Beyond Buzzwords: Making Sustainability a Pillar of the Computing Curriculum

Nikhila Alavala
nalavala6@gatech.edu
Georgia Institute of Technology
Atlanta, USA

Nimisha Roy
nroy9@gatech.edu
Georgia Institute of Technology
Atlanta, USA

Melinda McDaniel
mcdaniel@cc.gatech.edu
Georgia Institute of Technology
Atlanta, USA

Max Mahdi Roozbahani
mahdir@gatech.edu
Georgia Institute of Technology
Atlanta, USA

Rodrigo Borela
rborelav@gatech.edu
Georgia Institute of Technology
Atlanta, USA

Parisa Babolhavaeji
pbabolhavaeji3@gatech.edu
Georgia Institute of Technology
Atlanta, USA

Abstract

The rapid digitalization of the global economy, driven by big data and artificial intelligence, has significantly increased energy consumption, reshaped labor markets, and impacted politics and communities to an extraordinary extent. Addressing these challenges involves educating future computer scientists about the carbon emissions associated with their code and the broader societal consequences of the technologies they design. Traditionally, computing education has focused on optimizing runtime and memory efficiency, frequently overlooking the links to energy efficiency and carbon footprint considerations. Additionally, the integration of ethics into the curriculum has not been comprehensive. This paper proposes a framework for integrating sustainability into the computing curriculum, prioritizing it as a critical consideration for students. It outlines the competencies required for sustainability education and identifies topics directly related to the UN SDGs as a natural entry point for sustainability concepts. Additionally, it reports on a pilot framework implementation at a major US public university, where over 3,200 students from over 30 disciplines were exposed to sustainable coding practices and the ethics of AI and machine learning. The curriculum incorporated transformative teaching and learning methodologies with lectures, supplemental materials, and interactive projects highlighting these themes. Challenges to implementation, which may be encountered by other institutions, are also discussed. Survey results demonstrate that sustainability can be seamlessly integrated into early coding education, encouraging ongoing effort to accentuate energy-efficient coding in computer science courses.

CCS Concepts

• **Social and professional topics → Model curricula**; **Computer science education**; **Sustainability**.

Keywords

Sustainability education, CS1, capstone courses, machine learning

ITiCSE 2025, Nijmegen, Netherlands

ACM ISBN 979-8-4007-1567-9/2025/06
https://doi.org/10.1145/3724363.3729034

ACM Reference Format:
Nikhila Alavala, Nimisha Roy, Melinda McDaniel, Max Mahdi Roozbahani, Rodrigo Borela, and Parisa Babolhavaeji. 2025. Beyond Buzzwords: Making Sustainability a Pillar of the Computing Curriculum. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2025), June 27-July 2, 2025, Nijmegen, Netherlands.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3724363.3729034

1 Introduction

The rapid growth of digital infrastructures, including data centers, cryptocurrencies, and AI, has significantly increased global electricity consumption. In 2022, these sectors accounted for nearly 2% of the world's electricity [11], and projections suggest a sharp rise by 2026, potentially matching the electricity demand of a country like Sweden [12]. AI, particularly generative models, is a major driver of this trend, with some systems consuming as much energy during training as hundreds of households [15].

Beyond their environmental impact, AI-driven technologies bring about far-reaching societal and economic pressures [28]. They can amplify biases embedded in their training data [25] facilitate cybercrimes [8], influence political discourse [13] and reshape labor markets [5]. Given the rapid pace at which these digital infrastructures evolve, the question is no longer whether to integrate sustainability into the computing curriculum, but how.

Simultaneously, the United Nations introduced 17 Sustainable Development Goals (SDGs) [29] to tackle urgent global challenges such as poverty, environmental degradation, and inequality. These SDGs fall into three overarching categories, People (SDGs 1, 2, 3, 4, 5, 10, 16), Planet (SDGs 6, 7, 12, 13, 14, 15), and Prosperity (SDGs 8, 9, 11, 17), underscoring the broad spectrum of sustainability. Among them, SDG 4.7 highlights the critical importance of embedding sustainability within educational systems to cultivate responsible global citizens [29]. As higher education institutions mold future leaders and innovators, they occupy a pivotal role in guiding the sustainable development of computing technologies [30].

This paper argues that integrating the UN Sustainable Development Goals (SDGs) into the computing curriculum is critical for cultivating ethically grounded and environmentally conscious computing professionals. Beyond acquiring technical expertise, students must understand the societal and environmental implications of their skills. By bridging the technical and ethical dimensions, educators can empower students to design solutions that not only perform efficiently but also contribute meaningfully to societal progress and environmental sustainability. Leveraging transformative teaching and learning frameworks [27], we propose a holistic approach to embedding SDG-aligned topics within the computer science curriculum, as outlined in CSED 2023 [14].

To illustrate the feasibility of this integration, we present findings from an initial implementation at a large public university in the United States. In this pilot program, course content, lectures, assignments, and student projects were restructured to incorporate SDG-aligned learning outcomes. By Fall 2024, the program reached over 3,200 students across 30 majors, offering valuable insights into the practicalities of embedding sustainability in computing education on a broad and scalable level. Finally, we explore implementation challenges, propose future directions for integrating sustainability in education, and advocate for the widespread adoption of SDG-focused curricula to prepare students for the pressing global challenges of the 21st century.

2 Sustainability in Computing Education

Despite a growing emphasis on sustainability in many higher education institutions, student awareness of the UN SDGs remains remarkably low [31]. A central reason for this gap is the insufficient integration of SDG-related content into formal curricula. For example, a study at TU Dublin revealed that SDGs 10–14 were seldom addressed, and SDG 12 (Responsible Consumption and Production) appeared explicitly in only 12 courses [31]. Similarly, a different survey showed that although many subject coordinators recognized the importance of SDGs, they often did not incorporate these links into course materials [22]. Notably, the College of Computing had the lowest integration rate, with just 29% of its subjects referencing at least one SDG [22].

The lack of integration is particularly evident in computing education, where sustainability principles are frequently overlooked. Mishra [17] observed that many programs still lack comprehensive modules on Sustainable Software Engineering (SSE) and emphasized the need for practical assignments and projects that allow students to apply these principles in real-world contexts. Similarly, Manotas et al. [16] found that software professionals from companies such as Google and Microsoft reported a lack of preparedness to address sustainability, highlighting the shortcomings of current educational practices. In addition, a survey conducted by Saraiva et al. [24] revealed that 14 out of 21 researchers and educators rated the availability of teaching materials for green software courses as inadequate. Finally, barriers to implementation, such as lack of interest, staff training, and tradition, have also been noted [3], even when sustainability is viewed as a means of revitalizing computing education and attracting more diverse students [3].

Despite the gaps in SDG integration, several successful programs illustrate the potential for embedding sustainability into computing education to enhance student engagement and awareness. A notable initiative is a multi-university European partnership, which developed a structured framework for teaching sustainability in computing [21]. This collaboration emphasized collaborative curriculum design, project-based learning, and real-world challenges, enabling students to apply sustainability principles within computing contexts.

Research on integrating sustainability into computing education has identified several strategies, including the development of new green computing courses, modular integration into existing courses, adopting an approach where sustainability is prioritized throughout course design [3], and reimagining the structure and purpose of computing education [6]. However, studies suggest that most revisions remain smaller, incremental updates rather than comprehensive overhauls [20].

Saraiva et al. [24] advocate for introducing sustainability concepts both "early" and "often." By "early," they emphasize starting at the CS1 or CS2 programming levels, and by "often," they recommend integrating sustainability across upper-level undergraduate courses, such as data structures, software engineering, and graduate-level topics like machine learning. Their findings also demonstrate the tangible impact of such interventions: small changes in teaching sustainable coding practices increased the proportion of students writing energy-efficient code (under 10 joules) from 45% to 81%. This highlights the effectiveness of embedding sustainability concepts even at the introductory level.

Similarly, Erkan [7] shows that incorporating sustainability into traditional computer science topics, such as data structures and algorithms, can boost student engagement by connecting technical concepts to real-world environmental challenges, such as power grid resilience and sustainable resource management. Embedding sustainability directly into the core curriculum not only enhances technical learning but also highlights the relevance of computing skills in addressing critical global issues like climate change and resource consumption.

These examples underscore the growing recognition that the integration of sustainability, whether through collaborative programs, active learning, or targeted interventions, can significantly improve student learning outcomes. However, scaling these successes into broader curricular reforms remains a significant challenge, as discussed in the following section.

3 Aligning CompEd with the UN SDGs

3.1 Pedagogical Strategy

The gaps in integrating sustainability into computing education, as highlighted in the preceding section, call for a structured and intentional approach to embedding SDG-related content throughout the curriculum. Research has demonstrated that sustainability concepts, when introduced systematically, not only enhance student engagement but also improve their technical competencies and preparedness for addressing global challenges [17, 20, 22]. Building on these findings, we propose a pedagogical framework that emphasizes progressive and integrated learning.

This framework draws on successful strategies identified in prior studies, such as frequent integration, active learning methodologies, and sustainability-focused projects. These insights provide the foundation for our proposed strategy.

Our framework follows a progressive approach, embedding SDG content from introductory programming courses through advanced capstone projects. By aligning content with students' evolving technical expertise, the curriculum ensures a deepening understanding of sustainability concepts. Moreover, real-world, SDG-focused projects empower learners to develop computing expertise while addressing critical global issues and nurture their commitment to advancing these goals. The implementation strategy is built around these three core elements:

(1) Frequent and distributed integration: Embedding SDG-related content multiple times within courses and across a program ensures consistent reinforcement and builds a cumulative understanding of sustainability themes.
(2) Active and experiential Learning: Incorporating methodologies like PBL [20] and PjBL [22] immerses students in problem-solving scenarios that directly link computing concepts to sustainability applications.
(3) SDG-Themed Projects: Designing assignments and projects around sustainability challenges illustrates how computing can drive solutions for global goals.

To unify these elements, we adopt the transformational teaching framework by Slavich and Zimbardo [27], positioning instructors as intellectual coaches who foster collaboration, shared goals, and meaningful engagement. This approach emphasizes establishing a shared vision by clearly integrating sustainability themes with core computing objectives, providing hands-on, SDG-aligned projects to build technical and sustainability-related skills, and challenging students' critical thinking and technical abilities in a supportive environment. It also includes offering personalized guidance to help students understand their potential impact on sustainability challenges, creating experiential lessons that connect theoretical knowledge with practical application through fieldwork, community engagement, or industry collaborations, and promoting iterative learning through reflection and preflection, encouraging students to assess their assumptions and evaluate project outcomes.

As the next step, it is essential to connect this pedagogical approach with the specific knowledge areas outlined in CSED2023 [14].

3.2 Mapping the Curriculum to the UN SDGs

By mapping each knowledge area to relevant SDGs, we can create a comprehensive blueprint for curriculum design that integrates sustainability themes across the breadth of computer science education. In this section, we outline how the knowledge areas and learning outcomes defined in CSED2023 [14] align with the UN SDGs. By connecting existing outcomes to specific SDGs and proposing targeted enhancements where necessary, we aim to position sustainability as a foundational element of computer science education, ensuring it is deeply embedded rather than treated as a peripheral consideration.

3.2.1 Artificial Intelligence (AI). Apply machine learning and knowledge representation techniques to optimize energy usage and mitigate climate change (SDG 12, 13). Assess AI models for fairness, accountability, and transparency, ensuring equitable access and responsible innovation (SDG 5, 9, 10, 16).

3.2.2 Algorithms and Complexity (AL). Develop and analyze algorithmic solutions that minimize computational overhead and resource consumption (SDG 12, 13). Adapt fundamental algorithmic paradigms (e.g., graph algorithms, greedy strategies) to address infrastructure needs in transportation and city planning (SDG 9, 11).

Table 1: The 17 UN Sustainable Development Goals [29]

Goal Number	Goal Name
1	No Poverty
2	Zero Hunger
3	Good Health and Well-being
4	Quality Education
5	Gender Equality
6	Clean Water and Sanitation
7	Affordable and Clean Energy
8	Decent Work and Economic Growth
9	Industry, Innovation, and Infrastructure
10	Reduced Inequalities
11	Sustainable Cities and Communities
12	Responsible Consumption and Production
13	Climate Action
14	Life Below Water
15	Life on Land
16	Peace, Justice, and Strong Institutions
17	Partnerships for the Goals

3.2.3 Architecture and Organization (AR). Optimize hardware performance by designing and evaluating CPU/GPU architectures for power efficiency and reduced e-waste (SDG 12, 13). Implement parallelism and pipelining techniques to improve system throughput while maintaining sustainability and resilience (SDG 9).

3.2.4 Data Management (DM). Design secure and compliant data storage systems that respect user privacy, enhance data security, and minimize ethical risks (SDG 10, 16). Employ lifecycle and archival strategies (e.g., compression, deduplication) to reduce storage overhead and energy consumption (SDG 9, 12).

3.2.5 Foundations of Programming Languages (PL). Compare language paradigms and abstractions for efficient use of memory and computing resources (SDG 12, 13). Implement energy-aware programming constructs (e.g., lazy evaluation, efficient data types) that lower runtime costs (SDG 12).

3.2.6 Graphics and Interactive Techniques (GIT). Develop high-performance rendering pipelines that minimize energy usage and hardware requirements (SDG 12, 13). Incorporate ethical design principles when creating and manipulating visual media (e.g., deepfakes, AR/VR) to ensure responsible deployment (SDG 10, 16).

3.2.7 Human-Computer Interaction (HCI). Design accessible and inclusive interfaces that accommodate diverse user needs, reducing digital inequalities (SDG 10). Apply user-centered research methods (e.g., usability testing, ethnographic studies) to maintain privacy, trust, and ethical standards (SDG 16).

3.2.8 Networking and Communication (NC). Configure and manage network infrastructure for efficient data transmission and minimal energy usage (SDG 7, 12). Implement connectivity solutions that expand access to remote and underserved communities, bridging digital divides (SDG 10).

3.2.9 Operating Systems (OS) & Systems Fundamentals (SF). Use advanced scheduling and frequency scaling to lower power consumption and balance system performance (SDG 12, 13). Architect core system services (e.g., memory management, I/O handling) for reliability and sustainability (SDG 9).

3.2.10 Parallel and Distributed Programming (PD). Employ concurrency and synchronization techniques to optimize workload distribution and reduce energy overhead (SDG 12, 13). Design resilient distributed systems (e.g., fault-tolerant clusters) that support sustainable infrastructure and global collaboration (SDG 9, 11).

3.2.11 Software Development Fundamentals (SDF). Select and evaluate efficient data structures and algorithms for performance and lower resource consumption (SDG 12, 13). Adopt maintainable coding standards (e.g., modular design, clean interfaces) to reduce software bloat and extend product lifecycles (SDG 9, 10).

3.2.12 Software Engineering (SE). Integrate green software practices (e.g., profiling, refactoring) into the development lifecycle for long-term resource efficiency (SDG 9, 12, 13). Coordinate multi-person, multi-version projects with robust communication, testing, and ethical review processes (SDG 8, 10, 16).

3.2.13 Security (SEC). Implement secure design principles (e.g., zero-trust, defense-in-depth) to protect critical infrastructure and maintain public trust (SDG 9, 11). Address adversarial threats and vulnerabilities by integrating risk assessment and human-centered security measures (SDG 16).

3.2.14 Social Issues and Professional Practice (SP). Embed sustainability ethics (people, planet, prosperity) into computing projects, assessing social and ecological impacts (SDG 4, 10, 12, 13). Engage in reflective practice and policy discussions regarding equal access, inclusivity, and long-term effects of tech on society (SDG 10).

4 Pilot Implementation

To validate the proposed framework and assess its effectiveness, we conducted a year-long pilot implementation spanning two academic terms at a large public university in the United States. The pilot was integrated in six courses representing different stages of the curriculum, providing a comprehensive perspective on its applicability and impact. Detailed descriptions of these implementations are included to illustrate practical aspects for integrating the framework into diverse course contexts. Student reactions to the curriculum were evaluated using survey instruments, offering insights into their perceptions and the framework's effectiveness. The evaluation is ongoing and future publications will provide a more detailed analysis of the framework's impact within individual courses, along with a comprehensive description of the survey instruments used.

4.1 Intro to Computing and Data Manipulation

4.1.1 Overview. Our initiative redesigned the CS1 (Introduction to Computing) and CS1.5 (Data Manipulation) courses, categorized under the Algorithms and Complexity knowledge area per the CSED2023 classification [14]. The redesign focused on incorporating SDGs 13 (Climate Action) and 12 (Responsible Consumption and Production), aiming to achieve the following sustainability-aligned learning outcomes:

(1) Recognize the environmental impact of code execution.
(2) Calculate the carbon footprint of large-scale code execution.
(3) Evaluate code efficiency through the lens of energy consumption.
(4) Implement strategies for writing energy-efficient code.

4.1.2 Theoretical Foundation. Lectures and supplemental materials emphasized energy efficiency by exploring algorithmic efficiency and the performance implications of data structures, such as comparing list and dictionary lookups. Students learned to avoid redundant computations (e.g., repeated list sorting within loops) and explored code refactoring strategies through computational complexity analysis and trade-off evaluations.

4.1.3 Hands-on Assigments. Practical assignments encouraged students to apply these theoretical principles by assessing the energy consumption and carbon footprint of their code. Using tools like CodeCarbon [4], students quantified energy usage and optimized algorithms to reduce environmental impact. These assignments incorporated sustainability themes, such as analyzing datasets of carbon emissions from commercial and private flights. This contextualization encouraged students to adopt energy-efficient coding practices while reflecting on how computing can address broader environmental challenges and the role of systems contributing to environmental degradation.

4.1.4 Outcomes and Impact. The curriculum redesign reached approximately 1,100 students in CS1 and 300 students in CS1.5 over two terms in 2024.

To evaluate the impact of the redesign, students completed a 5-minute survey measuring their perceptions of sustainability in computing. The survey consisted of 13 items: three demographic questions, two free-response questions, and eight Likert-scale items. Pre-surveys were administered during the first month of instruction, and post-surveys were conducted in the final weeks to track changes in student perceptions.

A total of 58 students completed the pre-survey, and 32 completed the post-survey. Only fully completed responses were included in the analysis. Given the small sample size and non-normal distribution, a two-tailed Fisher's Exact Test was used to analyze the data, consolidating responses into a 2x2 contingency table (e.g., ratings 1–2 vs. 3–5).

The key findings from the curriculum redesign indicate significant improvements in students' confidence and awareness of sustainability in computing. Confidence in energy-efficient coding increased notably, with mean scores rising from 1.44 to 2.15 ($p = 0.001$). Additionally, the percentage of students rating the course as highly effective in teaching energy-efficient coding more than doubled, increasing from 14% to 33%. Furthermore, the proportion of students who strongly agreed that programmers have a responsibility to write energy-efficient code rose by 30%. In addition to self-efficacy gains, the curriculum redesign also yielded measurable improvements in students' ability to write energy-efficient code. An assignment asked students to implement data queries using both list-of-tuples and dictionary-based approaches. With dictionary implementations reducing search complexity from $O(n)$

to $O(1)$, students achieved emissions ratios ranging from 80 to 120, indicating their optimized code produced up to 120 times fewer carbon emissions than the less efficient versions. These results highlight the effectiveness of the redesign in promoting both practical skills and a deeper understanding of sustainability's relevance to computing.

4.2 Software Engineering and Capstone Project

4.2.1 Overview. The redesign of CS 3.0 (Software Engineering), CS 3.1, and CS 3.2 (Capstone Sequence) aimed to integrate sustainability principles into the curriculum, preparing students to develop environmentally responsible and socially aware software systems. These courses align with the Software Engineering knowledge area as defined by CSED2023 [14]. The learning outcomes and content were mapped to SDGs 10 (Reduced Inequalities), 11 (Sustainable Cities and Communities), 12 (Responsible Consumption and Production), and 13 (Climate Action). Building on the foundational knowledge from CS1, the redesigned curriculum focused on the following objectives:

(1) Evaluate the environmental and social implications of software development, including energy consumption, e-waste, and human impact.
(2) Integrate broad social and environmental considerations into the requirements engineering phase of software development.
(3) Implement and evaluate software systems with an emphasis on efficient use of cloud resources, data storage, and green technologies.

4.2.2 Theoretical Foundation: Lectures covered key topics such as sustainability in software engineering (SE), the SDGs, and Sustainable Software Engineering (SSE) principles. Drawing on foundational frameworks by Dick et al. [10] and Brooks et al. [2], the curriculum emphasized minimizing software's societal, economic, and environmental impacts while promoting eco-goals like eco-efficiency and eco-collaboration. Practical techniques such as SPELL (Spectrum-based Energy Leak Localization) [19] were introduced to help students identify and address energy inefficiencies in their code, connecting theory with actionable practices.

4.2.3 Hands-On Assignments: Assignments ranged from analyzing energy consumption in coding practices to evaluating project requirements through a sustainability lens. Students implemented energy-efficient code, created sustainability analysis diagrams addressing environmental, social, and technical dimensions [2], and redesigned software architectures and UML diagrams with sustainability as a key consideration. These activities encouraged critical evaluation and optimization, fostering a practical understanding of sustainability in diverse SE scenarios.

4.2.4 Outcomes and Impact. The redesigned curriculum was implemented in these courses with a combined enrollment of approximately 300 students. Unlike CS1 and CS1.5, where pre- and post-surveys were used, the CS3.X courses utilized a control group (prior to SDG-aligned content) and an intervention group (with the redesigned curriculum). Survey data compared results between CS 3.1 control and intervention groups, as well as between CS 3.0 and CS 3.1 intervention groups (data for CS 3.2 was unavailable).

The intervention group demonstrated superior performance across all "General Outlook" metrics, with $n = 38$ responses in both groups. Key findings included:

- GO1: Enhanced understanding of sustainability in the early stages of software development.
- GO2: Improved recognition of sustainable technical practices.
- GO3: Increased agreement on the importance of ethical considerations in SE.
- GO4: Greater awareness of the impact of coding practices on sustainability.

The intervention group consistently reported higher confidence levels across all self-assessment metrics:

- SA1: Increased awareness of sustainability concepts.
- SA2: Greater familiarity with sustainability tools and techniques.
- SA3: Improved understanding of the UN SDGs, validating the new lecture modules' effectiveness.
- SA4: Enhanced preparedness to integrate sustainability into professional practice.

While both groups recognized the importance of sustainability, with "General Outlook" scores averaging above 3.0, the control group scored below 3.0 in "Self-Assessment," indicating lower confidence in applying sustainability concepts. In contrast, the intervention group showed significant gains, particularly in student awareness (SA1), highlighting the success of the redesigned curriculum in enhancing both students' skills and their understanding of sustainability's role in software development.

4.3 Machine Learning

4.3.1 Overview. The redesign of the CS 4.0 and CS 7.0 courses (Machine Learning at undergraduate and graduate levels) aimed to equip students with technical expertise while fostering an understanding of the ethical, environmental, and societal implications of AI. These updates addressed the common omission of sustainability and ethics in machine learning (ML) curricula [23], despite their critical importance [9, 18, 26]. The revised curriculum aligned with SDGs 3 (Good Health and Well-being), 5 (Gender Equality), and 10 (Reduced Inequalities), introducing the following learning outcomes:

(1) Understand sustainability issues in AI, including societal impacts, energy consumption, and bias mitigation.
(2) Use interactive explainability tools and visualization to identify and address model bias.
(3) Analyze the mathematical foundations of ML algorithms to uncover and address bias.
(4) Integrate ethical practices into machine learning pipelines to address real-world challenges.

4.3.2 Theoretical Foundations. Lectures provided a comprehensive introduction to sustainable AI principles, emphasizing their relevance to societal impacts and alignment with SDGs. Topics included the ethical implications of AI, such as algorithmic bias, privacy concerns, and the environmental impact of training and deploying ML models. Students examined real-world case studies, including the energy consumption of large-scale ML systems and their role in reinforcing or mitigating societal inequalities.

4.3.3 Hands-On Assignments: Homework assignments combined technical problem-solving with reflection questions, prompting students to evaluate the societal and environmental implications of their solutions. This dual focus fostered both skill development and ethical awareness.

4.3.4 Outcomes and Impact. The redesigned curriculum reached approximately 1,500 students. Its effectiveness was evaluated through a survey comparing the Fall 2023 control group (pre-redesign, $n = 51$) with the Spring 2024 intervention group (post-redesign, $n = 59$). The survey measured familiarity with SDGs, confidence in addressing ethical challenges, and perceptions of the curriculum's effectiveness. Key findings included:

- Familiarity with SDGs: 0% of control group respondents were "Extremely familiar" with SDGs, compared to 15.6% in the intervention group.
- Awareness of bias and environmental issues: Agreement that AI can mitigate bias and amplify environmental concerns increased from 26.7% to 34.4%.
- Confidence in handling ethical challenges: "Strongly agree" responses rose from 6.7% to 25.0%.

These findings demonstrate the success of the redesigned curriculum in integrating sustainability and ethics into AI education. Students gained not only technical proficiency but also a nuanced understanding of AI's broader societal impacts, equipping them to address these challenges in their future careers.

5 Key Learnings and Next Steps

The pilot implementation of this framework for integrating the (SDGs) across the computer science curriculum highlighted several critical elements required for success. Interdisciplinary collaboration through a Community of Practice, involving professors from other disciplines, enriched the initiative by providing diverse perspectives and shared resources. Within the department, strong support from leadership, including opportunities to present intermediate results, fostered momentum and engagement among faculty.

One of the most impactful strategies for faculty engagement was providing clear, actionable examples of how to align coursework with the SDGs. Demonstrating how these integrations could enhance student engagement and contextualize technical knowledge in real-world applications inspired many instructors to adopt these changes.

Teaching assistants (TAs) were another pivotal element of the initiative's success. With their frequent and direct interactions with students, TAs significantly influenced students' perceptions of sustainability in computing. Research underscores the role of TAs as motivators and peer leaders [1], making their enthusiasm for sustainability topics essential. Creating a strong community among TAs, emphasizing camaraderie and shared goals, further enabled the successful adoption of tools like carbon emission tracking software (e.g., CodeCarbon [4]) across multiple courses. These efforts helped embed sustainability practices consistently throughout the curriculum.

Despite these successes, integrating the SDGs into the curriculum faced challenges. Faculty attitudes toward the initiative varied. While some instructors were inspired by initial results and discussions at departmental meetings, others expressed skepticism about explicitly linking technical coursework with sustainability goals.

Addressing this skepticism required clear communication and concrete examples to demonstrate the feasibility of integrating sustainability into existing courses. Small, incremental changes proved effective in alleviating resistance and encouraging broader adoption.

In courses where TAs played a significant role in designing assignments and implementing tools, motivating TAs as a cohesive team emerged as a critical strategy for driving adoption. The success of this approach highlighted the importance of fostering a collaborative culture among TAs and faculty, empowering them to become champions of sustainability.

Our goal is to expand this framework to all courses across our institution, guided by the proposed learning outcomes in the context of CSED23. The successes and challenges of the pilot implementation provide a roadmap for scaling these efforts. Key strategies, such as leveraging interdisciplinary collaboration, fostering faculty engagement through clear and actionable examples, and empowering teaching assistants as sustainability champions, will continue to be central to this process. By embedding incremental changes and emphasizing the feasibility of aligning technical coursework with the UN SDGs, we aim to address skepticism and inspire broader participation. This ongoing effort seeks to cultivate strong, conscientious professional identities in students while reinforcing their technical expertise.

5.1 A Call to Action

The integration of SDGs into computer science education is both a responsibility and an opportunity to prepare students for a sustainability-conscious world. The successes and challenges of this pilot highlight the importance of interdisciplinary collaboration, departmental support, and leveraging the unique influence of teaching assistants.

We call on educators to explore opportunities to align their courses with the SDGs, even through small, incremental changes that require minimal disruption. Providing real-world examples and building strong communities of practice can help overcome skepticism and inspire faculty engagement.

Finally, as the field of computing continues to intersect with global challenges, integrating sustainability into the curriculum is no longer optional but essential. Let us commit to fostering a generation of technologists who are not only technically proficient but also deeply committed to addressing the pressing issues of our time. Together, we can make computing education a catalyst for meaningful change.

Acknowledgments

This work was partially supported by an Undergraduate Sustainability Education Innovation grant provided by the Georgia Tech Sustainability Next initiative and administered through Georgia Tech's Sustainability Education and Curriculum Committee and the College of Computing.

References

[1] Grace Barkhuff, Ian Pruitt, Vyshnavi Namani, William Gregory Johnson, Rodrigo Borela, Ellen Zegura, Anu G. Bourgeois, and Ben Rydal Shapiro. 2025. Exploring the Humanistic Role of Computer Science Teaching Assistants across Diverse Institutions. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2025)* (Pittsburgh, PA, USA). ACM, New York, NY, USA, 7 pages. doi:10.1145/3641554.3701861

[2] Stoney Brooks, Xuequn Wang, and Saonee Sarker. 2012. *Unpacking Green IS: A Review of the Existing Literature and Directions for the Future*. Springer Berlin Heidelberg, Berlin, Heidelberg, 15–37. doi:10.1007/978-3-642-27488-6_2

[3] Yu Cai. 2010. Integrating sustainability into undergraduate computing education. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (Milwaukee, Wisconsin, USA) *(SIGCSE '10)*. Association for Computing Machinery, New York, NY, USA, 524–528. doi:10.1145/1734263.1734439

[4] Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stęchły, Christian Bauer, Lucas-Otavio, JPW, and MinervaBooks. 2024. *mlco2/codecarbon: v2.4.1*. doi:10.5281/zenodo.11171501

[5] Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models. arXiv:2303.10130 [econ.GN] https://arxiv.org/abs/2303.10130

[6] Elina Eriksson and Daniel Pargman. 2014/08. ICT4S Reaching Out: Making sustainability relevant in higher education. In *Proceedings of the 2014 conference ICT for Sustainability*. Atlantis Press, 40–47. doi:10.2991/ict4s-14.2014.5

[7] Ali Erkan, Tom Pfaff, Jason Hamilton, and Michael Rogers. 2012. Sustainability themed problem solving in data structures and algorithms. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (Raleigh, North Carolina, USA) *(SIGCSE '12)*. Association for Computing Machinery, New York, NY, USA, 9–14. doi:10.1145/2157136.2157146

[8] Emilio Ferrara. 2024. GenAI Against Humanity: Nefarious Applications of Generative Artificial Intelligence and Large Language Models. *Journal of Computational Social Science* 7 (2024), 549–569. doi:10.1007/s42001-024-00250-1

[9] Luciano Floridi and Mariarosaria Taddeo. 2016. What is data ethics? 20160360 pages.

[10] Markus Hirsch-Dick and Stefan Naumann. 2010. Enhancing Software Engineering Processes towards Sustainable Software Product Design. In *International Conference on Informatics for Environmental Protection*. Shaker Verlag, Aachen, 706–715. https://api.semanticscholar.org/CorpusID:12256854

[11] IEA. 2022. Data Centres and Data Transmission Networks. https://www.iea.org/reports/data-centres-and-data-transmission-networks License: CC BY 4.0.

[12] IEA. 2024. Electricity 2024. https://www.iea.org/reports/electricity-2024 Licence: CC BY 4.0.

[13] Sarah Kreps, Miles M. McCain, and Miles Brundage. 2022. All the News That's Fit to Fabricate: AI-Generated Text as a Tool of Media Misinformation. *Journal of Experimental Political Science* 9, 1 (2022), 104–117. doi:10.1017/XPS.2020.37

[14] Amruth N. Kumar, Rajendra K. Raj, Sherif G. Aly, Monica D. Anderson, Brett A. Becker, Richard L. Blumenthal, Eric Eaton, Susan L. Epstein, Michael Goldweber, Pankaj Jalote, Douglas Lea, Michael Oudshoorn, Marcelo Pias, Susan Reiser, Christian Servin, Rahul Simha, Titus Winters, and Qiao Xiang. 2024. *Computer Science Curricula 2023*. Association for Computing Machinery, New York, NY, USA.

[15] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2024. Estimating the carbon footprint of BLOOM, a 176B parameter language model. *J. Mach. Learn. Res.* 24, 1, Article 253 (March 2024), 15 pages.

[16] Irene Manotas, Christian Bird, Rui Zhang, David Shepherd, Ciera Jaspan, Caitlin Sadowski, Lori Pollock, and James Clause. 2016. An empirical study of practitioners' perspectives on green software engineering. In *Proceedings of the 38th International Conference on Software Engineering (ICSE 2016)*. ACM, 237–248. doi:10.1145/2884781.2884788

[17] Alok Mishra and Deepti Mishra. 2021. Sustainable Software Engineering: Curriculum Development Based on ACM/IEEE Guidelines. In *Software Sustainability*. Springer, 269–285. https://dblp.org/rec/books/sp/21/MishraM21

[18] Cathy O'neil. 2017. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown.

[19] Rui Pereira, Tiago Carção, Marco Couto, Jácome Cunha, João Paulo Fernandes, and João Saraiva. 2020. SPELLing out energy leaks: Aiding developers locate energy inefficient code. *J. Syst. Softw.* 161, C (mar 2020), 15 pages. doi:10.1016/j.jss.2019.110463

[20] Anicia Peters, Rafael Capilla, Vlad C. Coroamă, Rogardt Heldal, Patricia Lago, Ola Leifler, Ana Moreira, João Paulo Fernandes, Birgit Penzenstadler, Jari Porras, and Colin C. Venters. 2023. Sustainability in Computing Education: A Systematic Literature Review. *ACM Transactions on Computing Education* 24 (2023), 1 – 53. https://api.semanticscholar.org/CorpusID:258741008

[21] Jari Porras, Maria Palacin-Silva, and Birgit Penzenstadler. 2017. The Evolving Perceptions of Sustainability in CS and SE Education: Findings from a Master's Programme. In *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEET)*. IEEE, 19–28. doi:10.1109/CSEET.2017.15

[22] Abbas Rajabifard, Masoud Kahalimoghadam, Elisa Lumantarna, Nilupa Herath, Felix Kin Peng Hui, and Zahra Assarkhaniki. 2021. Applying SDGs as a systematic approach for incorporating sustainability in higher education. *International Journal of Sustainability in Higher Education* (2021). https://api.semanticscholar.org/CorpusID:236280974

[23] Jeffrey Saltz, Michael Skirpan, Casey Fiesler, Micha Gorelick, Tom Yeh, Robert Heckman, Neil Dewar, and Nathan Beard. 2019. Integrating ethics within machine learning courses. *ACM Transactions on Computing Education (TOCE)* 19, 4 (2019), 1–26.

[24] João Saraiva, Ziliang Zong, and Rui Pereira. 2021. Bringing Green Software to Computer Science Curriculum: Perspectives from Researchers and Educators. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (Virtual Event, Germany) *(ITiCSE '21)*. Association for Computing Machinery, New York, NY, USA, 498–504. doi:10.1145/3430665.3456386

[25] Patrick Schramowski, Cigdem Turan, Nico Andersen, Constantin A. Rothkopf, and Kristian Kersting. 2021. Language Models have a Moral Dimension. *CoRR* abs/2103.11790 (2021). arXiv:2103.11790 https://arxiv.org/abs/2103.11790

[26] Paul M. Schwartz. 2011. Privacy, Ethics, and Analytics. *IEEE Security & Privacy* 9, 3 (2011), 66–69. doi:10.1109/MSP.2011.61

[27] George M. Slavich and Philip G. Zimbardo. 2012. Transformational Teaching: Theoretical Underpinnings, Basic Principles, and Core Methods. *Educational Psychology Review* 24, 4 (2012), 569–608. doi:10.1007/s10648-012-9199-6

[28] Irene Solaiman, Zeerak Talat, William Agnew, Lama Ahmad, Dylan Baker, Su Lin Blodgett, Canyu Chen, Hal Daumé III, Jesse Dodge, Isabella Duan, Ellie Evans, Felix Friedrich, Avijit Ghosh, Usman Gohar, Sara Hooker, Yacine Jernite, Ria Kalluri, Alberto Lusoli, Alina Leidinger, Michelle Lin, Xiuzhu Lin, Sasha Luccioni, Jennifer Mickel, Margaret Mitchell, Jessica Newman, Anaelia Ovalle, Marie-Therese Png, Shubham Singh, Andrew Strait, Lukas Struppek, and Arjun Subramonian. 2024. Evaluating the Social Impact of Generative AI Systems in Systems and Society. arXiv:2306.05949 [cs.CY] https://arxiv.org/abs/2306.05949

[29] United Nations. 2015. Transforming our world: the 2030 Agenda for Sustainable Development. https://sdgs.un.org/2030agenda Accessed: 2024-11-13.

[30] NU Yamaguchi, EG Bernardino, MEC Ferreira, BP de Lima, MR Pascotini, and MU Yamaguchi. 2023. Sustainable development goals: a bibliometric analysis of literature reviews. *Environmental Science and Pollution Research International* 30, 3 (Jan. 2023), 5502–5515. doi:10.1007/s11356-022-24379-6 Epub 2022 Nov 24.

[31] Francisco Zamora-Polo, Jesús Sánchez-Martín, Mario Corrales-Serrano, and Luis Espejo-Antúnez. 2019. What Do University Students Know about Sustainable Development Goals? A Realistic Approach to the Reception of this UN Program Amongst the Youth Population. *Sustainability* 11, 13 (2019). doi:10.3390/su11133533