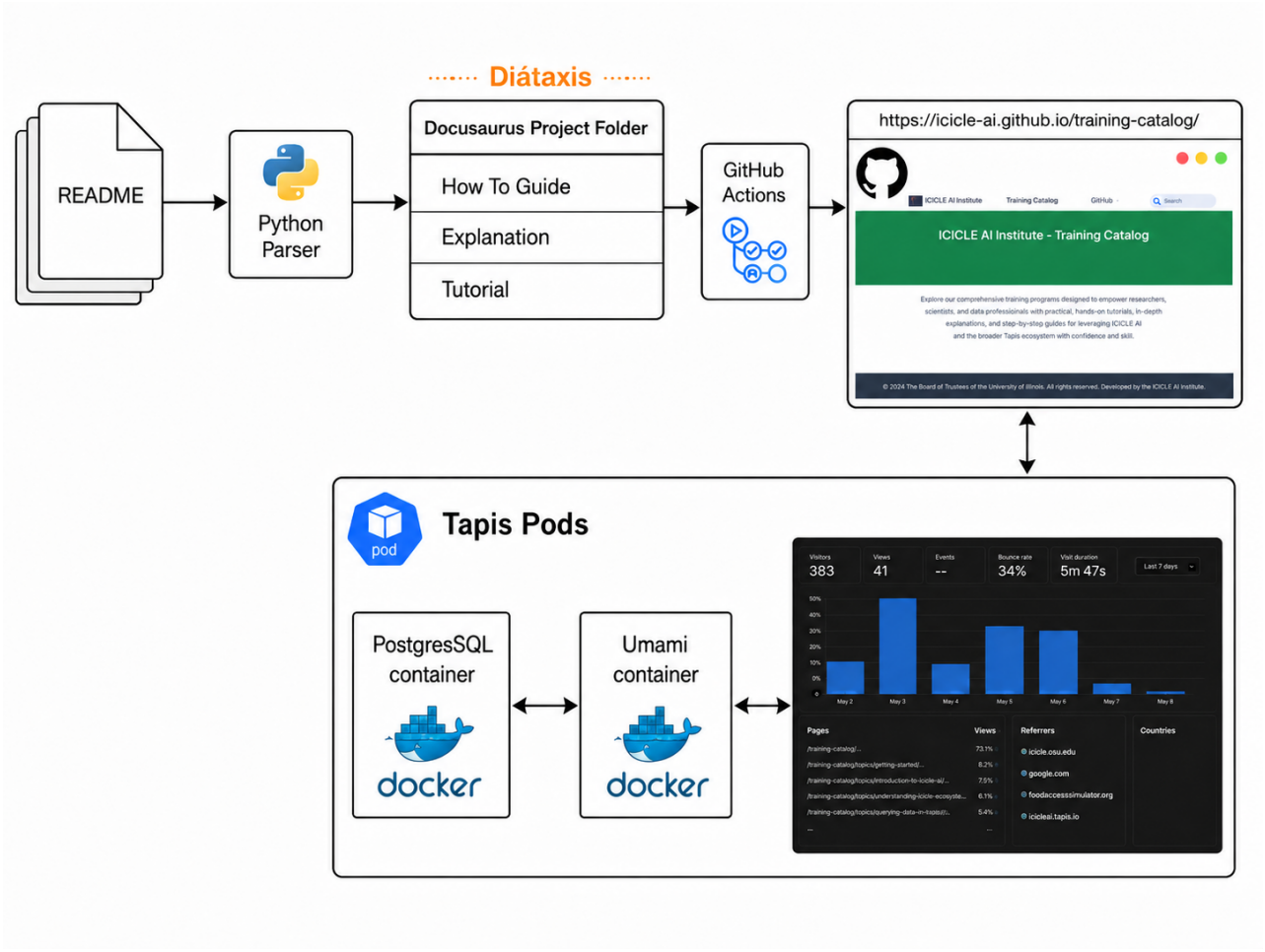


Figure 1: Architecture of the ICICLE Training Catalog.

3.4 Education Material and Workforce Development

The ICICLE AI Institute is committed to advancing AI and CI literacy through educational initiatives spanning K–12 education and professional development. The Training Catalog includes a dedicated education section that provides resources tailored to diverse learning levels and backgrounds.

These materials include concise "nutshell" videos that provide high-level overviews of activities across the ICICLE ecosystem, giving users a rapid orientation to key concepts and applications.

The platform additionally hosts educational documents that can be viewed inline through embedded interfaces or downloaded for offline and classroom use.

Educational tags such as *youth* and *video* support filtering and discovery, helping users locate materials matched to their level and interests.

3.5 Search and Thrust-Based Tagging

To enhance discoverability, the Training Catalog organizes content using thrust-based tags aligned with ICICLE's core research areas. Each resource is associated with one or more tags, allowing users to filter materials based on specific domains or topics.

The platform also integrates full-text, client-side search that enables keyword queries across all documentation pages. The search index is built during static-site generation and queried in the browser, eliminating the need for a hosted search service. The search interface highlights matches in titles and body content, enabling rapid identification of relevant pages.

3.6 Usage Analytics and Adoption

To characterize early adoption, we analyzed privacy-preserving Umami data collected from September 2025, when analytics collection was configured for the Training Catalog, through April 2026. The data comprise aggregate site metrics and CSV exports grouped by page path, referrer, device class, browser, and operating system.[1] The adoption analysis below emphasizes views, visits, visitors, and page-level navigation.

[1]In Umami, a *view* is a page load or client-side route change; a *visit* is a session containing one or more views; and a *visitor* is an approximate unique browser or device.

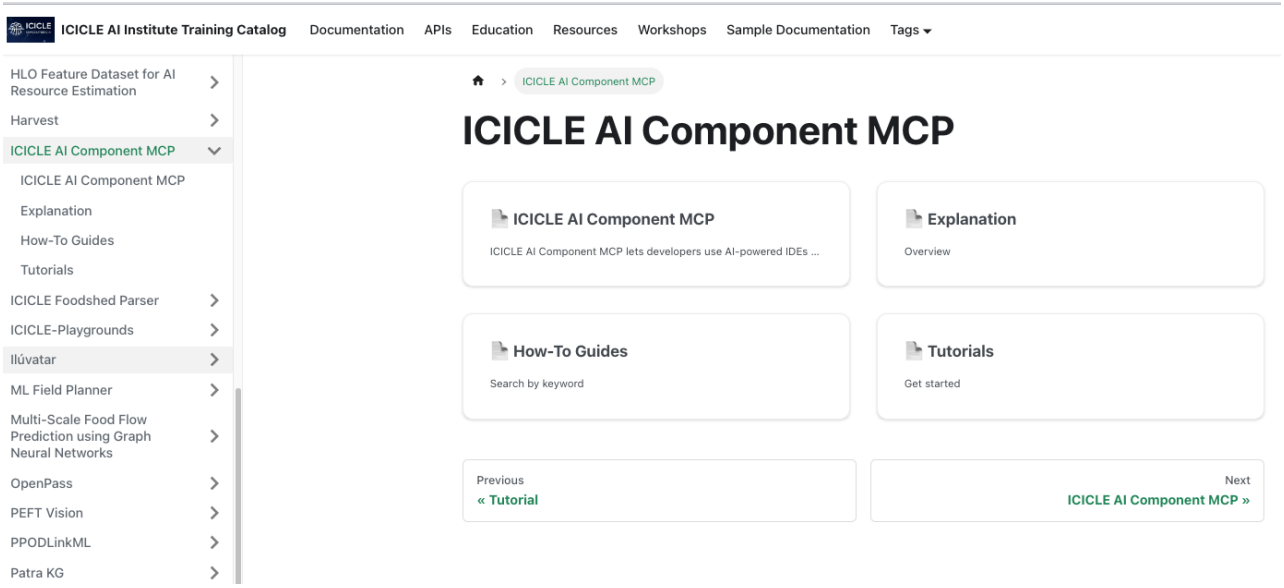


Figure 2: Component documentation page organized using the Diátaxis format.

Over the September 2025–April 2026 collection period, Umami recorded 2,690 views, 1,040 visits, and 244 visitors. Tables 1–5 summarize the resources accessed, the referral sources, and the client environments observed during this period. The per-attribute breakdowns do not sum to the overall view and visit totals: Umami records a row for an attribute (page path, referrer, device class, browser, or operating system) only when the value is identifiable from the request, and reports the top-ranked values for each category. The page-path export accounts for 1,862 of 2,690 views; the referrer export captures 169 visits with an identifiable referrer, of which the top entries shown in Table 2 account for 155; device class, browser, and operating system each cover 336 visits with a classified client environment.

The page-path export records, for each Training Catalog URL or route, the number of views it received; it does not count unique users. It contains 1,862 views distributed across 309 distinct paths. Grouping these paths by catalog area yields: documentation 1,273 views, education resources 158, the catalog home 139, API pages 137, workshops 63, and other or uncategorized paths 92. This distribution indicates that users routinely navigated beyond the landing page into the documentation, education, API, and workshop areas of the Training Catalog.

Table 1 reports the ten most-accessed catalog paths; the top five alone account for 18.0% of page-path views, with a long tail of project and education pages contributing smaller but non-trivial use. Referrers (Table 2) include ICICLE-owned entry points, general search engines, a domain-specific application site, Tapis-hosted infrastructure, and shared links from Microsoft Teams. Device classes (Table 3) indicate that the Training Catalog is used predominantly for large-screen technical work: laptop and desktop visits total 285 of 336 device-classified visits (84.8%), while mobile and tablet visits account for the remaining 51 (15.2%).

Table 1: Top ten catalog resources by view count.

Resource	Views
Catalog home	139
Documentation introduction	62
Other resources introduction	51
Education introduction	47
Workshops introduction	37
FEAST category	30
AI Ethics Curriculum	26
FEAST documentation	25
API introduction	23
ScienceAgent interface category	22

Table 2: Top referrers by visit count.

Referrer	Visits
icicle.osu.edu	52
google.com	28
foodaccesssimulator.org	19
icicleai.tapis.io	15
Microsoft Teams CDN	14
bing.com	11
linkedin.com	9
duckduckgo.com	7

Because the Training Catalog uses a self-hosted, privacy-preserving Umami deployment, these data do not include demographic attributes, institutional role, or individual identity. The deployment is also subject to common web-analytics limitations: browser tracking protections may undercount some visitors, automated traffic may not be completely filtered, and single-page-application routing can

Table 3: Device classes by visit count and share.

Device	Visits	Share
Laptop	202	60.1%
Desktop	83	24.7%
Mobile	46	13.7%
Tablet	5	1.5%

Table 4: Browsers by visit count.

Browser	Visits
Chrome	240
Safari	25
iOS browser	23
Firefox	19
Edge Chromium	16
Chrome iOS	5
iOS WebView	3
Opera	3

Table 5: Operating systems by visit count.

Operating system	Visits
Windows 10	164
Mac OS	92
iOS	32
Linux	21
Android OS	19
Windows 7	4
Chrome OS	3
Windows XP	1

affect how page transitions are recorded. We therefore treat the analytics as evidence of adoption and navigation behavior rather than as a causal evaluation of learning outcomes or demographic reach. Future evaluations will pair these passive analytics with voluntary surveys or workshop-specific feedback instruments that can measure how the learn–play–configure model affects user confidence, completion rates, and reuse of ICICLE software.

4 LEARN–PLAY–CONFIGURE PATHWAY

The ICICLE ecosystem serves users whose needs change as they move from initial exploration to reuse and extension. To support that progression, the Training Catalog organizes resources around a learn–play–configure pathway. Users *learn* by reading structured documentation, short-form videos, and tutorials; *play* by interacting with hosted services and examples; and *configure* by following repository links, cloning code, modifying configuration files, and adapting components for their own environments. Figure 3 illustrates this pathway with the FEAST Food Equity Access Simulation as a concrete example.

This pathway was used in practice by students in the San Diego Supercomputer Center Research Experience for High School Students Summer Program [8]. Students used the Training Catalog to identify ICICLE projects, move from project documentation to source repositories, download and install selected software, and run several projects on the SDSC Expanse system. Their use of the Training Catalog highlighted the importance of visible prerequisites, clear repository links, and compute-environment notes. These observations informed the Training Catalog's emphasis on metadata, how-to guides, and links between documentation pages and executable resources.

5 USER PERSONAS

The learn–play–configure pathway is intended to support several user groups without requiring separate sites for each audience.

5.1 Researchers

Researchers use the Training Catalog to discover AI tools, datasets, and workflows that can be applied to scientific problems. Thrust-based tags help them move from a domain such as food access or animal ecology to relevant components, while project pages connect explanations, tutorials, APIs, and source repositories. In the FEAST example, a researcher can move from the catalog entry to the hosted simulator and then to the source repository for adapting scenario parameters.

5.2 Students

Students benefit from structured learning resources that introduce AI and CI concepts through examples that can be run or inspected. Tutorials and workshop materials support first exposure, while links to hosted services and repositories let students move beyond static reading material. The SDSC REHS use case showed that explicit prerequisites and compute-environment notes are especially important when students transition from reading documentation to running code.

5.3 Educators

Educators use the Training Catalog to find teaching materials and incorporate AI-related modules into their courses. The structured documentation format helps instructors adapt resources for classroom environments and connect students to runnable examples beyond static reading material.

5.4 Research Software Engineers

Research software engineers (RSEs) are both users and contributors. As users, they rely on the Training Catalog to understand how components are documented, built, and deployed. As contributors, they use the README template, AI-assisted `icicle-readme-skill` workflow, parser feedback, and metadata conventions to prepare project documentation that can be ingested into the catalog. This role connects the user-facing documentation experience to the maintenance work required to keep component pages current.

6 LESSONS LEARNED

Several lessons emerged from developing the Training Catalog across institutions. First, a shared documentation framework is a coordination tool, not only a presentation choice. Diátaxis gave contributors a common vocabulary for deciding whether material should teach a first-time workflow, answer a specific task question, explain concepts, or provide reference details.

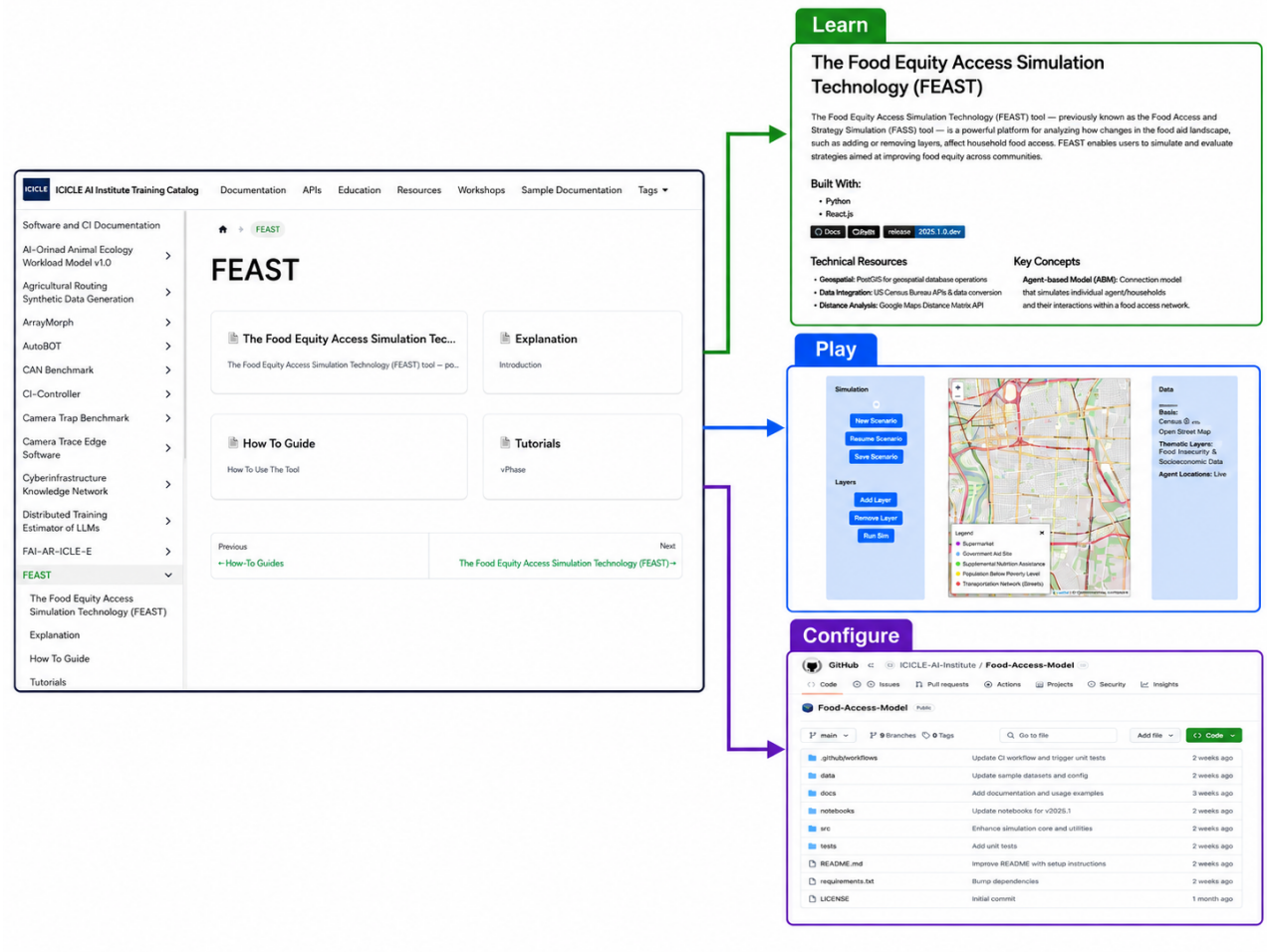


Figure 3: Learn–play–configure pathway, illustrated with FEAST.

Second, automation needs an explicit metadata contract. The Python parser reduced manual formatting work, but only after the team agreed on required fields, section names, link conventions, and update timestamps. Without that contract, automated ingestion can reproduce inconsistencies at larger scale. In early onboarding attempts, parser warnings often pointed to practical problems such as missing canonical tags, ambiguous section headings, and repository links that had drifted from the current project layout.

Third, repository-based documentation reduced cross-institutional friction. Docusaurus, GitHub Actions, and Markdown allowed contributors from different institutions to work in familiar tools while still producing one public catalog. A recurring friction point was that contributors had different local documentation habits: some maintained detailed README files, some stored installation notes in workshop slides, and some assumed users already knew the relevant compute environment. The repository-based workflow made these differences visible and easier to resolve through review.

Fourth, privacy-preserving analytics are useful but incomplete. Umami provided enough information to identify popular resources, referrers, and device patterns, but it did not answer demographic questions or measure learning outcomes. We therefore plan to pair analytics with voluntary feedback mechanisms when evaluating training impact.

Fifth, links between documentation and executable resources require active maintenance. Hosted services, repositories, and workshop materials can change independently across institutions. Dead links and outdated setup instructions were among the most visible failure modes for users, so the Training Catalog needs regular link checks and ownership metadata in addition to initial onboarding documentation.

7 CONCLUSION

The ICICLE Training Catalog serves as a central hub for the broader AI and cyberinfrastructure community, providing a single, structured entry point into the full range of resources produced across the ICICLE ecosystem. By aggregating research software documentation, API specifications, educational materials, workshop content, and multimedia resources into one discoverable platform, the Training Catalog lowers the barrier for external audiences—students,

researchers, educators, and practitioners—to explore and engage with ICICLE's work.

The platform's design combines the Diátaxis documentation framework with thrust-based tagging, full-text search, and the learn–play–configure interaction model to support users with varying levels of expertise and intent. Whether a user is encountering ICICLE tools for the first time or integrating them into active research workflows, the Training Catalog provides the structure and navigation needed to find relevant content quickly.

Early Umami analytics show measurable adoption: from September 2025 through April 2026, the Training Catalog recorded 2,690 views, 1,040 visits, and 244 visitors, while the page-path CSV export showed engagement across documentation, education, API, and workshop pages. These metrics are an initial baseline rather than a complete evaluation, but they provide a foundation for improving content organization and identifying resources that need clearer onboarding paths.

Looking ahead, the Training Catalog will continue to evolve alongside the ICICLE community. Planned improvements include richer filtering by compute requirements and experience level, publication of the documentation parser and validation workflow, more explicit compute-environment guidance for hands-on examples, and voluntary feedback instruments that can complement privacy-preserving analytics with learning-outcome data.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Award No. OAC-2112606 through the ICICLE AI Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would like to thank the ICICLE community and partner institutions for their contributions to the development of open AI cyberinfrastructure and training resources.

The authors used Claude (Anthropic) to assist with spell-checking, grammar, and minor writing corrections. All technical content, analysis, and conclusions are solely the work of the authors.

REFERENCES

[1] ICICLE AI Institute. 2021. ICICLE: Institute for Intelligent Cyberinfrastructure with Computational Learning in the Environment. https://icicle.ai. Accessed: 2026-03-25.

[2] ICICLE AI Institute. 2024. ICICLE Training Catalog. https://icicle-ai.github.io/training-catalog/. Accessed: 2026-03-25.

[3] ICICLE AI Institute. 2025. ICICLE README Skill. https://github.com/ICICLE-ai/training-catalog/blob/docusaurus-demo/icicle-readme-skill/SKILL.md. Accessed: 2026-05-18.

[4] ICICLE AI Institute. 2025. ICICLE README Template. https://github.com/ICICLE-ai/training-catalog/blob/docusaurus-demo/icicle-readme-skill/templates/README.md. Accessed: 2026-05-18.

[5] Meta Open Source. 2024. Docusaurus. https://docusaurus.io/docs. Accessed: 2026-03-25.

[6] Dhabaleswar K. Panda, Vipin Chaudhary, Eric Fosler-Lussier, Raghu Machiraju, Amit Majumdar, Beth Plale, Rajiv Ramnath, Ponnuswamy Sadayappan, Neelima Savardekar, and Karen Tomko. 2024. Creating intelligent cyberinfrastructure for democratizing AI. *AI Magazine* 45, 1 (2024), 22–28. https://doi.org/10.1002/aaai.12166

[7] Daniele Procida. 2021. Diátaxis: A systematic approach to technical documentation authorship. https://diataxis.fr/. Accessed: 2026-03-25.

[8] San Diego Supercomputer Center. n.d.. UCSD Research Experience for High School Students (REHS). https://education.sdsc.edu/studenttech/rehs/. Accessed: 2026-03-27.

[9] Joe Stubbs, Richard Cardone, Mike Packard, Anagha Jamthe, Smruti Padhy, Steve Terry, Julia Looney, Joseph Meiring, Steve Black, Maytal Dahan, Sean Cleveland, and Gwen Jacobs. 2021. Tapis: An API Platform for Reproducible, Distributed Computational Research. In *Advances in Information and Communication*, Kohei Arai (Ed.). Springer International Publishing, Cham, 878–900.

[10] Umami Software, Inc. 2022. Umami Analytics. https://umami.is/. Accessed: 2026-03-25.

Understanding Computational Science and Engineering (CSE) and Domain Science Skills Development in National Laboratory Postgraduate Internships

Morgan M. Fong
University of Texas at Austin
mmfong@utexas.edu

Hilary Egan
National Laboratory of the Rockies
Hilary.Egan@nrel.gov

Marc Day
National Laboratory of the Rockies
Marc.Day@nrel.gov

Kristin Potter
National Laboratory of the Rockies
Kristi.Potter@nrel.gov

Michael J. Martin
National Laboratory of the Rockies
Michael.Martin@nrel.gov

ABSTRACT

Background and Context: Harnessing advanced computing for scientific discovery and technological innovation demands scientists and engineers well-versed in both domain science and computational science and engineering (CSE). However, many universities lack the resources needed to provide integrated training. National laboratories offer internship opportunities capable of developing these skills.

Objective: We present an evaluation of federally-funded postgraduate internship outcomes in CSE at a national laboratory. We seek to answer three questions: 1) What computational, research, and professional skills do students improve through internships at the selected national laboratory? 2) Do students gain knowledge in domain science topics through their internships? 3) Do students' career interests change after these internships?

Method: We developed a survey and collected responses from past participants of five federally-funded internship programs and compare participant ratings of their prior experience to their internship experience.

Findings: Our results indicate that participants improve CSE skills and domain science knowledge, and become more interested in careers at national labs. Participants found skills developed during their internships to be relevant and useful in their post-internship positions.

Implications: The perceived impact of postgraduate internships often centers on *career opportunities and applications of course material*. We show national laboratory internships are an opportunity for students to build critical technical skills, in both CSE and domain science, that may not be available at all institutions. This suggests internships and other experiential learning in settings such as national laboratories have the potential to make substantial contributions to building technical competency in graduate students.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/8

KEYWORDS

Internship, Graduate Students, Computational Science and Engineering, National Lab

1 INTRODUCTION

1.1 The need for integrative training in computational science and engineering and domain science

Scientific discovery and technological advancement increasingly depend upon the use of advanced computing, including modeling and simulation (mod/sim), data science, and machine learning and artificial intelligence (ML/AI) [37, 38, 40]. The development and application of these tools has emerged as its own scientific discipline, *computational science and engineering, or CSE* [38], distinct from computer science, or CS. In many cases, the most effective use of advanced computing for scientific discovery and technology development occurs when students are cross-trained in both CSE and *domain science* areas such as engineering, physics, chemistry, or biology [28]. The value of these skills increases when scientists and engineers are able to make effective use of high performance computing (HPC) systems which are characterized by increasing complexity, including massive parallelization, heterogeneous architectures, and the use of complex programming models [13].

The need for CSE and HPC expertise is not confined to academic and research institutions. Industry's growing reliance on CSE and HPC is evident in the major investments in world-class supercomputers as listed in the TOP500 [43] and accelerating adoption of cloud-based HPC [31]. The ability to effectively use commercial computational software and scientific programming skills in languages such as C++, Python, and MATLAB has been identified as highly sought after skills for engineers at the bachelors, masters, and doctoral levels through analysis of job advertisements [16] and surveys of Ph.D. earners employed in industry [46]. Furthermore, both programming [16] and HPC [41] skills are associated with significant salary premium for engineers and scientists pursuing careers in industry.

1.2 Obstacles to integrated CSE/domain science training

Previous research on interdisciplinary CS and CSE training at the graduate level has focused heavily on students shifting into CS and CSE after completing undergraduate degrees in domain sciences (e.g., physics, chemistry, and biology). Depending on the major, courses, and experiential education options pursued by individual students, non-CS students can complete their degree with widely varying range of programming skills and experience.[1]. This variation can increase among students who gain professional experience prior to enrolling in graduate programs [25]. These challenges have led to the creation of ad hoc CS "boot-camps" within graduate programs to provide all students with the skills needed to succeed in future graduate courses [1, 26].

However, these models do not address the increasing demand for domain scientists trained in CSE skills [28, 42]. There are limits to the ability of academic institutions to provide both the expertise and hardware needed for effective training [22]. Training may be limited by lack of access to hardware, with many universities only able to support small-scale systems (e.g., [34, 45]). As a result, relatively few university students have access to modern supercomputers, which often have vastly different capabilities and architectures than those of smaller, more affordable clusters. Based on the November 2024 ranking of the top 500 most powerful supercomputers in the world [43], 173 supercomputers are located in the United States (US), the context in which this study is situated. Of those in the US, 67 are hosted in government research facilities, including 38 at Department of Energy (DOE) National Laboratories, 3 are hosted in research foundations, and 80 are hosted in industry and vendors. However, only 23 top-500 supercomputers are hosted by US universities. These computers are concentrated in 17 very high research activity universities [18] located in 9 states, further limiting access to hands-on training. These challenges increase at non-R1 universities, where access to smaller-scale clusters is often limited. [14].

Even in universities with access to larger-scale compute facilities, integrating domain sciences with CSE training faces institutional challenges. These include the constraints of existing curriculum requirements (e.g., [22]), lack of evaluation of learning outcomes (e.g., [47]), or inadvertently de-emphasizing computational skills when integrating domain science topics (e.g., [3, 11]). These challenges are often exacerbated by a shortage of faculty in key areas of CSE [48].

1.3 Models for integration of national laboratory resources with academic training and prior assessment

Making national laboratory computing resources and staff expertise available to university students is a possible solution to these challenges with multiple strategies. Because DOE computing resources are used by academic research groups across the United States, many research groups have developed their own internal documentation and training for the systems that they have access to, and the tools that are necessary of support their work.

A second related approach is to provide national laboratory computing time and staff support directly to university partnerships seeking to provide training at the institutional level [2, 36, 39]. These programs are often designed to increase skills needed for using national laboratory computing systems as part of dissertation research.

A third approach is to integrate students into national laboratory research that uses HPC through internships in a situated learning model [5]. In this approach, students gain direct access to communities of practice by working in a team of researchers using on-site HPC facilities.

Previous assessments of national laboratory internships not specifically focused on CSE shows multiple impacts, including:

- Increased retention in science, technology, engineering, and mathematics (STEM) careers [7, 17]
- Increased research skills [8, 17]
- Increased professional networks [8]

Internships in the broader field of computer science are linked multiple benefits, including:

- Development of technical skills [20, 21, 29, 40]
- Opportunities for career exploration [20, 21, 23]
- Expanded professional networks that contribute to future professional success [15, 23, 29]

However, the question of the development of specific technical skills in national laboratory internships, specifically in CSE and HPC, remains open. While there is substantial literature on training in HPC, these papers tend to focus on descriptions of internships at academic computing centers [19] or workshops designed for specific supercomputer sites (e.g., [4, 6, 12, 47]). The potential benefits of internships in the national laboratory environments, which combine expertise in CSE, domain science, and HPC operations has not been examined.

1.4 Research Questions

Our study aims to address this gap through an evaluation of federally funded, postgraduate (including postbaccalaureate-, master's-, and doctoral-level) internships hosted at the National Laboratory of the Rockies, or NLR. (At the time of the internships, the facility was known as the National Renewable Energy Laboratory, or NREL). To this end, we ask the following research questions:

Research question 1: What computational skills, research skills, and professional skills do postgraduate students improve through their internships at NLR?

Research question 2: What domain-science topics do postgraduate students gain more familiarity with through their internships at NLR?

Research question 3: Do postgraduate students' career interests change after their internships at NLR?

To answer these questions, we conducted a survey study with past participants of five federally funded internship programs hosted at NLR. These programs serve post-baccalaureate and graduate participants from a variety of disciplines and educational institutions. While this survey is specifically focused on the intersection of CSE and NLR's core research, results are likely to be broadly applicable to interdisciplinary training that combines CSE with other domain sciences.

We provide additional context of the internship programs and internship site in sections 2.1 and 2.2. Section 2.3 details the methods

used to design the survey as well as collect and analyze survey data. Section 5 includes discussion of our results, conclusions, and possibilities for future work.

2 METHODS

2.1 Host Institution

This study examines the impact of *federally-funded internships* in CSE at the National Laboratory of the Rockies (NLR), a Department of Energy (DOE) National Laboratory located in Golden, Colorado. The laboratory has additional campuses in Arvada, Colorado, Fairbanks, Alaska, and Washington, DC. As of 2024, the laboratory had 3,675 staff, including regular employees, postdoctoral researchers, visiting professionals, and subcontractors.

NLR's research focuses on energy systems and energy efficiency in support of the research objectives of multiple DOE offices, including the Office of Critical Minerals and Energy Innovation (CMEI), the Office of Electricity (OE), the Hydrocarbons and Goethermal Energy Office (HGEO), and the Office of Science (SC). NLR increasingly uses computational science tools, including mod/sim, data science, and ML/AI to perform this research. NLR's efforts in these areas are led by the Computational Science Center (CSC). CSC hosts Kestrel, NLR's third generation of High Performance Computing (HPC) that is available to all researchers funded by CMEI. The center also engages in research in computational science spanning fundamental work on algorithms, software development, scientific visualization, and application to a range of topics in energy systems [27]. As of November 2024, CSC staff includes 7 managers, 11 administrative staff, 65 scientific staff, 44 technical staff (responsible for operation of the HPC facilities), and 14 postdocs, or a total of 141 staff. Staff expertise covers a broad range of CSE topics, including modeling and simulation, data science and visualization, and machine learning and artificial intelligence. Staff expertise also includes software engineering, systems administration, and data center infrastructure management and facility integration.

NLR hosts interns in CSE through two mechanisms: federal educational fellowships, with financial support from the fellowship sponsor, and as temporary staff working on specific projects, with financial support from project funds. Federal educational fellowships specify that the student be achieving specific educational objectives while contributing to active research projects. Both types of internships are available to undergraduate, postbaccalaureate, and graduate students. Because CSE is often an advanced undergraduate or graduate-level topic, NLR's internships in this area are heavily weighted towards postbaccalaureate and graduate students. Based on this, the current study focuses on postgraduate students supported through federal educational fellowships.

2.2 Internship Programs Included

NLR currently brings postbaccalaureate and graduate student interns in CSE to the laboratory through five federally-funded educational internship programs. Four of these programs are funded by the DOE: the Computational Science Graduate Fellowship (**CSGF**), the Science Undergraduate Laboratory Internship (**SULI**) program, the CMEI High Performance Computing for Energy Innovation Internship program (**HPC4EI**), and the National Nuclear Security Administration's IMPACT Internship (**NNSA-IMPACT**). A fifth program is funded by the National Science Foundation (NSF): the Mathematical Science Graduate Internship (**NSF-MSGI**).

For all of these programs, each intern is assigned a mentor (i.e., a staff member at the national lab), and each mentor is expected to submit a plan to the sponsor that identifies learning objectives as well as scientific objectives for the internship. Once the internship begins, interns have the opportunity to learn from direct interaction with mentors and other members of the technical staff. In addition to this mentorship, NLR's HPC operations staff offers a broad range of structured trainings and tutorials on topics in HPC, as well as providing one-on-one advising when needed. Each of these programs has unique goals, eligibility criteria, and post-internship report requirements, which we describe next.

DOE CSGF is a highly competitive four-year fellowship for doctoral students in science and engineering who incorporate computational science into their work [24]. Applicants must be US citizens or permanent residents. As part of their fellowship, students must spend at least one semester at one of 21 DOE facilities pursuing a project outside of their dissertation work. The 20-30 students seeking placement each year work directly with practicum coordinators at different facilities to identify a facility, mentor, and project that meet their interests. Students must complete an initial practicum within the first two years of their fellowship and are free to participate in second or third practicums, either with the same DOE facility and mentor, or a different DOE facility and/or mentor. CSC manages placement within NLR, though assigned mentors and projects can within CSC or in another center with computationally intensive projects. While placements are generally made during the summer, they are also occasionally made in the fall or spring semesters.

DOE SULI is a one-semester internship program open to US citizens or permanent residents who are undergraduates, postbaccalaureate, or graduate students in their first two years since completing their bachelor's degree [10]. Students are not required to be majoring in a STEM field but must have at least 6 credit hours in STEM and a minimum 3.0 out of 4.0 grade point average. SULI students are eligible for placement at 17 different DOE facilities. Students may complete up to two placements, either with the same DOE facility and mentor, or a different DOE facility and/or mentor. Students may be placed for the summer, fall, or spring semesters.

The **CMIE HPC4EI** internship program is linked to the DOE's HPC4EI program [9]. The HPC4EI program matches DOE national laboratory computing capabilities with industrial partners to solve key challenges in industrial energy use and the manufacture of energy materials. As part of its industrial competitiveness mission, HPC4EI offers internships to US citizens in HPC-relevant fields as undergraduate, postbaccalaureate, and graduate students [35]. HPC4EI interns work directly on projects sponsored by the HPC4EI program.

NNSA-IMPACT targets building research capacity at Minority Serving Intuitions (MSIs) as defined by the US Department of Education [44]. NNSA-IMPACT is open to US citizens of all backgrounds, academic levels, and majors at these institutions [32]. Students can complete an in-person or virtual internship and have the option of extending into the academic year.

NSF-MSGI is open to doctoral students in mathematics, statistics, and applied mathematics. The goal of NSF-MSGI is to expose these

students to non-academic careers [33]. Unlike the other programs described, NSF-MSGI is open to international students. Students may complete multiple internships, but each internship must be with a different mentor.

All of these programs operated as fully virtual programs in 2020 and 2021 due to the COVID pandemic. DOE CSGF deferred all summer 2020 placements to summer 2021 placements, but did allow virtual placements in fall of 2020 and spring of 2021. Neither CMIE HPC4EI nor NSF-MSGI operated in summer of 2024 due to budget uncertainties.

2.3 Survey Design

The research team collaboratively designed the survey. Given the purpose of the survey to understand interns' 1) technical and professional skill development and 2) pathways to future careers across a variety of internship programs and projects, survey questions were organized and designed around the following themes: Computational Skills, Familiarity with NLR-relevant domain science topics such as energy generation and efficiency, Research Skills, Professional Skills, and Career Interests. Full questions for each theme are provided in Table 1.

For each theme, the research team identified relevant skills and topics based on their extensive experience with the internship programs described in Section 2.2. More specifically, four of the co-authors (Martin, Egan, Day, and Potter) have extensive involvement in NLR's internship programs. Michael Martin has served as NLR's coordinator for the DOE CSGF, CMIE HPC4EI, and NNSA-MSIIP programs as well as mentored students through the NSF-MSGI program. Hilary Egan has served as NLR's co-coordinator for the DOE CSGF program and has mentored students through the DOE SULI program. Marc Day has mentored students through the DOE CSGF and NSF-MSGI programs. Kristi Potter has mentored students through the DOE CSGF and SULI programs.

For the questions related to computational skills, inspiration was taken from a previous report on CSE education [38]. The research team iteratively adjusted the questions for clarity and understanding across each internship program's context.

Additionally, optional open-ended response options were provided for participants to describe other computational skills, research skills, professional skills, and topics that they improved or gained familiarity with during their internship to capture areas we did not explicitly ask about.

2.4 Participant Background Information

Participants were also asked to self-report the following academic background information:

- Prior experience at other Department of Energy National Labs (i.e., no experience, once, 2-3 times, or 4 or more times)
- Prior experience at other federal laboratories (i.e., no experience, once, 2-3 times, or 4 or more times)
- Year of first internship at NREL, as some participants may have extended their internship or returned to NREL at a later time
- Academic stage and major / field of study during internship at NREL or, for participants who were not enrolled in a degree program during their internship, major / field of study during degree prior to internship
- Location of internship (i.e., in person, hybrid, remote)
- Focus of internship on domain science topics (i.e., not at all, somewhat focused, very focused, or extremely focused)
- Current position at time of completing survey (i.e., in the same or different degree program, looking for employment, or currently employed)

Additional questions were asked based on participants' current positions to understand how their internship experiences may or may not have carried beyond their time at NLR. These questions are also provided in Table 1.

2.5 Participants

We recruited participants from five federally-funded internships at NLR as described in Section 2.2. Because these internship programs are open to a broader population than our intended focus, we applied the following inclusion criteria:

- Participants must have completed a bachelor's degree or equivalent prior to starting their internship at NLR
- Participants must have started their internship at NLR in 2018 or later
- Participants incorporated CSE into their internships, as evidenced by (1) the nature of the internship program (CSGF, HPC4EI, and NSF-MSGI), or, (2) for SULI and NNSA-MSIIP students, either (a) a mentor in CSC, or (b) use of NLR's supercomputer during their internship in another NLR center.

Many potential participants were identified given the research team's extensive involvement with the internship programs of interest. Public records of some internship programs were cross-referenced with internal NLR records to identify additional participants. Additionally, we contacted NSF-MSGI administrators for the list of past NSF-MSGI participants hosted at NLR. All study procedures follow our IRB-approved protocol (#DOE001036).

2.6 Data Collection

Participants were sent either a recruitment email or LinkedIn message in September 2024. The survey accepted responses for three weeks.

Because the authors have extensive experience working with these internship programs (see Section 2.3 for more details), Morgan Fong was recruited as a graduate student intern to the research team to manage distribution of the survey and removal of personally identifiable information from responses. Martin, Egan, Day, and Potter only had access to de-identified and aggregated data.

2.7 Data Analysis

All Likert-scale questions, except for questions related to Professional Skills, are coded on a scale of zero (i.e., "No experience," "No improvement," "No familiarity," "No increase in familiarity," "Not interested," "Not at all," "Not useful," "Not related,") to three (i.e., "Expert," "Substantial Improvement," "4 or more times," "Extremely familiar," "Substantial increase in familiarity," "Extremely interested,"

Table 1: Survey questions organized by theme. All questions were asked with respect to before the internship and during the internship. Questions on career interest were split by domain and sector. All questions use the program terminology in effect at the time of the survey.

Theme	Question Context, Response Options	Skills and Topics
Computational Skills	Prior to participating in [internship program], how much experience did you have with the following? *No experience, Some experience (e.g., 1-2 projects), Extensive experience (e.g., took advanced coursework), Expert (e.g., focus of degree, years of experience)*	1. Writing general-purpose programs 2. Writing programs for scientific problems 3. Writing distributed or parallel programs 4. Writing software collaboratively 5. Performing data visualization and analysis 6. Using AI/ML to develop models 7. Developing simulations of systems or phenomena (e.g., computational fluid dynamics, computational chemistry) 8. Using high performance computers or supercomputers 9. Using software libraries or open-source code
	During your internship at NREL, to what extent did you improve your skills in the following? *No improvement, Some improvement, Moderate improvement, Substantial improvement*	
Familiarity with Energy and Efficiency Energy Topics	Prior to participating in [internship program], how familiar were you with the following? *No familiarity, Somewhat familiar (e.g., news, social media), Very familiar (e.g., took advanced coursework) Extremely familiar (e.g., focus of degree)*	1. Renewable energy 2. Energy efficiency 3. Materials usage 4. Circular economies 5. Environmental justice 6. Energy equity 7. Sustainable Computing
	To what extent did your internship at NREL increase your familiarity with the following topics? *No increase in familiarity, Some increase in familiarity, Moderate increase in familiarity, Substantial increase in familiarity*	
Research Skills	Prior to participating in [internship program] how much experience did you have with the following? *No experience, Once, 2-3 times, 4 or more times*	1. Interview for a research position 2. Contribute to a research project 3. Work in a team with varying types of expertise 4. Author or co-author a research publication (e.g., journal article, conference paper, book chapter)
	During your internship at NREL, how much experience did you have with the following? *No experience, Once, 2-3 times, 4 or more times*	
Professional Skills	Prior to participating in [internship program], to what extent would you agree or disagree with the following statements? *Disagree, Somewhat disagree, Somewhat agree, 4: Agree*	1. I'm part of a research community where I can get feedback on my work 2. I can create an effective oral research presentation 3. I can design an informative research poster 4. I can discuss research findings with a technical audience 5. I can discuss research findings with a non-technical audience
	During your internship at NREL, to what extent do you agree or disagree with the following statements? *Disagree, Somewhat disagree, Somewhat agree, 4: Agree*	
Career Interests	Prior to participating in [internship program], how interested were you in a career in the following? *Not interested, Somewhat interested Very interested, Extremely interested*	1. Sustainability 2. Renewable energy 3. Another domain other than sustainability or renewable energy
	As a result of your internship at NREL, how interested were you in a career in the following? *Not interested, Somewhat interested Very interested, Extremely interested*	1. Industry 2. Academia 3. National Renewable Energy 4. Other Department of Energy National Laboratory 5. Other federal research facility 6. Non-Profit 7. Another sector other than the above
If currently in Different Degree Program, Looking for Employment, or Currently Employed	To what extent is your [field of study/positions applying for/ current position] related to sustainability or renewable energy? *Not related, Somewhat related, Very related, Extremely related* How useful are the skills you improved at NREL in applying for [degree programs/positions]? *Not useful, Somewhat useful, Very useful, Extremely useful*	1. Computational Skills 2. Familiarity with Sustainability and Renewable Energy Topics 3. Research and Professional Skills
If currently in the Same Degree Program, Different Degree Program, or Currently Employed	How often do you use the skills improved during your internship at NREL in your [current position]? *Never, Some of the time, Most of the time, All of the time*	1. Computational Skills 2. Familiarity with Sustainability and Renewable Energy Topics 3. Research and Professional Skills

"Extremely focused," "Extremely related," "Extremely useful"). Professional Skills were coded on a scale of 1 (i.e., "Disagree") to 4 (i.e., "Agree").

For professional skills and career interests, one-sided, paired samples t-tests were used to determine if participant ratings after internships are greater than their rating from before internships. For computational skills, research skills, familiarity with domain science topics, one-sided, one sample t-tests were used to determine if participants reported at least "Some improvement" for computational skills, one experience for research skills during the internship, "Some increase in familiarity" for familiarity with domain science topics. In other words, the t-tests for computational skills, research skills, and familiarity with domain science topics determined if participants' post-internship ratings were > 1.

Similarly, one-sided one sample t-tests were used to determine if participants current positions are at least "Somewhat related" to NLR domain science topics, if their skills are at least "Somewhat useful" in applications, and if their skills are used at least "Some of the time." $\alpha < 0.05$ served as the cutoff for statistical significance.

2.8 Limitations

The inclusion of a variety of internship programs provided a broader participant pool, and we are encouraged to see relatively high response rates across programs that indicate we may indeed have a representative sample of the intended population. However, the sample remains subject to self-selection bias based on the selectiveness of the internship programs and who chose to respond to the survey. Additionally, our statistical analysis remains limited due to small sample size. As a result, we rely on simple t-tests to perform simple comparisons and avoid non-normality tests with a small sample size. Similarly, we do not adjust the cutoff for statistical significance to avoid over correcting for a small sample size, though this may increase the rate of false positive findings. Throughout the results, we include figures that document the percentages of responses as well as the number of respondents to avoid overstating our claims.

For participants who completed their internships less recently, their responses may be less accurate. We attempted to combat this by ensuring our response options were specific as possible to help participants select the most accurate response.

3 RESULTS

The total number of interns by program as well as corresponding response rates are included in Table 2. Participants' background data are included in Table 3. For anonymity purposes, the following categories are grouped together: participants who reported being in the first or second year of their PhD during their internship, participants who reported being in the third year of their PhD or later during their internship. Additionally, participants' fields of study were grouped as follows: science (e.g., physical, life, and biological sciences), engineering (e.g., chemical and mechanical engineering), mathematics (e.g., math, applied math, statistics), computing (e.g., computer science, computer engineering), and interdisciplinary (e.g., fields of study that included more than one of the above).

Table 2: Total number of interns meeting inclusion criteria by program and year of internship and response rates. "NO" indicates the program was not operational for that year. "NP" indicates the program was operational, but NLR had no participation for that year.

Year	DOE CSGF	DOE SULI	NSF MSGI	CMEI HPC4EI	NNSA IMPACT	Total
2018	2	3	1	NO	NO	6
2019	3	1	3	NO	NO	7
2020	1	5	0	0	NP	6
2021	3	4	2	1	NP	10
2022	7	2	1	2	NP	12
2023	2	2	6	0	NP	10
2024	5	0	NO	NO	2	7
Total	23	17	13	3	2	58
Response Rate	52%	29%	38%	67%	0%	41%

Table 3: Participants' academic backgrounds and internship location.

Academic Stage During Internship	Post-Baccalaureate	3
	Master's	3
	Early Stage (1st or 2nd year) PhD	13
	Late Stage (3rd year or later) PhD	5
Major or Field of Study	Science	4
	Engineering	5
	Mathematics	5
	Computing	3
	Interdisciplinary	7
Internship Location	In-person	15
	Hybrid	4
	Remote/Online	5

Three participants reported at least one prior experience at a DOE National Laboratory, and no participants reported prior experience at non-DOE federal research facilities.

Participants generally reported their internships were focused on domain science topics (only 3 participants responded "Not at all"). At the time of their internships, participants were studying a variety of domain sciences and mathematics.

Averages and standard deviations for the main survey questions are included in Table 4. Next, we describe results by theme.

3.1 Computational Skills

On average, participants enter their internships with at least some experience across most computational skills. Additionally, participants report at least "some improvement" in general-purpose programming ($t(23) = 4.65, p = .000$), scientific programming ($t(23) = 4.66, p < .000$), data visualization & analysis ($t(23) = 5.17, p = .000$), use of HPC and supercomputers ($t(23) = 2.53, p = .009$), and using software libraries or open-source code ($t(23) = 5.35, p = .000$). The distribution of responses is provided in Figure 1.

Table 4: Descriptive statistics and t-test results. * indicates questions related to usefulness of skills in applying to positions, and + indicates questions related to use of skills in current position.

Question	Pre Internship Mean (SD)	Post Internship Mean (SD)	t-test
Computational 1	1.54(0.78)	1.67(0.70)	$4.65, p = 0.000$
Computational 2	1.67(0.92)	1.79(0.83)	$4.66, p = 0.000$
Computational 3	0.71(0.69)	0.79(0.98)	$-1.04, p = 0.846$
Computational 4	0.71(0.81)	1.08(1.06)	$0.39, p = 0.352$
Computational 5	1.50(0.72)	1.96(0.91)	$5.17, p = 0.000$
Computational 6	0.62(0.77)	1.00(1.22)	$0.00, p = 0.500$
Computational 7	1.29(1.00)	1.29(1.20)	$1.19, p = 0.122$
Computational 8	1.21(0.93)	1.62(1.21)	$2.53, p = 0.009$
Computational 9	1.62(0.97)	2.04(0.95)	$5.35, p = 0.000$
Familiarity 1	1.25(0.74)	1.83(0.76)	$5.36, p = 0.000$
Familiarity 2	1.21(0.66)	1.38(0.97)	$1.89, p = 0.035$
Familiarity 3	0.75(0.68)	0.88(0.99)	$-0.62, p = 0.728$
Familiarity 4	0.38(0.49)	0.50(0.83)	$-2.94, p = 0.996$
Familiarity 5	0.79(0.66)	0.62(0.82)	$-2.23, p = 0.982$
Familiarity 6	0.67(0.64)	0.54(0.83)	$-2.70, p = 0.994$
Familiarity 7	0.67(0.76)	0.88(0.90)	$-0.68, p = 0.749$
Research 1	0.79(0.78)	0.38(0.65)	$-4.73, p = 1.000$
Research 2	1.96(0.75)	1.50(0.83)	$2.94, p = 0.004$
Research 3	1.75(1.11)	1.46(0.93)	$2.41, p = 0.012$
Research 4	1.33(1.01)	0.88(0.85)	$-0.72, p = 0.761$
Professional 1	3.33(0.70)	3.67(0.56)	$-1.88, p = 0.036$
Professional 2	3.58(0.50)	3.46(0.98)	$0.68, p = 0.749$
Professional 3	3.42(0.58)	3.04(1.27)	$1.33, p = 0.902$
Professional 4	3.38(0.65)	3.54(0.83)	$-0.75, p = 0.231$
Professional 5	3.17(0.70)	3.00(0.98)	$0.64, p = 0.736$
Domain 1	1.54(1.02)	1.92(0.88)	$-2.39, p = 0.013$
Domain 2	1.62(1.01)	2.25(0.79)	$-3.98, p = 0.000$
Domain 3	1.38(0.77)	1.71(0.62)	$-3.39, p = 0.001$
Sector 1	1.29(0.55)	1.17(0.70)	$0.83, p = 0.792$
Sector 2	1.12(1.03)	1.08(0.97)	$0.24, p = 0.593$
Sector 3	1.54(0.78)	2.42(0.78)	$-4.14, p = 0.000$
Sector 4	1.54(0.83)	1.83(0.87)	$-2.60, p = 0.008$
Sector 5	1.25(0.79)	1.29(0.75)	$-0.27, p = 0.394$
Sector 6	0.67(0.70)	0.67(0.87)	$0.00, p = 0.500$
Sector 7	0.38(0.58)	0.46(0.59)	$-0.81, p = 0.213$
Computational*	2.40(0.74)		$7.36, p = 0.000$
Familiarity*	1.93(1.03)		$3.50, p = 0.002$
Research & Professional*	2.40(0.74)		$7.36, p = 0.000$
Computational+	2.26(0.75)		$8.04, p = 0.000$
Familiarity+	1.48(1.04)		$2.21, p = 0.019$
Research & Professional+	2.26(0.92)		$6.61, p = 0.000$
Related	1.80(1.32)		$2.35, p = 0.017$

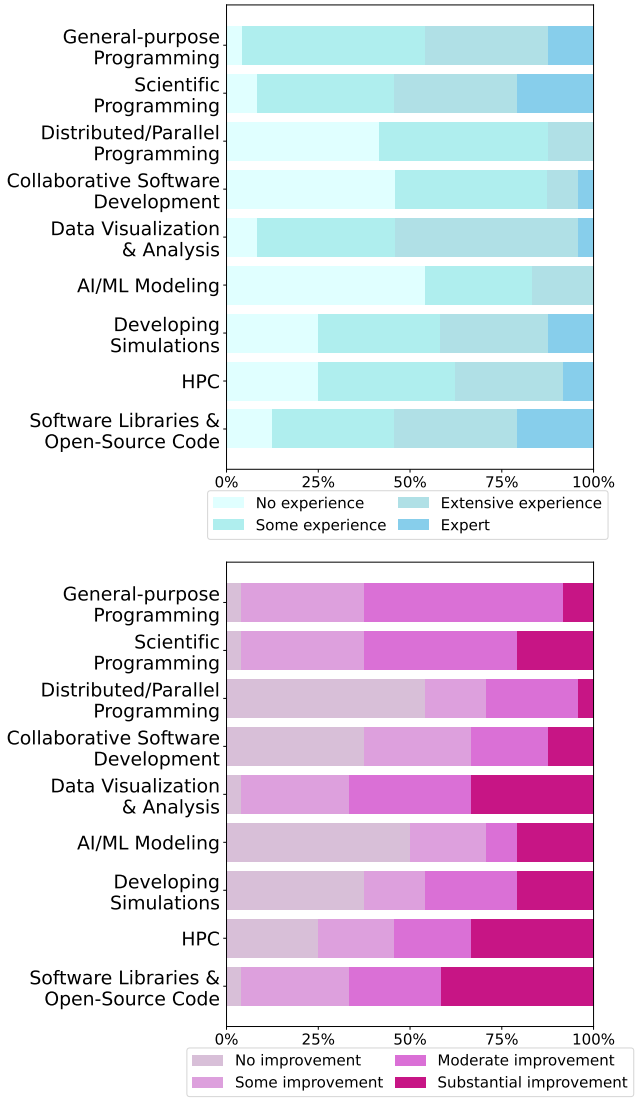


Figure 1: Participants' self-reported prior experience (left) and improvement of computational skills (right).

While the improvements were not statistically significant across all skills, we note that 21 participants reported "moderate improvement" or "substantial improvement" in multiple skills.

The open-ended responses mentioned additional specific skills such as MPI (Message Passing Interface) and manifold optimizations.

3.2 Familiarity with Domain Science topics

On average, participants enter their internships with some familiarity with the domain science topics of energy systems and energy efficiency, and little to no familiarity with other topics. Participants report at least "some increase" in familiarity in renewable energy ($t(23) = 5.36, p = .000$) and energy efficiency

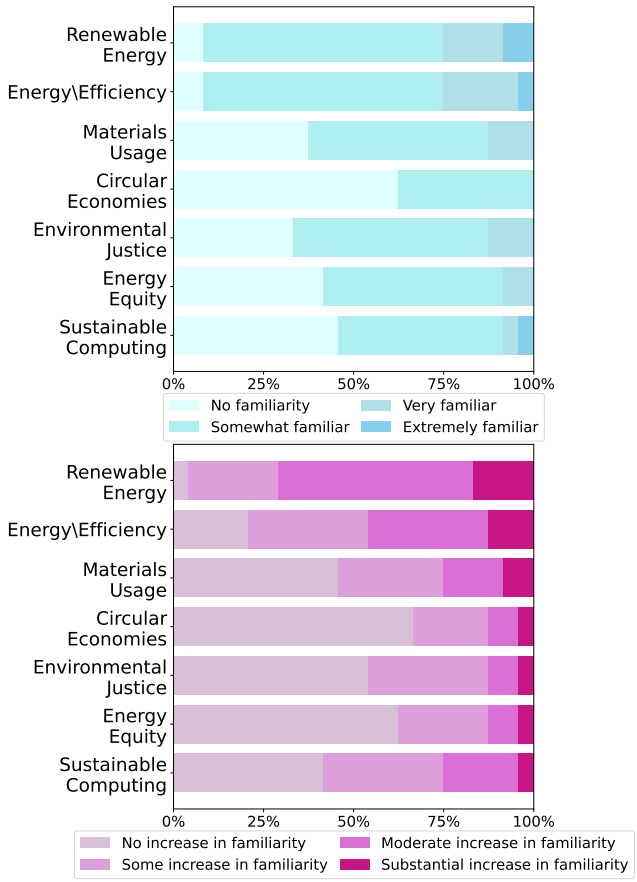


Figure 2: Participants' self-reported prior familiarity (left) and improvement of familiarity (right) with domain science topics.

($t(23) = 1.89, p = .035$). The distribution of responses is provided in Figure 2.

While the increases in familiarity were not statistically significant across all skills, we note that six participants reported "moderate increase" or "substantial increase" in one topic, and 13 participants reported "moderate increase" or "substantial increase" in multiple topics.

The open-ended responses echoed additional computational skills (e.g., computational methods for physics and chemistry applications) and other domain science topics (e.g., wind energy, power flows).

3.3 Research Skills

On average, participants enter their internships with at least one prior research experience with respect to working on research projects, in research teams, and toward authoring publications. Additionally, participants report working on at least one research project ($t(23) = 2.94, p = 0.004$) or at least one research team during their internship ($t(23) = 2.41, p = 0.012$). The distribution of responses is provided in Figure 3.

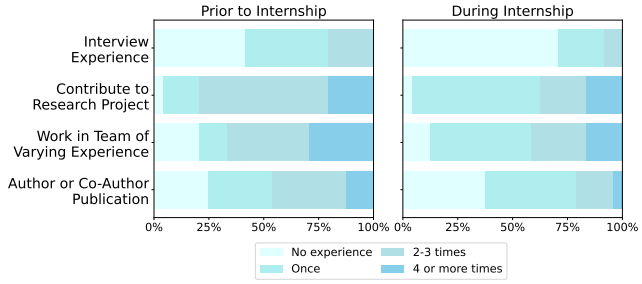


Figure 3: Participants' self-reported research experience before (left) and during (right) their internship.

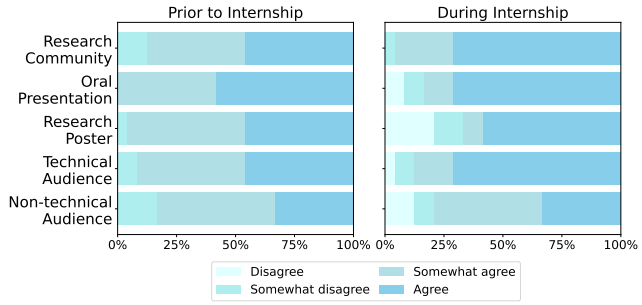


Figure 4: Participants' self-reported agreement with being part of a research community and with ability to communicate ideas before (left) and during (right) their internship.

The open-ended responses echoed similar experiences as our questions. For example, one participant noted "Remote collaboration" as a skill that improved, and another participant mentioned working in teams with varying levels of expertise as an important skill for their post-internship position.

3.4 Professional Skills

On average, participants enter their internships with the ability to communicate research across a variety of modes and are already part of research communities. Participants' agreement with the statement, "I'm part of a research community where I can get feedback on my work," is higher during their internship compared to before their internship ($t(23) = -1.88, p = .036$). While not statistically significant, we also note that more participants responded both "Agree" and "Disagree" with respect to these skills during their internship compared to before their internship. The distribution of responses is provided in Figure 4.

Five participants also mentioned networking or the networks they developed through their internships as professional skills in the open-ended responses.

On average, participants are interested across a variety of domains and sectors prior to their internships, except for non-profits or other unspecified sectors. Additionally, interest across all domains increased after participants' internships, and no participants responded "Not interested" to any of the domains. Across sectors, participants were more interested in careers at NLR ($t(23) =$

$-4.14, p = 0.000$) and other Department of Energy national laboratories ($t(23) = -2.60, p = 0.008$). The distribution of responses is provided in Figure 5 and Figure 6.

3.5 Career Interests

When asked about their current position at the time of completing the survey, 14 participants reported they were in the same or different degree program, and 10 participants reported they were looking for employment or currently employed. Of those in the same or different degree program, all participants reported enrollment in

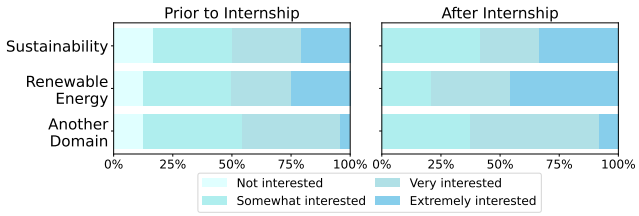


Figure 5: Participants' self-reported interest in career domain before (left) and after (right) their internship.

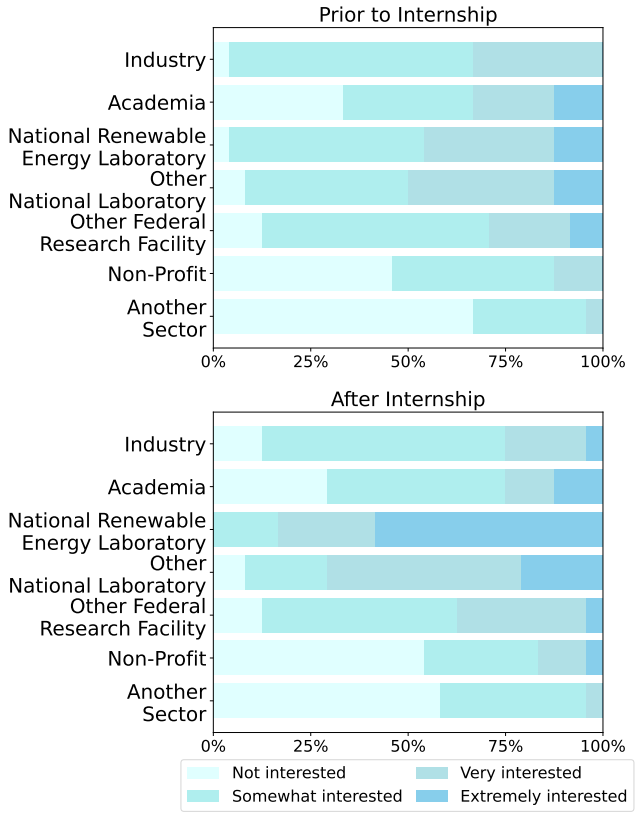


Figure 6: Participants' self-reported interest in career sector before (left) and after (right) their internship.

doctoral programs. Of those currently employed, participants reported working at DOE National Laboratories (including NLR), industry, academia, and government.

Out of the 15 participants who are not in the same degree program, only four participants reported they were looking for positions or currently in a field not related to NLR domain science topics. Additionally, participants on average reported skills improved during the internship were at least "Somewhat useful" in applying for positions.

Participants also reported skills improved during the internship were at least "Somewhat useful" in applying to their current position, and that they used these skills at least "Some of the time" in their current positions (see Table 4).

4 DISCUSSION

Students supported by federally funded graduate internships in CSE topics at NLR reported the largest improvement in computational science skills, followed by domain science skills. This improvement in computational skills may reflect both hands-on experience during their internship and the availability of tutorials at NLR. Students reported smaller improvements in their research skills. This may reflect that students often begin this internship with prior research experience through their undergraduate or graduate degree programs.

Students reported some shift in their career interests. They report increased interest in careers in the domain science topics researched at NLR. However, apart from a small increase in students interested specifically in NLR, students do not appear to have changed their interests in the career sector significantly. This may reflect that graduate students participating in internships outside of academia are more selective of the experiences they pursue to match their existing interests.

5 CONCLUSIONS

These results showed that the primary outcomes from CSE internships at NLR were increases in CSE and domain science capabilities. Improvements in research skills and career insights were present but less significant. This is the reverse of models that hold that the purpose of internships and other experiential learning is to develop non-technical skills. The increase in technical skills shown in this study suggests that internships at the postgraduate level may hold a role beyond introducing students to non-academic careers.

At a larger level, our results echo other calls to encourage graduate student participation in internships to provide supplemental training and additional learning opportunities outside of formal degree programs [30]. Our results emphasize the importance of federally-funded or other externally-funded programs in providing these internship opportunities that focus on educational objectives, rather than internships funded from research projects that may focus on hiring candidates whose skills best align with individual projects. This approach may be particularly valuable in CSE, where there is a large disparity in institutional resources and ability to offer cross-training between institutions.

To the best of our knowledge, this is the first study to evaluate graduate-level internship programs at the intersection of CSE and domain science. We encourage others to use our survey as part of

evaluation efforts of other internship programs to see if our results generalize to other host institutions and program contexts. We also suggest future work on the impacts of other models of using national laboratory capabilities to provide training in CSE and HPC, such as academic partnership programs.

ACKNOWLEDGMENTS

The authors would like to thank Addyson Sisemore, Jahi Simba, and Veronica Waller from NLR's Human Resources team for their help in identifying potential participants based on internal records; Tony Magri from NLR's IT Support team for his help in identifying interns who had worked NLR's HPC resources. We would also like to thank Ken Oswald from the Oak Ridge Institute for Science and Education for providing the full list of NSF-MSGI recipients who completed their internship at NLR.

No discussion of computational science internships at NLR is complete without mentioning Mo Nelson, who passed away as this paper wound its way through the publication process. In her 18 years as an administrator in the Computational Science Center, Mo mastered the art of clearing up the paperwork disasters that well-meaning scientists excel at creating. She did so with humor, competence, and above all, compassion. More than one of these paperwork disasters was an obstacle that had to be overcome to make an internship possible. Her skill and dedication were also critical to linking the lab with several of the programs whose outcomes are described in this paper. Being able to quantify in a peer-reviewed journal the impact of these internships is a unique opportunity to celebrate our friend and colleague.

This work was authored by NLR for the U.S. Department of Energy (DOE), operated under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NLR. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

REFERENCES

[1] Mireilla Bikanga Ada and Mary Ellen Foster. 2023. Enhancing postgraduate students' technical skills: perceptions of modified team-based learning in a six-week multi-subject Bootcamp-style CS course. *Computer Science Education* 33, 2 (2023), 186–210. https://doi.org/10.1080/08993408.2021.1959174

[2] ALCF. 2024. ALCF Lighthouse Initiative, https://www.alcf.anl.gov/lighthouse. https://www.alcf.anl.gov/lighthouse

[3] Maria Eftychia Angelaki, Fragkiskos Bersimis, Theodoros Karvounidis, and Christos Douligeris. 2024. Exploring the Awareness Outcomes of Educating ICT Students on the Environmental Implications of e-Waste Recycling and Energy Consumption in Data Centers. *IEEE Transactions on Education* 67, 1 (Feb. 2024), 143–152.

[4] Elizabeth Bautista and Nitin Sukhija. 2024. Data Analytics Program in Community Colleges in Preparation for STEM and HPC Careers. *The Journal of Computational Science Education* 15, 1 (March 2024), 59–63.

[5] Mordechai Ben-Ari. 2004. Situated Learning in Computer Science Education. *Computer Science Education* 14, 2 (2004), 85–100. https://doi.org/10.1080/08993400412331363823 arXiv:https://doi.org/10.1080/0899340041233136382

[6] Carolyn Connor, Amanda Bonnie, Gary Grider, and Andree Jacobson. 2016. Next Generation HPC Workforce Development: The Computer System, Cluster, and Networking Summer Institute. In *2016 Workshop on Education for High-Performance Computing (EduHPC)*. 32–39.

[7] Laleh E. Coté, Astrid N. Zamora, Julio Jaramillo Salcido, Seth Van Doren, Aparna Manocha, Gabriel Otero Munoz, Esther W. Law, and Anne M. Baranger. 2025. Where are they now? Academic and career trajectories of national laboratory STEM internship alumni from community colleges, compared to those from universities. *bioRxiv* bioRxiv 2025.03.24.645145 (2025). https://doi.org/10.1101/2025.03.24.645145

[8] Laleh E. Coté, Seth Van Doren, Astrid N. Zamora, Julio Jaramillo Salcido, Esther W. Law, Gabriel Otero Munoz, Aparna Manocha, Colette L. Flood, and Anne M. Baranger. 2025. "When I talk about it, my eyes light up!" Impacts of a national laboratory internship on community college student success. *PLOS ONE* doi.org/10.1371/journal.pone.0317403 (2025). https://doi.org/10.1371/journal.pone.0317403

[9] Department of Energy. 2024. High Performance Computing for Energy Innovation, hpc4energyinnovation.llnl.gov. https://hpc4energyinnovation.llnl.gov

[10] DOE Office of Science. 2024. Science Undergraduate Laboratory Internships (SULI): Where Students Come To Change The World, science.osti.gov/wdts/suli. https://science.osti.gov/wdts/suli

[11] Abel J. Duarte, Benedita Malheiro, Elisabet Arnó, Ignasi Perat, Manuel F. Silva, Pedro Fuentes-Durá, Pedro Guedes, and Paulo Ferreira. 2020. Engineering Education for Sustainable Development: The European Project Semester Approach. *IEEE Transactions on Education* 63, 2 (May 2020), 108–117.

[12] Anshu Dubey, Katherine M. Riley, and David E. Bernholdt. 2020. Teaching Software Sustainability for High Performance Computing at ATPESC. In *2020 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC)*. IEEE, GA, USA, 19–24.

[13] Thomas M Evans, Andrew Siegel, Erik W Draeger, Jack Deslippe, Marianne M Francois, Timothy C Germann, William E Hart, and Daniel F Martin. 2022. A survey of software implementations used by application codes in the Exascale Computing Project. *The International Journal of High Performance Computing Applications* 36, 1 (2022), 5–12. https://doi.org/10.1177/10943420211028940 arXiv:https://doi.org/10.1177/10943420211028940

[14] Scott Feister and Elizabeth Blackwood. 2022. HPC Workforce Development of Undergraduates Outside the R1. *Journal of Computational Science Educatione* 13, 2 (2022), 1–11. https://doi.org/10.22369/issn.2153-4136/13/2/2

[15] Camille Ferguson, Vanora Thomas, Juan Del Toro, Daniel Light, Kamau Bobb, Peta-Gay Clarke, Shameeka Emanuel, Ed Gronke, Mary Jo Madda, and Imani Jennings. 2024. The Important Role Social Capital Plays in Navigating the Computing Education Ecosystem for Black Girls. *ACM Transactions on Computing Education* 24, 2 (April 2024), 17:1–17:17.

[16] Gabriella Coloyan Fleming, Michelle Klopfer, Andrew Katz, and David Knight. 2024. What engineering employers want: An analysis of technical and professional skills in engineering job advertisements. *Journal of Engineering Education* 113, 2 (2024), 251–279. https://doi.org/10.1002/jee.20581 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jee.20581

[17] Laura G Foltz, Sam C Gannon, and Stephanie L Kirschmann. 2011. *Making Comparisons & Examining Experiences: A Program Evaluation of the Department of Energy's Student Undergraduate Laboratory.* Doctoral Capstone Project. Vanderbilt University. https://ir.vanderbilt.edu/handle/1803/5198

[18] Indiana University Center for Postsecondary Research. 2021. The Carnegie Classification of Institutions of Higher Education, carnegieclassifications.acenet.edu/carnegie-classification/. https://carnegieclassifications.acenet.edu/carnegie-classification/

[19] Patricia Jacobs, Phillip List, Mobeen Ludin, Aaron Weeden, and Robert M. Panoff. 2014. The Blue Waters Student Internship Program: Promoting Competence and Confidence for Next Generation Researchers in High-Performance Computing. In *2014 Workshop on Education for High Performance Computing*. 49–55.

[20] Jin Kang and Audrey Girouard. 2022. Impact of UX Internships on Human-computer Interaction Graduate Students: A Qualitative Analysis of Internship Reports. *ACM Transactions on Computing Education* 22, 4 (Dec. 2022), 1–25.

[21] Amanpreet Kapoor and Christina Gardner-McCune. 2019. Understanding CS Undergraduate Students' Professional Development through the Lens of Internship Experiences. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. ACM, Minneapolis MN USA, 852–858.

[22] Mayank Kejriwal and Victoria Petryshyn. 2024. Advancing computational sustainability in higher education. *Nature Computational Science* 4, 6 (June 2024), 382–383.

[23] Kristina Kramarczuk, Kate Atchison, Monica Hilliard, Jandelyn Plane, Sally Bond, Caitlin Rudy, and David Weintrop. 2024. Micro-internships and Career Focused Programs as Mechanisms for Diversifying Computing. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. ACM, Portland OR USA, 694–700.

[24] Krell Institute. 2024. Computational Science Graduate Fellowship, www.krellinst.org/csgf/. https://www.krellinst.org/csgf/

[25] Quintin Kreth, Mary Eve Spirou, Sarabeth Budenstein, and Julia Melkers. 2019. How prior experience and self-efficacy shape graduate student perceptions of an online learning environment in computing. *Computer Science Education* 29, 4 (2019), 357–381. https://doi.org/10.1080/08993408.2019.1601459

[26] Louise Ann Lyon and Emily Green. 2020. Women in coding boot camps: an alternative pathway to computing jobs. *Computer Science Education* 30, 1 (2020), 102–123. https://doi.org/10.1080/08993408.2019.1682379

[27] M. J. Martin, A. Anderson, D. Sickinger, C. Tripp, and K. Munch. 2025. Integrating Energy-Efficient Computing with Computational Research to Accelerate Energy Technology. *Computing in Science and Engineering* 27 (2025), 18–27. https://doi.org/10.1109/MCSE.2025.3549359

[28] Lois Curfman McInnes, Paige Kinsley, Mary Ann Leung, Daniel Martin, Suzanne Parete-Koon, and Sreeranjani Jini Ramprakash. 2023. Building a Diverse and Inclusive HPC Community for Mission-Driven Team Science. *Computing in Science & Engineering* 25, 5 (Sept. 2023), 31–38.

[29] Mia Minnes, Sheena Ghanbari Serslev, and Omar Padilla. 2021. What Do CS Students Value in Industry Internships? *ACM Transactions on Computing Education* 21, 1 (March 2021), 1–15.

[30] Zachary F. Murguía Burton and Xiangkun Elvis Cao. 2022. Let graduate students do internships. *Matter* 5, 12 (Dec. 2022), 4100–4104.

[31] Marco A. S. Netto, Rodrigo N. Calheiros, Eduardo R. Rodrigues, Renato L. F. Cunha, and Rajkumar Buyya. 2018. HPC Cloud for Scientific and Business Applications: Taxonomy, Vision, and Research Challenges. *Comput. Surveys* 51, 1, Article 8 (Jan. 2018), 29 pages. https://doi.org/10.1145/3150224

[32] NNSA and ORAU. 2024. National Nuclear Security Administration Minority Serving Institutions Internship Program, orise.orau.gov/nnsa-msiip/. https://orise.orau.gov/nnsa-msiip/

[33] NSF. 2024. National Science Foundation Mathematical Sciences Graduate Internship Program (NSF-MSGI), orise.orau.gov/nsf-msgi//index.html. https://orise.orau.gov/nsf-msgi//index.html

[34] David S. Ochs and Ruth Douglas Miller. 2015. Teaching Sustainable Energy and Power Electronics to Engineering Students in a Laboratory Environment Using Industry-Standard Tools. *IEEE Transactions on Education* 58, 3 (Aug. 2015), 173–178.

[35] ORAU. 2023. 2023 High Performance Computing Summer Internships, www.zintellect.com/Opportunity/Details/DOE-EERE-HPC-2023. https://www.zintellect.com/Opportunity/Details/DOE-EERE-HPC-2023

[36] ORNL. 2024. OLCF Pathways to Supercomputing, www.olcf.ornl.gov/community/pathways-to-supercomputing. https://www.olcf.ornl.gov/community/pathways-to-supercomputing/

[37] Anne-Kathrin Peters, Rafael Capilla, Vlad Constantin Coroamă, Rogardt Heldal, Patricia Lago, Ola Leifler, Ana Moreira, João Paulo Fernandes, Birgit Penzenstadler, Jari Porras, and Colin C. Venters. 2024. Sustainability in Computing Education: A Systematic Literature Review. *ACM Transactions on Computing Education* (Jan. 2024), 3639060.

[38] Ulrich Rüde, Karen Willcox, Lois Curfman McInnes, and Hans De Sterck. 2018. Research and Education in Computational Science and Engineering. *SIAM Rev.* 60, 3 (Jan. 2018), 707–754.

[39] M. Sanchez. 2024. Students Harness Power of New Kestrel Supercomputer for Cutting-Edge Energy Research and Innovation. https://www.nrel.gov/news/program/2024/students-harness-the-power-of-new-kestrel-supercomputer-for-cutting-edge-energy-research-and-innovation.html

[40] Angela B. Shiflet. 2002. Computer science with the sciences: an emphasis in computational science. *ACM SIGCSE Bulletin* 34, 4 (Dec. 2002), 40–43.

[41] Winona Snapp-Childs, Claudia M. Costa, Daniel Olds, Addison Snell, Julie A. Wernert, and Craig A. Stewart. 2025. Economic value of HPC experience for new STEM professionals: Insights from STEM hiring managers. *Frontiers in Research Metrics and Analytics* 9 (Jan. 2025), 1–17.

[42] Jenett Tillotson. 2017. A Summary of the Workshop on Training and Sustaining Research Computing Systems Professionals: A Path to a Profession. In *Proceedings of the HPC Systems Professionals Workshop*. ACM, Denver CO USA, 1–6.

[43] TOP500.org. 2024. Top 500 The List, November 2024, top500.org/lists/top500/2024/11/. https://top500.org/statistics/overtime/

[44] U.S. Department of Education. 2024. Eligibility Designations for Higher Education Programs, www.ed.gov/grants-and-programs/grants-higher-education/eligibility-designations-higher-education-programs. https://www.ed.gov/grants-and-programs/grants-higher-education/eligibility-designations-higher-education-programs

[45] Gregor Verbič, Chanaka Keerthisinghe, and Archie C. Chapman. 2017. A Project-Based Cooperative Approach to Teaching Sustainable Energy Systems. *IEEE Transactions on Education* 60, 3 (Aug. 2017), 221–228.

[46] Joy Watson and Jed Lyons. 2011. Aligning Academic Preparation of Engineering Ph.D. Programs with the Needs of Industry. *International Journal of Engineering Education* 27, 6 (2011), 1394–1411.

[47] Lucas A. Wilson and S. Charlie Dey. 2016. Computational Science Education Focused on Future Domain Scientists. In *2016 Workshop on Education for High-Performance Computing (EduHPC)*. IEEE, Salt Lake City, UT, USA, 19–24.

[48] Remco Zwetsloot and Jack Corrigan. 2022. AI Faculty Shortages: Are U.S. Universities Meeting the Growing Demand for AI Skills? (Jan. 2022). https://doi.org/10.51593/20190049

Leveraging HPC in University Courses: A Cross-Disciplinary Workforce Development Model

F. Alex Feltus
Clemson University
ffeltus@clemson.edu

Jon C. Calhoun
Clemson University
jonccal@clemson.edu

Rong Ge
Clemson University
rge@clemson.edu

Jill Gemmill
Clemson University
gemmill@clemson.edu

Carlos Toxtli Hernandez
Clemson University
ctoxtli@clemson.edu

Bing Li
Clemson University
bli4@clemson.edu

Todd Allen Price
Clemson University
tprice2@clemson.edu

Justin Talbot
Clemson University
jjtalbo@clemson.edu

MinJae Woo
Clemson University
mwoo@clemson.edu

ABSTRACT

High-performance computing (HPC) is an increasingly essential skill across science, engineering, and data-intensive industries, yet most university students and faculty lack prior HPC experience. This paper presents a cross-disciplinary workforce development model in which nine faculty members at Clemson University integrated the Palmetto2 supercomputer (3.0 PFlops; 54,476 CPU cores; 300+ GPU nodes) into seven courses spanning Bioinformatics and Computational Biology, Business Data Analytics, Computational Chemistry, Computer Science and Parallel Architectures, Automotive Engineering and AI Applications, Healthcare Data Science, and HPC Systems Hardware. A democratized access policy—removing departmental and financial barriers to cluster accounts—enabled broad adoption across student populations ranging from wet-lab biologists to working business professionals. Common pedagogical challenges included HPC onboarding, software dependency management, and large-dataset transfer; shared solutions included priority course scheduling, scaffolded desktop-to-cluster exercises, and modular training materials developed by Clemson's Research Computing and Data Engagement (RCDE) team. Discipline-specific computational profiles varied from large-dataset I/O in bioinformatics and healthcare to GPU-intensive deep learning in automotive AI and intensive floating-point computation in chemistry and parallel architectures. Three transferable principles emerge: democratized access, guaranteed priority scheduling for courses, and domain-contextualized scaffolded exercises. Students completing these courses gain Linux proficiency, job scheduling, parallel programming, and large-scale data analysis skills directly applicable to research and industry careers in an economy increasingly dependent on data-intensive computation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/9

KEYWORDS

High-performance Computing, Workforce Development, Interdisciplinary Education, University Curriculum, Research Computing

1 INTRODUCTION

In the Fall 2025 semester at Clemson University, a diverse group of professors shared their experiences teaching undergraduate and graduate students using a campus cluster called Palmetto2 [3] in their university courses. Course topics included Bioinformatics and Computational Biology, Business Data Analytics, Computational Chemistry, Computer Science and Parallel Architectures, Automotive Engineering and AI Applications, Healthcare Data Science and Public Health. After Clemson's HPC Days workshop, the group saw value in presenting to a wider audience.

In this report, an overview of each course is presented along with a hardware training course called "Constructing HPC Systems: The Hardware Perspective." Then we provide a comparative analysis along with challenges and education solutions that are common for most, if not all disciplines. Finally we discuss workforce development opportunities.

1.1 Current State of HPC Education in Universities

High-performance computing (HPC) has become a strategic resource for scientific discovery, economic competitiveness, and national security, yet its integration into university curricula remains uneven. A survey of 133 domestic and international university HPC centers found that while HPC use in undergraduate education has grown significantly across STEM disciplines, the majority of students (63.5%) and faculty (69.0%) surveyed reported no prior HPC experience [4]. Key barriers include the absence of dedicated curricular materials, insufficient access to HPC hardware, and a lack of technical support staff. HPC-related skills are rarely embedded in standard undergraduate programs, and research opportunities are largely confined to campuses with existing HPC infrastructure [4].

1.2 National HPC Resources

Several United States initiatives have been established to address the HPC access and workforce development gap, providing training

materials, allocations, and programmatic support to researchers and educators across the United States.

1.2.1 National Science Foundation Cyberinfrastructure Worker Training Programs. The National Science Foundation (NSF) Training-based Workforce Development for Advanced Cyberinfrastructure (CyberTraining) program (NSF 23-520) is the flagship federal initiative for cyberinfrastructure (CI) workforce development [6]. The program pursues two primary goals: (i) ensuring broad adoption of CI tools, methods, and resources by the research community to catalyze major research advances; and (ii) integrating core literacy and discipline-appropriate advanced skills in CI and computational methods into undergraduate and graduate educational curricula. In 2021, NSF published the *Blueprint for Cyberinfrastructure Learning and Workforce Development* [7]. The blueprint defines three communities of concern: CI Contributors, CI Professionals, and CI Users and articulates a three-pillar architecture for sustainable CI community building: (1) promoting community vision and organization, (2) fostering sustainability through professional development and career pathways, and (3) measuring and communicating the impact of CI investments. The blueprint calls for curriculum development, career recognition, and the nurturing of regional and domain-centered CI communities.

1.2.2 XSEDE and ACCESS Programs. The Extreme Science and Engineering Discovery Environment (XSEDE), funded by NSF from 2011 to 2022, provided a national network of advanced computing resources, training, and user support to the U.S. research community [8]. XSEDE offered online lectures, workshops, and training distributed across its partner sites, and served as a model for multi-institutional HPC education. In 2022, XSEDE was succeeded by the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) [1], which continues to provide NSF-funded advanced computing allocations and support services at no cost to eligible U.S.-based researchers and educators.

1.2.3 The Carpentries HPC Training Materials. HPC Carpentry is a community-driven initiative that develops and delivers open, peer-reviewed training materials for high-performance computing, modeled on the pedagogy of The Carpentries (Software Carpentry, Data Carpentry, Library Carpentry) [5]. The core HPC Carpentry curriculum bridges the gap between laptop-scale computing and supercomputer use through three lessons: the Software Carpentry Unix Shell lesson, an *Introduction to High Performance Computing* lesson covering job scheduling, module systems, file transfer, and parallel execution, and a lesson on HPC Workflow Management using Snakemake. Materials are freely available, instructor-tested, and designed for two-day workshop delivery. The mission of HPC Carpentry is to lower the barrier to entry for HPC operations across a wide range of users, particularly those without formal computational training.

2 PALMETTO2: DEMOCRATIZING HPC AS A TRAINING RESOURCE

2.1 Infrastructure Overview

Named for South Carolina's state tree, the Palmetto2 Cluster was designed to suit many different research applications, with many powerful multi-core nodes, each with a significant amount of memory. Currently, Palmetto2 is comprised of 1202 compute nodes (totaling 54,476 CPU cores), and features:

- over 300 nodes are equipped with 2x or more NVIDIA Tesla GPUs (P100, V100, A100, H100, and H200)
- 20 large-memory nodes with greater than 1.5 TB of memory
- 250 GB home directory space for each user
- 5 TB scratch storage space for each user
- 10 and 25 Gbps Ethernet, and 56, 100, 200 and 400 Gbps Infiniband networks
- benchmarked at 3.0 PFlops

GPU-accelerated nodes are interconnected with high-speed Infiniband networking. Palmetto2 also contains special resources such as large-memory nodes featuring 1.5TB of RAM or more with the largest containing 6TB, and NVIDIA DGX and HGX systems for large-scale AI/ML workloads.

The Indigo Data Lake storage system is a research data repository and processing platform backed by an all-flash scalable storage system provided by VAST Data. Indigo has 5.5PB of data leveraging data reduction and de-duplication technologies. The data lake can support large numbers of simultaneous clients by leveraging an aggregate of 1.6 Tbps of network throughput over InfiniBand and 400 Gbps over ethernet. Indigo is accessible via NFS, SMB, or S3 protocols. Students, faculty, and staff can access their storage via the SMB protocol over the campus network. Faculty can purchase persistent storage on Indigo that also features backups and snapshots. The Palmetto2 HPC cluster is housed at Clemson's Information Technology Center (ITC). The ITC is a 24/7 monitored environment with proper power, cooling, and physical security. The Palmetto Cluster is UPS and generator backed to prevent unexpected interruptions to compute jobs.

2.2 Democratized Access Model and User Experience

All Clemson HPC resources (compute, storage, and staff support) are available across all departments, to all faculty and students. Faculty and graduate students simply need to request an account; undergraduate students are given accounts through faculty request, such as for a course or supervised research work. This democratized access model makes computing available to everyone, from engineering to humanities and the social sciences. User jobs are scheduled on the Slurm scheduler's fair-share algorithm. A shared ownership model (condominium model) is available for faculty who require on-demand compute resources. Compute node owners have priority access to queue jobs on their nodes, and other users on the cluster may take advantage of idle time on the node to run their jobs.

Courses using HPC are assigned priority scheduling, so that resources are guaranteed to be available during class hours if needed,

and the user support and training infrastructure provides the assistance users need to become proficient in their computing needs. This has been especially important for disciplines that do not typically include training in computing as part of their core curriculum. Integration with Clemson's Canvas course management system vastly simplifies account management for these students.

2.3 Clemson HPC Workshop Training Series

The Research Computing and Data Engagement (RCDE) team at Clemson University provides training for the HPC community. Most workshops are available as an online course, which supplements live training sessions. The online course allows people to review the materials at their own pace. Course topics include introductory materials (Intro to Linux, Intro to HPC); advanced HPC topics (Containers for HPC; Kubernetes; parallel methods); materials simulation codes (LAMMPS, VASP); software development (Git; CI/CD; AI code assistance); Python (Spark; Deep Learning; Machine Learning; Visualization); R (programming, Machine Learning/caret); Nextflow workflow manager; and Deep Learning workshops (running and fine-tuning LLMs on Palmetto; PyTorch; RAGs; Attention & Transformers).

3 INTERDISCIPLINARY HPC IMPLEMENTATION: DISCIPLINE-SPECIFIC UNIVERSITY COURSES

3.1 Bioinformatics and Computational Biology

3.1.1 Course Structure and HPC Integration. The Genetics and Biochemistry department at Clemson University offers multiple bioinformatics courses that demonstrates a scalable, multi-level approach to integrating HPC into life sciences education using the Palmetto2 HPC cluster. *Advanced Medical Bioinformatics* (GEN 8450) is a graduate course, delivered in a hybrid format over 15 weeks to 10-20 students, emphasizing applied genomics workflows for translational research applications. *Foundations of Genetics and Bioinformatics* (GEN 8410) provides an intensive 5-week summer foundation course for 10 students in a fully asynchronous online format and establishes core computational biology competencies. A *Bioinformatics* course (BIOCH/GEN 4400/6400) accommodates 100-150 undergraduate and graduate students annually in a hybrid self-paced format, serving as a workforce development pipeline for computational biology skills across multiple disciplines.

3.1.2 Learning Objectives and HPC Skill Development. All three courses utilize Palmetto2's computational resources to achieve progressive learning objectives spanning theoretical knowledge and hands-on technical skills. Students master Linux command-line operations for bioinformatics workflows, develop proficiency in Python and R programming for genomic data analysis, and learn to navigate high-performance computing environments including Slurm job scheduling, resource allocation, and parallel processing concepts.

The curriculum emphasizes real-world dataset manipulation using major biological databases (SRA, ENSEMBL, GTEx, NCI GDC), implementation of sequence alignment algorithms, and systems biology approaches for network construction and pathway analysis. Students engage in biomarker discovery workflows and population genetics analyses requiring substantial computational resources unavailable on personal workstations.

3.1.3 Educational Outcomes and Workforce Impact. The integration of Palmetto2 infrastructure enables students to tackle authentic research problems with real-world computational complexity, bridging the gap between classroom theory and professional practice. Students demonstrate measurable competency gains in computational thinking, scalable data analysis, and interdisciplinary problem-solving essential for modern life sciences careers.

The complexity across the three courses—from 10-student intensive workshops to 150-student large-enrollment implementations—establishes a replicable model for HPC integration that addresses diverse institutional needs and student populations. Assessment outcomes show consistent achievement of technical proficiency in genome-scale data processing, statistical analysis programming, and collaborative computational workflows that directly translate to research productivity and industry readiness in bioinformatics, biotechnology, and healthcare sectors. This multi-tiered approach demonstrates how HPC resources can be effectively scaled from specialized graduate training through broad undergraduate workforce development, providing a sustainable model for other institutions seeking to integrate computational infrastructure into life sciences curricula.

3.1.4 Example Computational Exercise: Sequence Alignment on Palmetto2. Students complete a comprehensive hands-on laboratory exercise demonstrating core computational biology workflows on high-performance computing infrastructure [2]. The exercise encompasses ten sequential steps that progressively build HPC competency: (1) secure cluster access and authentication, (2) interactive resource allocation using Slurm job scheduling, (3) Jupyter interface navigation for terminal access, (4) scratch filesystem utilization for temporary storage, (5) conda environment management for bioinformatics software installation, (6) large-scale genomic dataset acquisition from public repositories, (7) file compression and decompression operations, (8) BLAST database construction using `makeblastdb`, (9) comparative sequence alignment execution using both global (Needleman-Wunsch) and local (Smith-Waterman) algorithms, and (10) quantitative analysis of alignment results including statistical significance assessment.

This exercise directly addresses authentic research scenarios by utilizing human genome and cDNA datasets, requiring computational resources exceeding typical desktop capabilities. Students gain practical experience with Linux command-line operations, bioinformatics software deployment, parallel processing concepts, and large-scale data managemen that are essential skills for modern genomics research. The progressive complexity from basic cluster access to advanced alignment analysis demonstrates how HPC infrastructure enables sophisticated biological analyses previously accessible only to specialized research groups, effectively democratizing computational biology education across diverse student populations.

3.2 Business Data Analytics

3.2.1 Course Structure and HPC Integration. Effective management in the information age relies on the proper use and interpretation of data. Virtually every functional area of management relies on the statistical treatment of data and information. Knowing when and how to use statistics as a tool to make decisions in an uncertain business environment is an essential skill of a successful manager. *Business Data Analytics and Statistical Computing* (MBA 8040) teaches modern statistical inference in business operations. Topics include testing statistical hypotheses, consequences of making decisions with incomplete information, univariate and multivariate regression with emphasis on business applications and design of experiments and analysis of variance. Special attention is given to efficient and relevant data collection and interpretation.

3.2.2 Learning Objectives and HPC Skill Development. The objectives of this 3 credit course are to present the most important statistical techniques and illustrate their applications. Emphasis is placed on appropriate formation of problems, proper choice of statistical techniques, and on effective interpretation and communication of results. Conceptual understanding is facilitated by the use of technology to show how each of the statistical techniques can be implemented in spreadsheet (MS Excel) or HPC environments. Using Palmetto2, students install R and Python working environments, and explore calculation structures. By the end of the course, students recognize how to deal with situations that involve uncertainty using local and HPC technologies as well as understand how to apply and to interpret regression analysis and other managerially important analytical models.

3.2.3 Educational Outcomes and Workforce Impact. This class provides an entrance into HPC where most students do not have not have any experience. By allowing students to see what could be done with compute capabilities, they can then use HPC not only for their assignments, but also discuss with their current employers on how HPC can benefit their current work areas.

3.2.4 Example Computational Exercise. This course gives students access to Python and R notebook environments where they can apply statistical methodologies to business cases with provided datasets for business decision-making. An example teaching workflow is to (1) Access Palmetto2 using Open OnDemand, (3) open a terminal application, (3) create a working directory in scratch space, (4) install Python, (5) download case datasets into working directory on scratch, (6) uncompress data files, and (7) apply numpy and other libraries to analyze the dataset.

3.3 Computational Chemistry

3.3.1 Course Structure and HPC Integration. Computational Chemistry (CHE 8380) is a mixed lecture/lab course that introduces graduate students to various aspects of computational chemistry including fundamental computational techniques such as Hartree-Fock and density functional theory, energy decomposition analysis, ab initio molecular dynamics, and computational thermochemistry and spectroscopy. The course is explicitly designed to train students in the use of HPC resources and architectures, including remote environments, Linux-based workflows, batch scheduling, resource allocation, and basic concepts of scalability. While representative calculations in laboratory activities are executed on students' personal computers, the exercises are framed as small-scale analogs of workflows that are later deployed on HPC systems. The final project consists of a student-led computational project carried out entirely using HPC resources in which an insightful chemical question is investigated. The final project is intended to simulate a graduate-level research environment in computational chemistry.

3.3.2 Learning Objectives and HPC Skill Development. The primary learning objective of CHE 8380 is to equip students with the tools needed to translate abstract chemical concepts and hypotheses into computable quantities. Students develop a practical competency in HPC-based scientific computing and learn to interact with Linux-based systems, prepare and submit batch jobs, manage computational output, and construct reproducible computational workflows. The computational skills gained in this course are introduced incrementally through a progression of in-class lab activities which emphasize the scientific method: students learn how to formulate chemical questions, construct physical models to gain relevant insights, and implement those models computationally, while maintaining a conceptual understanding of the underlying mathematical formalism.

3.3.3 Educational Outcomes and Workforce Impact. Students completing the course demonstrate the ability to translate chemical research questions into HPC workflows. In addition to domain-specific knowledge in electronic structure theory and molecular modeling, students gain experience with technical practices common in HPC environments, including remote job execution, file system management, and post-processing of large datasets. The course therefore primarily prepares students to engage productively with HPC resources in graduate research, although the skills obtained are applicable to employment in industry.

3.3.4 Example Computational Exercise: Potential Energy Surfaces. An example of how HPC-oriented workflows are introduced at smaller scale: students complete a laboratory exercise in which they construct small molecules and perform electronic structure scans of potential energy surfaces by varying a single internal coordinate. These calculations are executed on a personal computer, but the workflow mirrors that used in HPC environments: students generate multiple independent calculations, organize a set of output files, and analyze trends in energy and molecular properties. This exercise prepares students for the final HPC project, where similar analyses are performed at a greater computational scale.

3.4 Computer Science and Parallel Architectures

3.4.1 Course Structure and HPC Integration. Parallel Architectures (CPSC8200) is a graduate-level course that focuses on the core elements of high-performance computing. Parallel computing has long been indispensable, enabling scientific discovery and engineering innovation, and it is also critical to the rapidly growing field of deep learning and its applications. Parallel computer architectures are ubiquitous today, spanning systems from mobile devices (e.g., smartphones and laptops) to large-scale data centers. The course covers modern parallel computer architectures, including multicore and manycore CPUs and GPUs, networking, system software,

parallel programming paradigms and libraries, algorithm design, and the measurement, analysis, evaluation, and optimization of performance and scalability.

3.4.2 Learning Objectives and HPC Skill Development. Programming assignments and course projects are performed on the Palmetto2 cluster. Upon completion of the course, students master the principles and design methods for parallel system from hardware to applications across the software stack. They become familiar with state-of-the-art computer architecture, systems, and applications, and they understand the technical details of software for parallel computing. Students can apply a range of parallel programming models—including SIMD, OpenMP, GPU programming with CUDA, and the Message Passing Interface (MPI)—to solve important problems. They also learn performance-analysis methodologies such as Amdahl's law, strong scaling, and weak scaling, as well as tools such as hardware performance counters, Nsight Systems (nsys), and Nsight Compute (ncu), to profile and evaluate performance and scalability and to identify inefficiencies and bottlenecks. In practice, they gain experience with batch scheduling systems to request resources and to submit interactive and batch jobs on compute clusters.

3.4.3 Educational Outcomes and Workforce Impact. By the end of CPSC 8200, students demonstrate practical readiness for advanced research and professional practice in parallel and distributed systems and high-performance computing. Because all assignments and projects are executed on the Palmetto2 cluster, students gain hands-on experience with realistic constraints—resource allocation, queueing policies, job scheduling, and performance variability—mirroring both academic and industry HPC environments.

The training directly supports workforce development by preparing students for research careers (MS, PhD) in parallel and distributed systems, HPC architecture and software, and performance analysis. They also prepare students for industry roles that demand strong systems and performance expertise, including *performance engineer*, *HPC/software engineer*, *systems engineer*, *GPU/CUDA engineer*, and *computational scientist*. Graduates leave the course with demonstrable skills in parallel programming, profiling and optimization, scalability analysis, and cluster-based execution—competencies that are increasingly essential across scientific computing, engineering simulation, and large-scale machine learning.

3.4.4 Example Computational Exercise: Accelerating a Matrix-Multiply Kernel on GPUs. Students are provided with a naïve GPU implementation of matrix multiplication, along with relevant technical papers, and are asked to apply multiple optimization techniques to accelerate execution. Students submit the optimized source code and a report describing the techniques used, the corresponding speedups, and the cumulative speedup achieved. The final performance target is to be comparable to an optimized CUDA library call (e.g., cuBLAS). A successful solution requires exposing sufficient parallelism, choosing an effective thread/block organization, making optimal use of shared memory, coalescing global-memory accesses, and managing register usage. Students also use profiling tools to identify performance bottlenecks—such as high cache-miss rates, low occupancy, and memory stalls—and to guide further optimization.

3.5 Automotive Engineering and AI Applications

3.5.1 Course Structure and HPC Integration. With continued advances in computing capabilities and increasing automotive system complexity, the Department of Automotive Engineering incorporates Clemson's Palmetto2 Cluster platform for BS, MS, and PhD program curricula at Clemson University International Center for Automotive Research (CU-ICAR)

Machine Perception and Intelligence (AUE 8200) is a graduate-level course designed to introduce students in automotive engineering, rather than computing backgrounds to modern perception and AI methods for intelligent vehicles and manufacturing systems. The course is structured in three coherent themes: sensing, 2D and 3D machine perception and AI methods, and covers physical principles of sensing, geometric representations and data-driven perception approaches for automotive and robot systems. HPC resources are incorporated throughout live lab sessions, homework assignments and course projects to support computationally intensive perception workflows, such as AI model training and inference, sensor data processing, and simulation data generation.

By working with real-world automotive and robotic datasets such as KITTI, Waymo, and nuScenes, students develop and evaluate perception pipelines that closely align with current research and industry practice. Grounded in problem-driven pedagogy, the course integrates multi-modal sensing, geometric learning, AI training and inference, and edge-AI deployment with hands-on project development, visualization and validation to support student experiential learning (ExL) cycle.

3.5.2 Learning Objectives and HPC Skill Development. The learning objectives of this course emphasize the development of applied AI and analytical skills required for modern automotive and machine perception applications. Students are trained to formulate sensing and machine vision problems, select baseline models, design and validate AI-based perception methods leveraging HPC resources. Students develop applied HPC skills such as Linux-based system usage, Bash commands and coding, and Python programming. The course introduces deep learning concepts and model development using PyTorch, with an emphasis on training and inference for perceptive sensor data recognition. Students implement Python-based workflows for processing multimodal sensor data such as radar, 2D and 3D camera and LiDAR data. Widely adopted libraries for Python, 2D and 3D processing and visualization such as NumPy, Matplotlib, scikit-image, OpenCV, and Open3D are used to support these workflows. Students learn and consolidate these skills through course projects involving the design, implementation, and evaluation of AI-enabled perception methods using real-world datasets and HPC platforms.

3.5.3 Educational Outcomes and Workforce Impact. The educational outcomes are closely aligned with the technical and professional workforce training needs in the automotive and intelligent systems fields. Through experiential and project-based learning, students will develop the ability to integrate sensing, geometric perception, and AI methods under practical computational constraints. Students will gain hands-on experience using GPU-accelerated HPC environments, learning how to interpret complex sensor data and

evaluate data-driven perception methods. In addition to technical skills, the course also emphasizes professional engineering competencies, including teamwork, technical communication, and project report writing. These outcomes will enable students to effectively participate in interdisciplinary teams in autonomous vehicles, robotics in smart manufacturing and data-driven engineering problems, where scalable computing, data-centric reasoning and system-level understanding are essential.

3.5.4 Example Computational Exercise: Machine Visual Recognition. A typical machine vision exercise in the course involves the development, training and evaluation of a convolutional neural network (CNN) using PyTorch on the CIFAR-10 dataset. CIFAR-10 is a widely used benchmark dataset for image classification, comprising ten semantic categories: {automobile, truck, ship, airplane, bird, cat, deer, dog, frog, horseİn this exercise, students first examine the dataset structure and perform basic data preprocessing and visualization using Python-based tools. They then fine-tune or design a CNN architecture, implement it within the PyTorch framework, and train the model using GPU-accelerated HPC resources, employing cross-entropy loss in conjunction with a softmax non-linear function at the output layer. Model performance is evaluated using classification accuracy, confusion matrices, and loss convergence. Students iteratively refine network architectures and training parameters to improve performance. Through this exercise, students gain hands-on experience with deep learning–based visual recognition workflows, model evaluation, and scalable HPC practices relevant to machine perception and intelligence applications.

3.6 Healthcare Data Science and Public Health

3.6.1 Course Structure and HPC Integration. Instruction of healthcare data science is implemented through *Predictive Analytics in Healthcare* (HLTH 8900) , a graduate-level course serving doctoral students from Applied Health Research and Evaluation (AHRE) and related public health programs at Clemson University. The course was designed for health science students with no prior programming or computing background and delivered over a full semester with an application-driven structure. Instruction progressed from foundational Python programming to the development, evaluation, and interpretation of machine-learning models applied to real clinical and public health datasets. HPC resources were integrated through the Palmetto2 cluster to support scalable model training, reproducible experimentation, and exposure to research-grade computational environments. Rather than positioning HPC as a standalone technical topic, cluster resources were embedded into the analytic workflow once students moved beyond toy datasets. As project complexity increased, Palmetto2 enabled students to work with large national datasets, ensemble learning methods, and repeated model comparisons that exceed the limits of personal laptops. This approach ensured that HPC usage was introduced as a practical necessity tied directly to analytic objectives rather than as an abstract infrastructure concept.

3.6.2 Learning Objectives and HPC Skill Development. The primary learning objective of HLTH 8900 is to equip public health students with the ability to design, implement, and critically interpret predictive models in healthcare contexts. Students developed proficiency in Python-based data science workflows, including data cleaning, feature engineering, statistical modeling, and machine-learning pipelines using libraries such as pandas, scikit-learn, statsmodels, and XGBoost. HPC skill development focused on applied competency rather than systems-level optimization. Students learned to execute reproducible analyses in shared computing environments, manage computational experiments at scale, and reason about tradeoffs between model complexity, interpretability, and computational cost. Exposure to Palmetto2 enabled parallel model training, repeated cross-validation, and survival modeling workflows that would be impractical on local machines. Importantly, students learned to align computational choices with clinical and public health constraints, such as data availability, contextual missingness, and interpretability requirements.

3.6.3 Educational Outcomes and Workforce Impact. Student outcomes demonstrated that health science trainees can rapidly acquire advanced data science competencies when instruction is grounded in domain-relevant problems and supported by scalable computing infrastructure. Final projects and posters showed strong technical execution, including ensemble classification, survival analysis, and model comparison, along with sophisticated contextual reasoning about clinical workflows, population health implications, and ethical considerations. Projects included cardiovascular disease risk prediction, diabetes classification using national surveillance data, and cancer survival modeling using both traditional statistical methods and modern machine learning approaches. Across these domains, students demonstrated the ability to balance predictive performance with parsimony, interpret model outputs in clinically meaningful terms, and articulate how predictive tools could inform screening, prevention, and intervention strategies. From a workforce development perspective, this implementation highlights how HPC can lower barriers for non-traditional computing students entering health analytics roles. Graduates leave the course prepared to collaborate effectively with data scientists, informatics teams, and clinicians, with practical experience executing scalable analyses in shared computing environments.

3.6.4 Example Computational Exercise: Predictive Modeling for Population Health Outcomes. A representative computational exercise focused on developing and evaluating predictive models for chronic disease risk using large-scale public health data. Students worked with the CDC Behavioral Risk Factor Surveillance System (BRFSS) dataset, comprising over 70,000 records and multiple demographic, socioeconomic, and health status variables. Using Palmetto2 resources, students implemented logistic regression, random forest, and XGBoost models, conducted systematic train-test splits, and evaluated performance using accuracy, AUC, and various feature importance analyses. The exercise emphasized iterative experimentation and interpretability. Students compared full feature models with reduced variable models to assess how data availability constraints affect predictive performance, a common challenge in real world public health settings. Students concluded by translating model findings into population level insights, such as identifying actionable versus non-modifiable risk factors and discussing how

predictive analytics could support scalable screening and prevention strategies in the field.

3.7 Constructing HPC Systems: The Hardware Perspective

3.7.1 Course Structure and HPC Integration. *Creative Inquiry: High-Performance Cluster Computing* (ECE 4990) is designed to be an HPC introductory course approachable to freshman, including those with limited or no programming/technical experience. Each student in the course is provided with a *Cluster Kit* which contains all of the supplies necessary to construct a four node Raspberry Pi cluster. Over the course of the semester the students follow a provided tutorial walking them through cluster assembly and installation of the key cluster software. This culminates in the student benchmarking their cluster using the LINPACK benchmark (HPL), the same benchmark used in the Top 500 List[1]. As students work their way through the tutorial, direct comparisons are made between the supplied hardware and that on large-scale systems such as Palmetto2. The *why* and *what* each software package and terminal command does are presented using high-level explanations and layman real-world analogies. Each year, the six best students in the course write and submit an application in collaboration with a hardware vendor to the Student Cluster Competition[2] held each November at Supercomputing[3]. The vendor supplies the team a cluster, allowing them to repeat the Raspberry Pi assignment, but with production-level hardware. During the competition, they team must complete a series of challenges such running benchmarks, running scientific applications, and reproducing results from a paper selected from the prior year's Supercomputing conference.

3.7.2 Learning Objectives and HPC Skill Development. Successfully completing this course provides students with a strong foundation for working with HPC systems. This enables them to translate that knowledge into their discipline and answer *What is HPC?* and *Why is HPC important to my domain and society at large?* Furthermore, students are able to list key hardware elements inside a node/rack and describe their role. Through benchmarking their Raspberry Pi clusters, they are able to understand and quantify application speedup and parallel efficiency. Finally, students are able to identify well-known HPC software in their domain.

3.7.3 Educational Outcomes and Workforce Impact. Most students who complete this course go on to enroll in graduate school. For those who compete in the Student Cluster Competition, many companies and organizations approach them about internship and job opportunities. Multiple students have obtained internships at United States Department of Energy laboratories.

3.7.4 Example Computational Exercise: Raspberry Pi Cluster Benchmarking. Students benchmark their Raspberry Pi clusters using the LINPACK benchmark (HPL) to evaluate floating-point performance and to compute speedup as they use additional cores on their cluster. Through repeated testing, the students gain hands-on experience tuning HPL parameters — e.g., process grid dimensions, problem size — to understanding how performance changes with additional resources. Two key metrics (speedup and efficiency) are introduced. Students compete among themselves to obtain the highest HPL score. The students' performance results are extrapolated to large process sizes to introduce concepts such as over-provisioning, queue wait time, and total time-to-solution.

[1] https://top500.org/
[2] https://www.studentclustercompetition.us/
[3] https://supercomputing.org/

4 COMPARATIVE ANALYSIS: CROSS-DISCIPLINARY CHALLENGES AND SOLUTIONS

Across seven disciplines, several recurring patterns emerged in how faculty integrated Palmetto2 into their courses. Despite differences in domain, student background, and computational scale, some challenges and solutions were consistent.

4.1 Common Technical Challenges

The most universal barrier was *onboarding*: configuring accounts, navigating the Linux command line, file ingress & egress, and submitting Slurm jobs for the first time. Students in MBA 8040, with little prior computing experience, faced the steepest initial learning curve, while even CS graduate students in CPSC 8200 required orientation to cluster-specific workflows. Across all courses, instructors reported that the first one to two weeks were dominated by environment setup rather than domain content.

Software dependency management was a second shared challenge. Each discipline relies on a distinct software ecosystem—bioinformatics tools (BLAST, conda environments), quantum chemistry codes (Gaussian, ORCA), deep learning frameworks (PyTorch, TensorFlow), and statistical packages (R, Python)—requiring reproducible, containerized, or module-based deployment strategies to ensure consistent student environments.

Data transfer and storage management presented practical friction, particularly in bioinformatics (large genomic datasets from NCBI and ENSEMBL) and automotive AI (multi-gigabyte KITTI and Waymo sensor datasets). Priority scheduling for course allocations, provided by RCDE (the university provided "humanware") team), mitigated resource contention during class hours and was identified by multiple instructors as critical to pedagogical success.

4.2 Discipline-Specific Computational Requirements

Table 1 summarizes the primary computational profile of each course. Bioinformatics and healthcare data science are characterized by large dataset I/O and moderate parallelism; computational chemistry and parallel architectures demand intensive floating-point computation and strong scaling; automotive AI and computer science require GPU acceleration; and business analytics prioritizes interactive notebook environments over raw throughput.

4.3 Student Background and Preparation Requirements

Student preparation varied widely. CPSC 8200 students typically hold CS bachelor's degrees and arrive with C-family and Linux familiarity. In contrast, MBA 8040 students are working professionals with no prior programming experience; the primary goal is computational exposure rather than proficiency. GEN 4400/6400

Table 1: Computational Profile by Discipline

Discipline	Primary Resource	Key Software
Bioinformatics	CPU + large storage	BLAST, conda, R/Python
Business Analytics	Interactive CPU	R, Python notebooks
Comp. Chemistry	CPU (multi-core)	Gaussian, ORCA
Computer Science	CPU + GPU (CUDA/MPI)	OpenMP, CUDA, MPI
Automotive AI	GPU (deep learning)	PyTorch, KITTI/Waymo
Healthcare Data Sci.	CPU + large storage	Python, R, SQL
HPC Systems (ECE)	Raspberry Pi cluster	HPL/LINPACK

students span a continuum from wet-lab biologists to computational scientists, requiring flexible, self-paced course designs. This diversity underscores the need for modular onboarding materials—such as the Clemson HPC Workshop Series—that can be tailored to discipline-specific entry points.

A common pedagogical solution across courses was the use of *scaffolded exercises*: small-scale desktop analogs that mirror HPC workflows before students transition to cluster execution. This approach, used explicitly in CHE 8380 and ECE 4990, reduced cognitive load and improved student confidence prior to full HPC deployment.

5 CONCLUSION

This paper presents a cross-disciplinary model for integrating HPC into university curricula, grounded in the shared experience of seven courses at Clemson University using the Palmetto2 HPC cluster The disciplines span life sciences, business, chemistry, computer science, engineering, and public health—demonstrating that HPC is not a niche resource but a broadly applicable educational infrastructure.

Three principles emerge as transferable to other institutions. First, *democratized access*—removing departmental and financial barriers to cluster accounts—is a prerequisite for broad adoption. Second, *priority scheduling for courses* ensures pedagogical reliability and instructor confidence. Third, *scaffolded, domain-contextualized exercises* are more effective than generic HPC tutorials for motivating diverse student populations.

The workforce development impact is substantial. Students completing these courses gain computational skills—Linux proficiency, job scheduling, parallel programming, large-scale data analysis—that are directly transferable to research careers and industry roles in an economy increasingly dependent on data-intensive computation [4, 7]. The Clemson model, supported by the RCDE team's training infrastructure and Palmetto2's democratized access policy, offers a replicable framework for institutions seeking to address the documented HPC skills gap [4].

Future work will focus on quantitative assessment of student learning outcomes across disciplines, expansion to humanities and social sciences, and development of open-source, discipline-specific HPC teaching modules for community adoption.

ACKNOWLEDGEMENTS

The authors thank the Clemson University Research Computing and Data Engagement (RCDE) team and the Palmetto2 support staff for their essential contributions to course delivery and student support. We also thank the students whose engagement and feedback continue to shape these courses.

REFERENCES

[1] ACCESS-CI. 2026. Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS). https://access-ci.org. NSF-funded successor to XSEDE, providing advanced computing allocations and support services to U.S. researchers and educators.

[2] F. Alex Feltus. [n. d.]. Sequence Alignment Methods. https://github.com/feltus/Sequence_Alignment_Methods Accessed: November 9, 2026.

[3] Jill Gemmill. [n. d.]. Palmetto 2 Documentation. https://docs.rcd.clemson.edu/palmetto/ Accessed: November 9, 2026.

[4] Jordan Hebert, Ryan Hratisch, Rahul Gomes, William Kunkel, Daniel Marshall, Abhimanyu Ghosh, Isabella Doss, Ying Ma, Derek Stedman, Blake Stinson, et al. 2024. High-performance computing in undergraduate education at primarily undergraduate institutions in Wisconsin: Progress, challenges, and opportunities. *Education and Information Technologies* 29, 14 (2024), 18451–18475.

[5] HPC Carpentry Community. 2026. HPC Carpentry: Open Training Materials for High-Performance Computing. https://www.hpc-carpentry.org. Community-driven open curriculum for HPC training, modeled on The Carpentries pedagogy.

[6] National Science Foundation. 2023. Training-based Workforce Development for Advanced Cyberinfrastructure (CyberTraining), NSF 23-520. https://www.nsf.gov/funding/opportunities/training-based-workforce-development-advanced/nsf23-520/solicitation.

[7] National Science Foundation, Office of Advanced Cyberinfrastructure. 2021. *Blueprint for Cyberinfrastructure Learning and Workforce Development.* Technical Report. National Science Foundation. https://nsf-gov-resources.nsf.gov/files/CI-LWD%202.pdf Fifth in the NSF OAC strategic vision series.

[8] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, and Nancy Wilkins-Diehr. 2014. XSEDE: Accelerating Scientific Discovery. *Computing in Science & Engineering* 16, 5 (2014), 62–74. https://doi.org/10.1109/MCSE.2014.80

Low-Cost Embedded Computer Vision and Neural Network-Enabled Autonomous Navigation for a Four-Wheeled Mobile Robot Using an ESP32 and Arduino Framework

Abhiram Kaakarla
Central Bucks High School South
abhiramkaakarla@gmail.com

ABSTRACT

This paper presents the design, construction, and testing of a low-cost autonomous navigation system built on the Freenove 4WD robot car. An ESP32-CAM microcontroller handles image capture and onboard machine learning while an Arduino Uno controls the four DC motors, with all hardware costing under 103 US dollars. A convolutional neural network trained through Edge Impulse was compressed to an 18.2 KB quantized model that runs directly on the ESP32, classifying 32 by 32 pixel grayscale camera frames into five sign categories: stop sign, left turn, right turn, T intersection, and forward arrow. An OpenCV image processing pipeline running alongside the neural network performs Canny edge detection and Hough line transform for lane tracking and pixel-density-based obstacle avoidance. Navigation decisions are encoded into short serial packets with CRC error checking and sent from the ESP32 to the Arduino, which converts them into PWM motor signals. Across 500 controlled test runs, the robot correctly identified signs 91.78 percent of the time under even lighting and 69.46 percent under uneven lighting. The full decision loop from camera capture to motor response averaged 66.6 milliseconds. A systematic failure analysis across all 500 trials categorized errors into hardware, environmental, and software causes; five targeted fixes reduced the overall failure rate from 36.8 percent to 11.4 percent. The project also served as a comprehensive self-directed computational science education exercise, and the paper includes a dedicated student reflection documenting technical lessons, methodological insights, and educational impact. All firmware, training data, and experimental records are publicly available for reproduction and improvement by other students and educators.

KEYWORDS

Freenove 4WD Robot Car, Hough Line Transform, Convolutional Neural Network, Edge Impulse, ESP32-CAM, Arduino Uno, OpenCV, Canny Edge Detection, PWM Motor Signals, Computational Science Education

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/10

1 INTRODUCTION

Self-driving robots require real-time firmware, image processing, machine learning, and motor control to work together inside a single physical machine. Commercial autonomous vehicle systems from companies like Waymo and Tesla use expensive sensor arrays and neural networks with hundreds of millions of parameters running on powerful onboard computers [7]. These systems are far beyond any classroom or student budget. The TinyML movement has changed what is achievable at the low end of the hardware spectrum: compact neural networks compressed to a few kilobytes can now run in real time on microcontrollers costing a few dollars [1] , [9].

The ESP32 microcontroller runs at 240 MHz, includes several megabytes of memory, and can classify camera images in real time. Paired with an Arduino Uno for motor control and an OpenCV image processing pipeline for lane tracking, the result is a functional autonomous robot for under 103 US dollars. The Freenove 4WD chassis appears in many classroom projects, mostly for simple reflectance-based line following. This paper goes further by deploying a trained neural network onto the ESP32, implementing a reliable two-board serial communication protocol, and conducting 500 controlled test trials to measure accuracy, timing, and failure modes with enough rigor to serve as a reproducible baseline.

Contributions: (i) a complete two-board robot combining neural network inference with classic computer vision; (ii) a UART serial protocol with CRC error detection between the two boards; (iii) 500-trial experimental results including per-sign accuracy, confusion matrix, per-step timing with variance, and failure analysis with before/after fix measurements; (iv) a student reflection documenting technical lessons and educational impact; (v) a public repository with all code, data, and build instructions at https://github.com/Abhiram03-2009/Low-Cost_Autonomous_Robot_Navigation_Research_Project.

2 RELATED WORK

2.1 Educational Robotics Platforms

Patil and colleagues [8] built a Raspberry Pi robot using OpenCV color filtering and the Hough transform for lane tracking, establishing that vision-based navigation is possible on inexpensive hardware. That work did not include a trained sign classifier, did not use a dual-board architecture, and did not report systematic accuracy measurements or failure analysis across many trials. Brooks [2] established the layered control principle that fast safety reflexes should override slower classification decisions, directly applied

here: the pixel-density obstacle check always takes priority over the neural network output.

2.2 TinyML on Microcontrollers

Warden and Situnayake [9] established the practical foundations of TinyML. Banbury and colleagues [1] found through benchmarking that environmental variation, especially lighting, causes more accuracy worsening than model architecture choices. Howard and colleagues [6] introduced MobileNet's depthwise separable convolutions, significantly reducing computation. Han and colleagues [5] showed INT8 quantization shrinks models fourfold with minimal accuracy loss. David and colleagues [3] described TensorFlow Lite Micro, and Edge Impulse [4] integrates these into an accessible training and deployment pipeline used in this project.

2.3 Gap Addressed

No prior work combines embedded neural inference with classic computer vision on an ESP32 and Arduino platform and reports per-sign accuracy, confusion matrix, step-by-step timing, and an itemized failure analysis with fix measurements across 500 trials at this cost tier. This paper provides all of that.

3 SYSTEM DESIGN AND IMPLEMENTATION

3.1 Hardware Architecture

Four subsystems make up the robot. The ESP32-CAM board with OV2640 camera and 4 MB PSRAM handles image capture and runs both the neural network and OpenCV pipeline. The Arduino Uno receives serial commands and drives the four motors. An L298N dual H-bridge motor driver converts PWM signals into motor power. A 7.4 V Li-ion battery pack with an LM2596 5 V regulator provides stable power to both boards. Table 1 itemizes all parts and costs. Figure 1 shows the system layout. Figure 2 shows the finished robot.

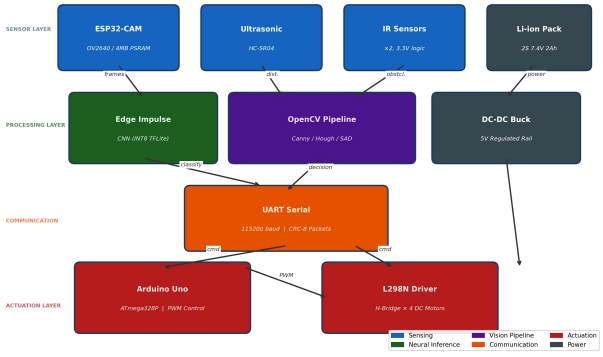


Figure 1: System architecture showing ESP32-CAM, Arduino Uno, motor driver, sensors, battery, and all data and power connections

The ESP32's two processor cores are partitioned: Core 0 handles the camera DMA and image pipeline; Core 1 handles serial communication and sensor polling. This prevented frame drops that occurred when all tasks ran on one core. A 100 microfarad capacitor across the motor power rail eliminated ESP32 brownout resets caused by current spikes when all four motors start simultaneously, a problem found through careful test logging and fixed with a single component. Per-motor PWM calibration values stored in Arduino EEPROM at startup correct for torque differences between motors that would otherwise cause the robot to drift sideways.

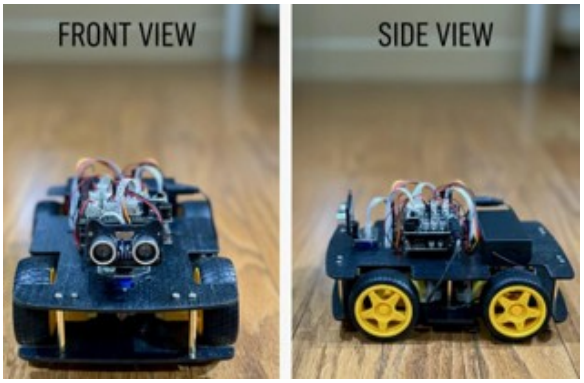


Figure 2: Finished Freenove 4WD robot with ESP32-CAM on front bracket and Arduino on chassis deck

Table 1: Parts List and Cost (USD, 2025)

Component	Notes	Cost
Freenove 4WD Kit	Chassis, motors, wheels	39.95
ESP32-CAM	OV2640, 4 MB PSRAM	8.99
Arduino Uno R3	ATmega328P, 16 MHz	27.60
Li-ion Battery Pack	7.4 V, 2000 mAh	14.99
Voltage Regulator	LM2596, 5 V out	4.99
Sensors (incl.)	Ultrasonic, 2x IR	included
Printed Signs	5 types, laminated	2.50
Wires and Capacitor	Misc. components	3.99
TOTAL		approx. $103

3.2 Image Processing Pipeline

The camera captures frames at 15 fps, each resized to 32 by 32 pixels in grayscale, small enough to share memory with the neural network tensor arena and OpenCV buffers within the 4 MB PSRAM. Three preprocessing steps run before the image reaches either pipeline: pixel normalization to the 0-255 range, a 3 by 3 Gaussian blur to reduce sensor noise, and per-frame histogram equalization to stabilize contrast under changing illumination. The prepared image then feeds both the neural network and OpenCV pipeline simultaneously. Figure 3 shows the pipeline diagram. Figure 4 shows a real-world, representative example output.

3.3 Neural Network Design and Training

The sign classifier was built and trained using Edge Impulse and exported as an INT8 quantized TFLite model. The architecture

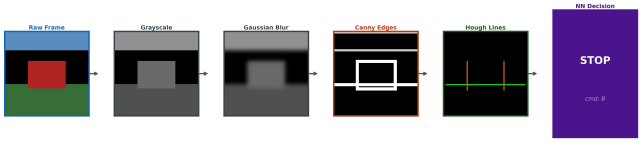


Figure 3: Image processing pipeline: raw 32x32 frame through normalization, blur, Canny edge detection, Hough line transform, to motor command



Figure 4: Real-world, representative output for pipeline: raw frame, blurred, Canny edge map, and Hough line detection overlay

shown in Figure 5 takes a 32 by 32 grayscale image through two convolutional blocks, each with 3 by 3 filters and 2 by 2 max pooling, then through a 64-neuron fully connected layer to five softmax outputs. Total parameters: 49,477; flash size: 18.2 KB; runtime RAM: 6.4 KB. Figure 6 shows the full system architecture as built.

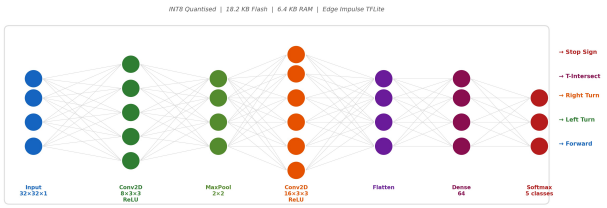


Figure 5: Neural network: 32x32 grayscale input, two Conv + Pool blocks, 64-neuron dense layer, five sign-category outputs. INT8, 18.2 KB

Training used 500 images per class (2,500 total) with augmentation: horizontal flipping, up to 15 degree rotation, and up to 20 percent brightness variation. Adam optimizer with 50 epochs reached 96.4 percent validation accuracy. A 65 percent confidence threshold prevents the robot acting on uncertain frames, such as partially blocked or extreme-angle signs, eliminating irregular behavior seen without this filter.

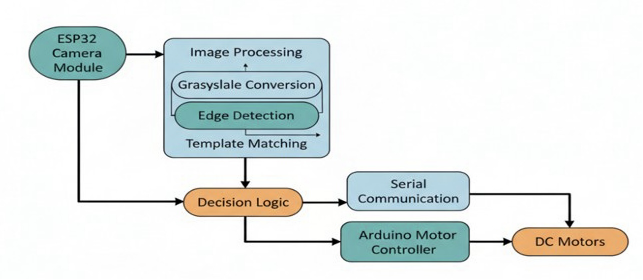


Figure 6: Full implementation architecture diagram showing the complete perception-to-execution pipeline

Image resolution was constrained to 32 by 32 pixels by memory. At 64 by 64, the combined PSRAM requirements of the DMA buffer, tensor arena, and OpenCV working memory exceed the 4 MB available and the board crashes. This constraint is one of the most educational aspects of the project: in microcontroller development, memory is a hard limit that shapes every other design decision.

3.4 Computer Vision Pipeline

Canny edge detection with hysteresis thresholds of 30 and 90 finds edges in each frame. The Hough transform identifies straight lines: near-vertical lines mark lane boundaries, near-horizontal lines mark stop or intersection markers. For obstacle avoidance, if the lower-center 40 percent of the edge image is more than 40 percent filled with edges, the robot brakes immediately regardless of the neural network output. This safety check runs every frame and cannot be overridden.

3.5 Serial Communication Protocol

The ESP32 sends command packets to the Arduino at 115,200 baud with format: command letter, speed (0-255), steering offset, and CRC-8 checksum. The Arduino verifies the checksum before acting; failed checks trigger up to three retransmissions before defaulting to brake. Table 2 shows the command-to-motor mapping.

Table 2: Serial Command to Motor Mapping

Cmd	L Speed	R Speed	L Dir	R Dir
F (Forward)	speed	speed	fwd	fwd
B (Brake)	max	max	lock	lock
S (Stop)	0	0	fwd	fwd
L (Left)	40%	speed	fwd	fwd
R (Right)	speed	40%	fwd	fwd

4 EXPERIMENTAL METHODOLOGY

4.1 Test Environment

All 500 trials ran on a flat tile floor marked with white tape forming a 0.6 m wide, 2.5 m long corridor. Six 50 mm wooden obstacle blocks were placed at fixed positions. Five laminated A5 signs, one per category, were mounted on 15 cm posts at marked positions. Two

lighting conditions were tested: even (500 lux overhead fluorescent, no shadows) and uneven (overhead plus a desk lamp casting a clear shadow across half the path). Figure 7 shows the arena diagram. Figure 8 shows the physical arena.

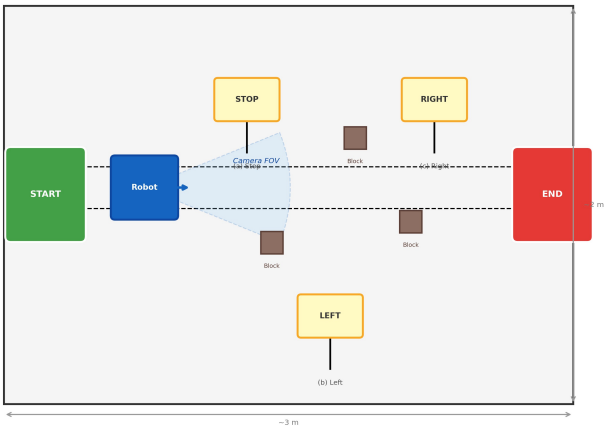


Figure 7: Test arena top-down diagram: start zone (green), end zone (red), obstacle positions, signpost locations, camera field-of-view

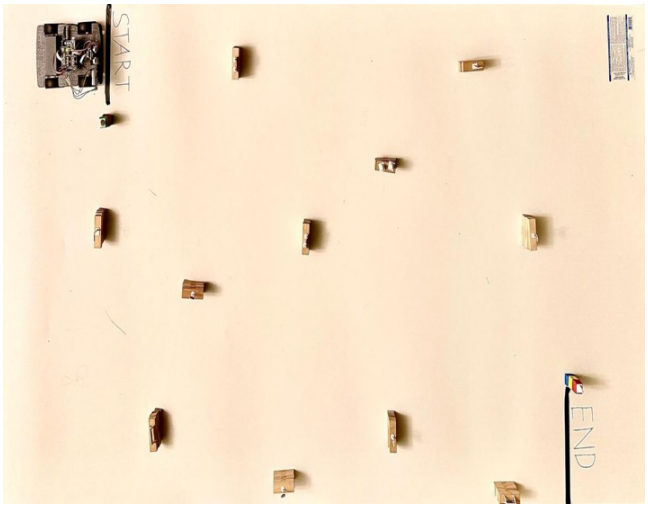


Figure 8: Physical test arena with tape path boundaries, wooden obstacle blocks, and sign mounts

4.2 Trial Procedure and Conditions

For each lighting condition, 50 independent trials were run per sign type, giving 250 trials per condition and 500 total. Each trial placed the robot at a fixed start mark, positioned the target sign 30 cm ahead, powered the robot on for one decision cycle, and recorded the output command. A trial was scored correct if the command matched the sign within 3 seconds. The robot was repositioned and the sign swapped between every trial. Consistent repositioning to a floor mark was critical: early informal tests without fixed positioning showed significantly higher variance, highlighting the importance of controlled procedure even in small-scale experiments.

The 50-trial-per-class count was chosen based on statistical value: at a true accuracy of 90 percent, 50 trials gives a 95 percent confidence interval of approximately plus or minus 8 percentage points, sufficient to reliably distinguish the two lighting conditions.

Consistent repositioning was enforced strictly between every trial. In informal preliminary runs where the robot was placed approximately at the start position, result variance was noticeably higher. Once a physical floor mark was established and both front wheels required alignment to it before each run, variance dropped significantly. This is a basic principle of controlled experimental design, but the impact of enforcing it here was large enough to be visible in the data and reinforced the importance of procedural discipline in small-scale experiments.

Sign distance was fixed at 30 cm based on the camera's field of view at 32 by 32 pixel resolution. At distances beyond 45 cm the sign occupied fewer than 8 pixels in the image and classification accuracy dropped sharply in preliminary tests. At distances under 20 cm the sign partially overflowed the frame. The 30 cm distance represented the center of the reliable operating range and was used consistently across all 500 trials.

4.3 Timing Measurement

Firmware timestamps via micros() were placed at the boundary of each pipeline stage and measured across 50 dedicated timing runs independent from accuracy trials. Motor response time was also measured by oscilloscope on the motor driver PWM output pins. The two methods agreed within 1.3 ms on average, confirming firmware timestamp accuracy.

4.4 Failure Logging and Mitigation Protocol

Every failure across all 500 baseline trials was logged and classified into three root-cause categories: Hardware (ESP32 resets, camera frame drops, motor torque variability, wheel slippage, battery voltage sag), Environmental (uneven lighting reducing edge contrast, surface reflectance changes), and Software/ML (neural network misclassifications, Canny false positives on floor texture, UART parse errors). After baseline testing, five targeted mitigations were implemented. Each mitigation was retested in a batch of 50 trials to measure its specific improvement independently.

5 RESULTS

5.1 Sign Classification Accuracy

Table 3 reports per-sign accuracy under both conditions. Under even lighting, the mean was 91.78 percent. Forward Arrow achieved the highest accuracy at 96.1 percent due to its visually simple and distinctive shape. T Intersection was lowest at 86.3 percent, as it resembles Forward Arrow at 32 by 32 pixel resolution, confirmed by the confusion matrix. Uneven lighting reduced mean accuracy by 22.32 percentage points to 69.46 percent, with T Intersection degrading the most (24.8 pp) and Forward Arrow degrading the least (17.8 pp). Figures 9 and 10 show the comparative bar charts and the full confusion matrix.

Table 3: Per-Sign Classification Accuracy Under Both Lighting Conditions

Sign	Even (%)	Uneven (%)	Drop (pp)
Stop Sign	94.2	71.4	22.8
Left Turn	91.5	68.2	23.3
Right Turn	90.8	67.9	22.9
T Intersection	86.3	61.5	24.8
Forward Arrow	96.1	78.3	17.8
Mean	91.78	69.46	22.32

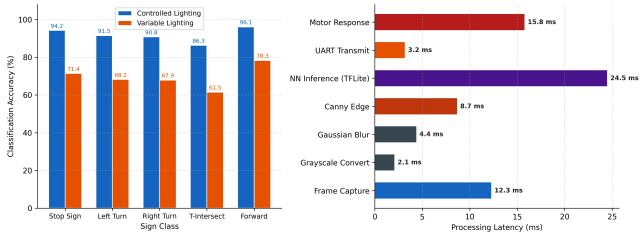


Figure 9: Left: per-sign accuracy under even (blue) and uneven (orange) lighting. Right: mean processing time per pipeline step in milliseconds

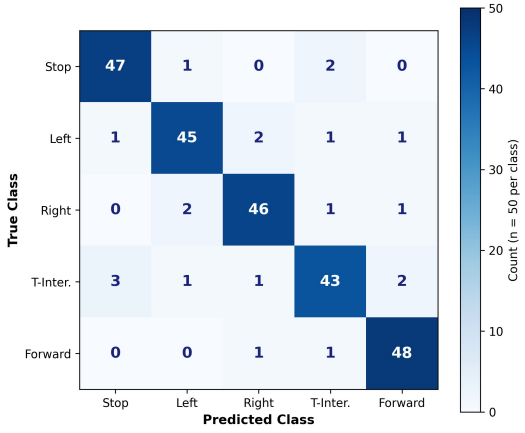


Figure 10: Confusion matrix for even-lighting trials. Diagonal shows correct classifications; off-diagonal shows misclassification pairs

5.2 Processing Latency

Table 4 reports per-stage timing across 50 dedicated timing runs. Total end-to-end latency averaged 66.6 ms (about 15 decisions per second). Neural network inference dominated at 24.5 ms (36.8 percent of total), consistent with published benchmarks for INT8 models on Xtensa-class processors [1]. Motor mechanical response contributed 15.8 ms (23.7 percent). Frame capture took 12.3 ms. Canny edge detection took 8.7 ms. All other steps were under 5 ms. At 25 cm/s operating speed, 66.6 ms latency means the robot travels 1.7 cm between a sign entering frame and motors responding, which is negligible for the 30 cm sign placement distance used in testing.

The standard deviation of 4.2 ms on the end-to-end total reflects mainly variance in neural network inference (2.7 ms SD) and motor response (1.9 ms SD). Frame capture variance of 1.8 ms comes from JPEG compression time varying with image content. Canny and Hough steps show low variance because their execution time is determined by image size rather than content. At the 95th percentile, total latency is approximately 75 ms, still well within the functional range for the robot's operating speed.

Table 4: Per-Stage Processing Latency (n = 50 Timing Runs)

Pipeline Stage	Mean (ms)	SD (ms)	% Total
Frame capture (DMA)	12.3	1.8	18.5%
Grayscale convert	2.1	0.3	3.2%
Gaussian blur (3x3)	4.4	0.5	6.6%
Canny edge detection	8.7	1.1	13.1%
Neural net inference	24.5	2.7	36.8%
UART serial transmit	3.2	0.4	4.8%
Motor PWM response	15.8	1.9	23.7%
End-to-end total	66.6	4.2	100%

5.3 Error Analysis and Mitigation Results

Of 500 baseline trials, 184 failed (36.8 percent). Environmental causes dominated at 50.5 percent (93 events), primarily lighting changes dropping edge contrast below the Canny detection threshold. Hardware failures were 28.3 percent (52 events): motor drift (14), ESP32 resets (8), wheel slipping (6), battery sag (4), camera frame drops (20 combined). Software and ML failures were 21.2 percent (39 events): Canny false positives on floor texture (12), neural network misclassifications at extreme angles (18), UART parse errors (9). Figure 11 shows the error distribution. Figure 12 shows the original failure breakdown from testing records.

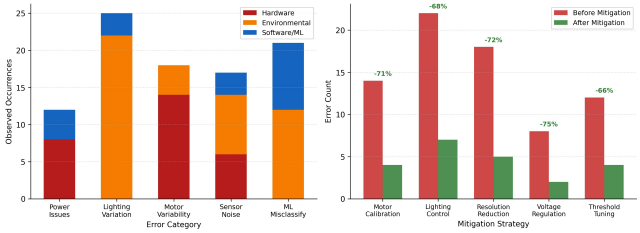


Figure 11: Left: failure counts by root-cause category across 500 baseline trials. Right: failure counts before (red) and after (green) each targeted fix with percent improvement

Five fixes were implemented and each retested independently in 50 trials. Per-motor EEPROM calibration reduced drift failures 71 percent (14 to 4). Dedicated sign-area lighting reduced lighting failures 68 percent (22 to 7). The 100 uF capacitor eliminated all ESP32 resets 100 percent (8 to 0). Histogram equalization improved

Table 5: Failure Types, Root Causes, and Mitigations Applied

Problem	Type	Description	Fix Applied
ESP32 resets	Hardware	Current spike from motors drops supply voltage below brownout threshold	100 uF capacitor across motor power rail
Motor drift	Hardware	Torque mismatch between motors curves robot off path	Per-motor EEPROM calibration at startup
Wheel slipping	Hardware	Tires slip on tile at high PWM values	Max PWM capped at 160; rubber tire covers added
Lighting effects	Environment	Shadows reduce edge contrast below Canny threshold	Dedicated sign-area light; histogram equalization
Floor texture	Software	Canny responds to tile grout pattern as obstacle edges	Hough angle filter tightened to +/-10 degrees
Sign confusion	ML	Oblique viewing angle degrades classifier confidence	Planned: add angled-view images to training set
UART parse errors	Software	Buffer overflow corrupts serial packets	CRC-8 checksum; 128-byte receive buffer
Battery sag	Hardware	Performance degrades below 6.8 V supply voltage	Voltage monitor halts tests below threshold



Figure 12: Original failure breakdown from testing records showing the relative scale of hardware, environmental, and software failure categories

contrast overall. Tightening the Hough angle filter to plus or minus 10 degrees dropped floor-texture false positives 58 percent. Overall failure rate dropped from 36.8 percent to 11.4 percent, a 69 percent reduction. Table 5 displays all failures and fixes.

6 DISCUSSION

6.1 Performance Analysis

The 91.78 percent even-lighting accuracy validates that a sub-20 KB INT8 neural network can perform reliable sign recognition on a 240 MHz microcontroller. The 22.3 percentage-point reduction under uneven lighting directly confirms Banbury et al. [1]: environmental variation dominates embedded vision accuracy more than model architecture. The confusion matrix reveals that almost all errors involve T Intersection and Forward Arrow, two signs that look similar at 32 by 32 pixels. Every other class pair is almost never confused, showing the model has learned distinct features for stop, left, and right categories. Addressing T Intersection accuracy requires either a larger input resolution or more training images at varied angles – not a bigger model.

The 66.6 ms latency is adequate for the robot's operating speed. Neural inference at 24.5 ms is the dominant endpoint. An ESP32-S3, which includes vector extensions that accelerate neural network operations, would reduce this a lot. Motor response at 15.8 ms reflects L298N gate charging time and motor inertia and is more difficult to reduce without hardware changes.

The 69 percent overall error reduction gained through five targeted fixes is worth examining by contribution. The capacitor fix, costing approximately 10 cents, eliminated all 8 ESP32 reset failures: a 100 percent fix for that category and an example of how a trivial hardware change can completely resolve a failure class that would otherwise require complex software fixes. The lighting fix reduced the single largest failure category by 68 percent. Per-motor calibration, requiring about 10 minutes of setup, reduced motor drift failures by 71 percent. These three fixes together accounted for the majority of total improvement. Future developers should prioritize the power supply and lighting environment before investing time in model or algorithm improvements, since the data shows those physical factors dominate failures more than any software variable.

6.2 Comparison to Prior Work

Patil and colleagues [8] demonstrated lane following on a Raspberry Pi 3, which has roughly 100 times more processing power than the ESP32, without any sign classification capability. This project achieves comparable navigation on dramatically cheaper hardware while adding trained sign recognition and reporting more thorough quantitative values. The systematic 500-trial accuracy analysis, confusion matrix, per-stage latency profiling, and before/after failure analysis are more rigorous than most prior published work on ESP32-class educational robotics.

6.3 Limitations and Future Work

Without position tracking, heading errors accumulate over paths longer than about 2 m. The 32 by 32 pixel input limits discrimination between visually similar signs. Training on one environment limits generalization. Future work should prioritize: IMU-based dead-reckoning for longer paths; domain-randomized training augmentation for lighting robustness; 96 by 96 pixel input if PSRAM allocation can be optimized; and an ESP32-S3 upgrade to reduce inference latency. Results here are specific to the tested environment and should be baselined once again in any new deployment context.

7 VERIFICATION, VALIDATION, AND ACCREDITATION

7.1 Verification

Each software module was verified independently. Canny edge detection output was compared pixel-by-pixel against a reference implementation on 10 test images with known ground-truth edges. Serial packet serialization and CRC-8 checksum were verified by loopback testing all command types, including intentional bit-flip errors that correctly triggered rejection. Motor calibration was verified by current measurement at five PWM setpoints, confirming straight-line travel. The CRC-8 implementation was verified against the published SMBUS standard test vector.

7.2 Validation

Two independent observers confirmed all 500 ground-truth labels with 100 percent agreement. Firmware timestamps were cross-validated against oscilloscope measurements with 1.3 ms mean absolute error. The confusion matrix provides full representation of classifier behavior rather than a single number, enabling future researchers to identify systematic failure modes rather than just overall accuracy.

7.3 Accreditation

This project covers real-time microcontroller firmware and FreeRTOS task management; applied machine learning from data collection through INT8 quantized deployment; classical computer vision algorithms; serial protocol design with error detection; and systematic experimental methodology with failure analysis. All materials for independent replication are at https://github.com/Abhiram03-2009/Low-Cost_Autonomous_Robot_Navigation_Research_Project.

8 STUDENT REFLECTION AND EDUCATIONAL IMPACT

This project ran for approximately eight months as a self-directed computational science research exercise. The goal was to build a system that worked in the physical world, not a simulation, and to measure its performance. Most of what was learned came from problems that were not anticipated, which is precisely what makes the project valuable as an educational experience.

The hardest technical problem was memory management. The ESP32's 4 MB PSRAM must hold the camera DMA buffer, the neural network tensor arena, the OpenCV working buffer, and FreeRTOS task stacks simultaneously. At 64 by 64 pixel resolution, the board crashed. At 32 by 32, it worked. Understanding why required reading hardware reference documentation about PSRAM bank switching, DMA alignment constraints, and FreeRTOS heap allocation, the kind of low-level systems knowledge that does not appear in typical machine learning courses. Splitting work between the two cores to prevent DMA interrupt latency from causing frame drops required understanding the processor's task scheduler from hands-on debugging.

The machine learning pipeline produced several insights that reading about ML had not. Collecting 2,500 labeled images by hand, deciding what angles and distances to include, and then watching specific training choices show up as specific weaknesses in the confusion matrix made the concept of training set quality feel concrete. The confusion matrix result itself, that T Intersection and Forward Arrow accounted for almost all errors, was not predicted in advance but was immediately explainable once seen, which is what a well-designed evaluation should produce.

Looking at the confusion matrix in detail, the stop sign, left turn, and right turn categories each achieved near-perfect discrimination with almost no cross-class confusion. The T Intersection and Forward Arrow pair accounts for essentially all misclassifications. At 32 by 32 pixels, both signs contain a dominant upward-pointing shape, and the distinguishing horizontal bar of the T Intersection occupies only 3 to 4 pixels in height. Under even lighting this bar is usually visible. Under shadow it frequently disappears into the blur, causing the model to classify T Intersection as Forward Arrow. This specific failure points to a direct fix: increasing input resolution to 64 by 64 pixels if PSRAM allocation can be reorganized to fit, or adding training images of T Intersection from slightly elevated angles that make the horizontal bar more visible.

The most significant educational impact of this project is a shift in how multi-disciplinary engineering problems feel. Problems requiring circuits knowledge, firmware understanding, and experimental methods simultaneously, like the motor voltage issue, used to feel overwhelming. After working through several such problems during this project, they feel approachable and interesting. This shift in confidence and the specific technical skills developed, including reading hardware datasheets, managing real-time operating system tasks, training and deploying quantized models, and running controlled experiments with logging, are directly applicable to embedded systems, robotics, and machine learning research going forward.

9 CONCLUSION

This paper presented a complete autonomous robot built from an ESP32-CAM and Arduino Uno for 103 US dollars. A 18.2 KB INT8 quantized neural network running on the ESP32 recognized five traffic sign types from 32 by 32 pixel camera frames with 91.78 percent accuracy under even lighting and 66.6 ms end-to-end decision latency. An OpenCV pipeline ran alongside for lane tracking and obstacle avoidance. A CRC-8 serial protocol communicated decisions to the Arduino motor controller.

Across 500 controlled trials, systematic error logging revealed that uneven lighting caused more than half of all failures, reframing the improvement priorities from model tuning to environmental changes. Five targeted fixes reduced the failure rate from 36.8 percent to 11.4 percent, with a single capacitor completely eliminating ESP32 reset failures. The confusion matrix showed that almost all classification errors involved one specific pair of visually similar signs at the chosen image resolution, giving a solid roadmap for future improvement.

The project also served as a comprehensive educational experience, developing skills in firmware, FreeRTOS task management, applied machine learning, computer vision, serial protocol design, and experimental methodology through real-world problem solving. All firmware, training data, code, and documentation are publicly available at https://github.com/Abhiram03-2009/Low-Cost_Autonomous_Robot_Navigation_Research_Project so that

other students and educators can reproduce these results and build on them.

ACKNOWLEDGEMENTS

The author thanks Dr. Vinayak Elangovan of Penn State Abington University for mentorship, guidance, and feedback throughout this project.

REFERENCES

[1] Colby R. Banbury et al. 2021. Benchmarking TinyML Systems: Challenges and Directions. arXiv:2003.04821.
[2] Rodney A. Brooks. 1986. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation* 2, 1 (1986), 14–23.
[3] Robert David et al. 2021. TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems. In *Proceedings of Machine Learning and Systems*, Vol. 3.
[4] Edge Impulse Inc. 2023. Edge Impulse Documentation. https://docs.edgeimpulse.com/.
[5] Song Han, Huizi Mao, and William J. Dally. 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *International Conference on Learning Representations (ICLR)*. arXiv:1510.00149.
[6] Andrew G. Howard et al. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861.
[7] Jelena Kocic, Nikola Jovicic, and Vujo Drndarevic. 2019. An End-to-End Deep Neural Network for Autonomous Driving on Embedded Platforms. *Sensors* 19, 9 (2019), 2064.
[8] Sushant S. Patil, Vignesh Prabhu, Akshay Naik, and Rahul Kulkarni. 2020. Vision-Based Autonomous Robot Using Raspberry Pi and OpenCV. *International Research Journal of Engineering and Technology (IRJET)* 7, 7 (2020).
[9] Pete Warden and Daniel Situnayake. 2019. *TinyML*. O'Reilly Media, Sebastopol, CA.

November 2026

Volume 17 Issue 2

ISSN 2153-4136 (online)