



**MIT Schwarzman College of Computing Task Force
Working Group on Organizational Structure
Final Report**

Asu Ozdaglar and Nelson P. Repenning, Co-Chairs
August 5, 2019

OVERVIEW

The applications of computing generally and computer science specifically have expanded significantly in the last two decades. At MIT, a growing segment of the faculty use computational methods in their research, and more than a quarter of our undergraduates major in computer science. An appreciable portion of this growth results from work that has happened at MIT.

It is unusual for interest in an academic discipline to expand so rapidly, and the ways in which the Institute organizes and manages research on and teaching about computing has not evolved quickly enough to match present needs. Launching the Schwarzman College of Computing (SCoC) thus constitutes a critical opportunity to rethink and reorganize the ways in which computation is studied, taught, and used at MIT.

We begin our report with a brief summary of the state of computing and computer science at MIT. We then argue that our current approach to organizing computation creates many of the observed challenges. Following that, we discuss new structural configurations that could begin to alleviate the current tensions, inefficiencies, and barriers to collaboration. While our working group agrees on many of the potential benefits of these models, we found the question of membership to be more complicated. We define this debate and evaluate both sides. Finally, we conclude with broader implications for the processes through which we organize and reorganize knowledge production and dissemination at MIT.

KEY IDEAS

OBSERVATIONS, EVALUATION, AND PROBLEM FORMULATION

Computing Within EECS

Electrical Engineering and Computer Science (EECS) is the largest department at MIT with 129 faculty members, 1347 students in its five majors, 737 SM and PhD students, and 331 MEng students.¹ EECS activity is organized as a matrix in which the research labs—RLE, MTL, CSAIL, and LIDS—provide space and research support and the department administers the teaching programs and handles the hiring and promotion of faculty.

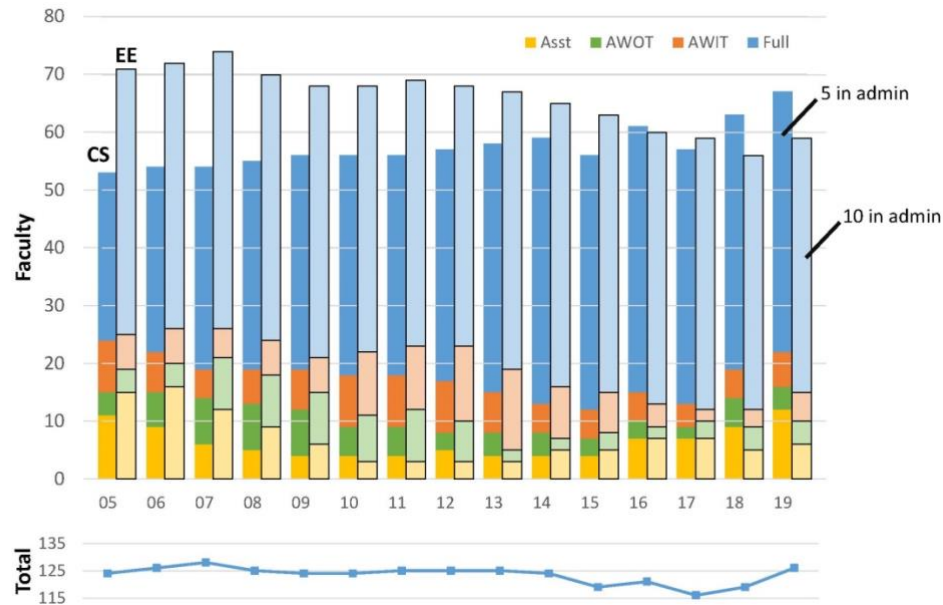
Leveraging the flexibility provided by this organizational structure, EECS has played a pioneering role in the major advances of many (if not most) aspects of computing during the last 70 years, including algorithms, theory of computing, systems, artificial intelligence, machine learning, information science, and hardware of computation. The result of this success has been unprecedented growth in the demand for computing education. The department has responded to this challenge by directing its resources (including faculty lines, TAs, and lecturers) to computer science.

The EECS system (the department plus the labs) now suffers from three major problems. First, demand for computing-related courses significantly exceeds the department's ability to supply them. Many of the core undergraduate courses are taught in very large sections, often exceeding 400 students per year in more than 10 courses (with more than 500 per term in five of the courses). The scale of these courses requires a high degree of standardization that both limits the quality of the student experience and the ability of faculty to adapt the course content to new developments in the underlying content.

Although the shift of faculty lines to CS has helped to address some of the immediate teaching demands, it has come at the cost of significant reductions in the resources dedicated to the electrical engineering side. Although there is little reason to believe that its role in delivering the

¹ The five majors managed by EECS are 6-1: Electrical Engineering, 6-2: Electrical Engineering and Computer Science, 6-3: Computer Science, 6-7: Computer Science and Molecular Biology, 6-14: Computer Science, Economics and Data Science (AY 19 numbers).

Institute’s mission has diminished, over the last decade the headcount dedicated to electrical engineering has contracted more than any other department at the Institute. The figure below shows the change in the number of EE and CS faculty from 2005 to 2019.²



Second, although the overwhelming majority of EECS faculty report belief in the synergy across their respective areas of research, the tension between the two groups is palpable and manifests in a variety of pathologies. Most notably, the two factions fight over contested research areas that do not obviously fit in either EE or CS or simultaneously fit in both (e.g., machine learning). In doing so, the department can both scare away potential hires and confuse current and prospective students.

Moreover, such tension is exhausting (particularly to administrators) and distracts from delivering the group’s core missions. Prior to the emergence of the SCoC, the department engaged in serious discussions about splitting EE and CS, but the consensus remained that both research and teaching benefitted from having the full computing stack—hardware, systems, theory, and inference and decision making—in one group.

Third, lab divisions have added to the strains. The research activities of EECS faculty are distributed across EECS-affiliated research labs that support interdepartmental collaborations and provide

² Within EECS, the number of EE faculty has dropped from an average of 70.5 faculty (2005-2010) to 59 in 2019 (-11.5), while CS faculty numbers have grown from 54.5 faculty (2005-2010) to 67 (+12.5) over the same timeframe.

invaluable intellectual diversity. The current lab structure, however, with EE faculty belonging to LIDS, RLE, and MTL and CS faculty to CSAIL, has created artificial boundaries between research topics that reverberate across the labs.

The convergence of research areas—machine learning, statistical inference and optimization, communication and networking, robotics and autonomy, information theory, coding and cryptography—has exacerbated the problem, causing EECS research activities to be increasingly split across these labs. Several research groups are interested in and working on exactly the same set of research questions but in separate pockets. That space is controlled and allocated by labs (with differentiated financial management falling under federal oversight regulations) results in these research groups being in distinct locations, separating faculty and students and limiting interactions and opportunities that being in the same department would otherwise provide.

At the root of the above challenges facing EECS is the binary characterization of faculty as either EE or CS. Making the lab/department matrix work requires effective evaluation and decision-making on both sides—the department hires and promotes; the labs provide resources and foster collaboration. Although the department is increasingly split, the decision-making processes are premised on a more uniform distribution of interests and preferences. Said division is unlikely to be problematic for candidates who fit squarely within one of the two identities but will create problems for those who straddle boundaries.

More generally, mustering political will on both sides slows the natural evolution of the smaller sub-group structures as the two sides vie for ownership of hot topics and strong candidates. The department side should probably evolve more slowly than the lab side, but that doesn't mean its structure should be fixed. The inability of the department side to reorganize around emerging topics puts the system at a disadvantage in recruiting candidates who seek critical mass. As perhaps a forced and unfortunate analogy to U.S. politics, the intensity of divisions between red and blue forces homogeneity within the two colors. We probably are losing various shades of each color and purple hues in the process.

Finally, as the growing division results in a less adaptable system, we risk creating additional complexity to work around the current structure. As organizations go, MIT is medium-sized, but it can compete with anyone when it comes to the number of labels used to organize its activities. Do

we need schools, departments, labs, centers, initiatives, institutes (within a larger Institute), and now a college, to deliver our mission? While this might be necessary given the breadth and complexity of the things MIT does, an equally credible argument can be made that the new distinctions and structures are patches to compensate for flaws in existing structures. During our investigations, several members of the faculty suggested that complexity and the increasingly fragmented ownership structure of certain areas such as machine learning and data science do not help MIT to develop a coherent intellectual agenda and present a unified image to the outside world.

Computing Outside of EECS

The creation and use of computational methods outside of EECS also (and not surprisingly) has grown considerably. As an important example, the Center for Computational Engineering (CCE) focuses on developing new computational methods to solve otherwise intractable problems in science and engineering as well as applying state-of-the-art methods to a variety of high impact problems. CCE's 75 affiliate faculty span 10 departments and offer two graduate degrees. Like several other computational research programs at MIT, the boundary between the work of CCE and computer science is becoming increasingly blurry. Several productive opportunities for collaboration appear to exist, only some of which are being exercised.

Other important examples include the Operations Research Center (ORC), which educates students in the fields of operations research and analytics, and the Institute for Data, Systems, and Society (IDSS), which focuses on the integration of statistics and data science, information and decision systems, and social science to address new and emerging societal challenges. ORC offers graduate degrees in operations research and business analytics, and IDSS offers a PhD in social and engineering systems and an interdisciplinary PhD in statistics.

More generally, a review of the computing white papers produced by each of the departments suggests that most of them are in the market for faculty who make significant use of computing in their research. While many fields in science and engineering have a long history of using computational approaches, other disciplines are just beginning. Even disciplines that once relied on office-bound approaches to conducting research are now recruiting faculty with startup packages

that include significant computing resources, packages that now much more closely resemble those offered in engineering and science.

Computationally-based research is clearly thriving at MIT, but we see challenges here, too. First, the explosion in the demand for CS courses and majors has come at the expense of other departments. While one might initially argue that this represents the natural evolution of student interest in response to the growing demand for those trained in computer science, extending this argument takes MIT to a bad place.

Interest in computer science and computational methods is clearly growing, but it seems unlikely that this growth is being accompanied by a commensurate reduction in the importance of other subjects taught at MIT. The world is still likely to continue to need excellent material scientists, biologists, economists, and people well trained in all the other disciplines at MIT.

The meaning of “well trained,” of course, is evolving rapidly. In many of these fields, the definition includes a heavy dose of computational methods, which is not the same as saying that all our students should be computer science majors. The problem MIT thus faces is how to satisfy the burgeoning demand for computer-science related topics without short-changing the important work being done in the development and application of computational methods in other critical fields.

Second, as noted above, it appears that we are not capitalizing on all the opportunities for research that combine computing with other disciplines. MIT is a big place and faculty are busy. Whereas novel research collaborations often emerge from random interactions and casual conversations, we have made few efforts as faculty to insure that they occur with sufficient regularity.

Third, several MIT faculty (and other prominent scholars) argue that the explosive growth of computing technology has not been accompanied by a sufficiently critical evaluation of the changes that such growth is catalyzing in broader society. Most notably, the ubiquity of connected devices enables the collection of unprecedented volumes of highly personal data, and advances in sensing and machine learning allow the increasing automation of tasks and decision that were once the exclusive province of human beings.

Whether and how such technologies should be understood, managed, and, perhaps, regulated demands careful consideration. Moreover, that consideration can neither be the exclusive province of the inventors of technology, who often are not trained to understand social and cultural contexts, nor can it be entirely outsourced to those who lack a deep understanding of the technology in question. Properly locating computing within the context of a civil, democratic society will require collaboration across disciplinary boundaries.

The Fundamental Tension

Combining these two sets of observations reveals a fundamental tension that probably afflicts all disciplines with methods that can be applied to multiple topics. Faculty who develop computational methods, particularly those in computer science, express a strong desire to maintain methodological focus and the highest scholarly standards in both research and teaching on computation. At the same time, there is an equally strong desire to diffuse computing throughout the disciplines, even though the state of art in applying computational methods varies widely. Some areas have used computational methods for several decades; others are just beginning. Developing a computing ecosystem that is both welcoming and inclusive and maintains high standards is a central challenge to which our efforts are directed.

Targets

Informed by the observations noted above, our working group identified four targets to aim at when developing a potential structure for the SCoC:

- Enable and support world-leading research and education in computing.
- Support interdisciplinary research between computing and other academic disciplines.
- Promote integration of computing into curricula across the Institute.
- Incorporate social science perspectives as a critical component of computing research.

CHALLENGES WITH THE MOST OBVIOUS SOLUTION

The problem of organizing the SCoC might initially appear to have an obvious solution: move all of the computer scientists currently in EECS into the SCoC and rename Course 6 Electrical Engineering. After careful consideration, our working group concluded that, though simple, applying this model creates multiple challenges. First, as mentioned above, whereas EE and CS are

often represented as mutually exclusive categories, much of the research and teaching in the department spans the two areas (e.g., machine learning and data science). Forcing each member to pick one of the two is likely to create additional impediments to research and teaching that cross and straddle this artificial divide. It is easy to imagine both the SCoC and the newly reconstituted EE department offering competing courses in topics such as machine learning that do not fit squarely into either category.

Second, given the resources and prominence associated with the SCoC, faculty who span the EECS divide are likely to elect to go to the SCoC, thus damaging the vibrant tradition of world-class research in electrical engineering at MIT. Electrical engineering as a discipline focuses on building systems for processing energy and information by marrying physics with mathematical and computational techniques (including signal processing and control). Splitting the department between the SCoC and SoE, combined with the resources and the excitement that the SCoC is generating, will likely result in the mathematics and computational side of EE joining the SCoC, leaving EE as an applied physics department that is well below critical mass.

Such a pathological sorting would not only be a break with MIT's history in this area but would strongly contrast with the trend in electrical engineering departments in peer institutions, where the mathematics of EE and computing are strongly present. An exodus of top EE faculty would not be out of the question in this case. To repeat a point made above, though the prominence of computer science has clearly grown, it is difficult to argue that it has come with an equal decline in the importance of electrical engineering. Nonetheless, forcing a binary choice between the two categories could effectively impose a zero-sum constraint on the Institute's activity in those two areas.

Third, such an approach risks reinforcing the notion that the SCoC is just a school by another name. As mentioned above, thanks to the lab structure, MIT is often a collaborative place when it comes to research. Teaching across departmental and school boundaries can be more challenging, and in the flux of busy semesters, we all risk retreat to our disciplinary corners. Such dynamics are particularly acute when the demand for core courses exceeds the natural supply. Though perhaps conducive to computer science research, giving the computer science faculty a new home in the SCoC without additional structure to promote and support collaboration is unlikely to achieve the goals of stimulating cross-disciplinary research and teaching.

A POTENTIAL STRUCTURE

Our group reached broad agreement on several dimensions of a potential structure for the SCoC. We outline this configuration in three steps. First, we discuss four different ways in which faculty might interact with (or be members of) the SCoC community. Second, building on the membership conception, we suggest amending the matrix structure, introducing a configuration we call the Common Ground that details how the execution of research and the delivery of teaching might be organized. Third, we discuss three ways that this new teaching structure might be staffed, eventually finding that the third option tallied many more pros than cons when measured against the challenges listed above and could serve the Institute's interests well.

Membership

Single vs. Multi-Community Faculty Members

Our working group discussed two ways in which faculty could be permanent members of the SCoC. Following the labeling suggested by the Faculty Appointments working group, we call these Single Community Faculty (SCF) and Multi-Community Faculty (MCF). SCF have their main or only appointment in the SCoC (similar to faculty who have their only appointment in a single department). SCF would be hired and managed by SCoC leadership and other faculty as needed.

MCF represents scholars who cross the boundary between another MIT department and the SCoC. The MCF appointment model would be based on a fixed division of effort between an existing department and the SCoC (e.g., 50/50). The suitability of existing MIT faculty for MCF positions would be decided by a cross-disciplinary/departmental entity that we call the Computing Council. Members of the Computing Council would be appointed by academic council, with oversight from the SCoC dean.

In this model, the search committee for an MCF line would originate in an existing department, but the SCoC would provide one member for each MCF search committee. Initially, all new MCF would satisfy their SCoC teaching commitments (based on the fraction of their appointment that resides within the SCoC) via co-teaching with a senior member of the SCoC. The SCoC also

would provide each junior MCF with a mentor. Promotion and tenure would happen within the home discipline, but one or more letters would be solicited from relevant SCoC senior faculty, and one SCoC faculty will sit on the promotions committee.

In this conception, MCF slots would not be permanent—a feature that we consider to be very important to this approach. At the discretion of the SCoC dean and the Computing Council, the SCoC portion of the slot could be pulled and the faculty member in question would return to full-time service in his or her home department. Obviously, this creates some risk for the home department as it could end up being overstaffed when surprised by a return. As discussed in more detail below, however, this system would create a strong incentive for the department to collaborate closely with the SCoC and to insure that its MCF hires are engaging in research that genuinely crosses the boundary between their home disciplines and the Common Ground of computation.

Incubator Groups

Our working group also discussed two options for how faculty who do not have a permanent home in the SCoC could engage with it. The first of these is as members of an incubator group. As discussed above, the nature and uses of computing are changing rapidly, and it would be impossible for any single individual or leadership team to anticipate all the potential future advances and applications. Consequently, we suggest that the SCoC offer groups of faculty the opportunity to apply for support and space in which to incubate new research topics and teaching opportunities.

We envision a competitive process in which interested groups submit proposals to an interdisciplinary faculty committee charged with making awards. Incubation awards targeting teaching would be largely focused on developing curriculum that pushes computing education from the SCoC to the other schools and departments. Research awards would focus on new topics and activities that are not supported by the existing labs and research programs.

In this model, incubation grants would be of finite duration, perhaps two to three years. At the end of that period, the activity in question would either become self-sustaining (in the form of a fundable research program or sustainable course), or it would end. Incubation awards would cover both faculty salary expense—so that originating departments can backfill teaching

responsibilities—and graduate students or postdoctoral scholars. In addition, awards would include dedicated space for collaboration that sits within the physical confines of the SCoC. Incubator groups also could include (though perhaps not exclusively) participants who are not members of the MIT community.

Faculty Fellows

Fellowships offer a second way for non-SCoC faculty to engage. These fellowships would allow interested faculty the opportunity to sit in the SCoC for a year or more in a manner similar to that offered by other research institutes such as the Simons Institute for the Theory of Computing and the Center for Advanced Study in Behavioral Science at Stanford. These slots could be dedicated either to faculty who wish to contribute to computing or those wishing to infuse their current research with computational methods. As with the incubation grants, these slots would be allocated in a competitive process administered by an interdisciplinary committee of faculty. Fellowships also could be used to attract faculty, research scientists, or post docs from other institutions (perhaps with the hope of future recruitment to a permanent position at MIT). A specific proposal on these fellowships is provided in the appendix.

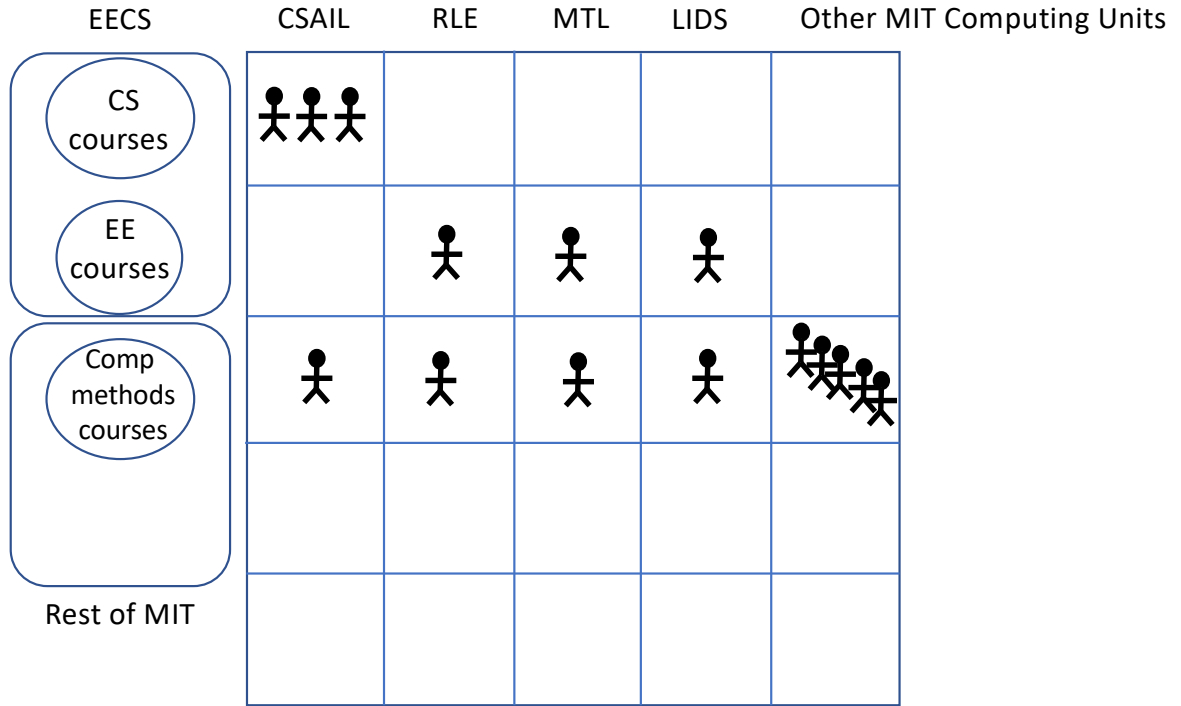
Updating the Matrix to Create Common Ground for Teaching

The matrix organization that links the EECS department and the research labs has served both the Institute and the field well, generating tremendous growth in the theory and applications of computing. As discussed above, however, the research on and uses of computational methods has outgrown the current incarnation. We propose two potential sets of changes to the matrix structure.

First, we envision creating a Common Ground for teaching foundational elements of computation by reorganizing the teaching side of the matrix into a set of new teaching groups that better matches current teaching needs and enables MCF faculty from other departments and schools to teach core computing courses. We have widespread enthusiasm in our working group for this reorganization. The second set of changes revolves around the composition of the teaching groups and the question of whether they would consist of a mix of SCF and MCF or MCF only. This point was controversial, but our working group ultimately found many more pros than cons in one of the models. We discuss both elements below.

Current Matrix Structure

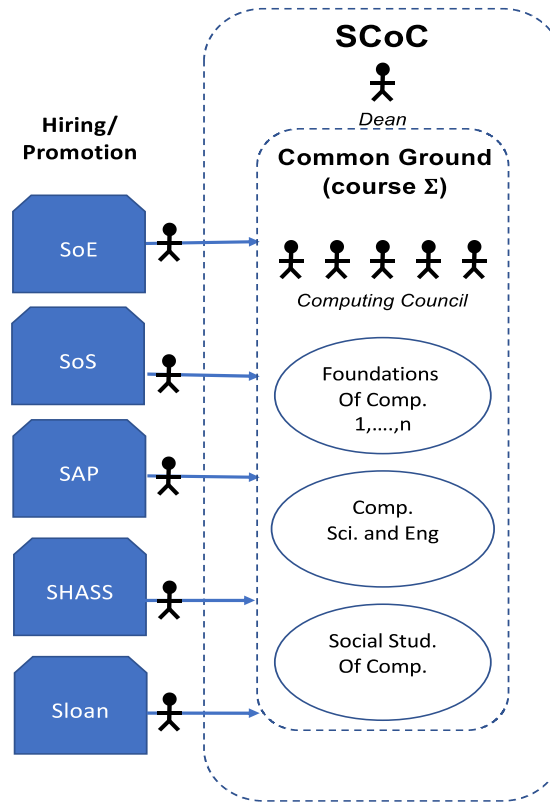
A stylized version of the current approach to structuring computing is shown in the figure below.



Three features are notable. First, the EECS department is built on a binary division between the two disciplines in its label. Perhaps most significant, although CSAIL draws faculty from 11 different departments, it does not include a single EECS department member as a PI who identifies as an electrical engineer. In fact, it appears that the default definition of EE in the department is not being a member of CSAIL. Second, although faculty from other departments participate in research on computer science, not all these faculty are engaging with or being engaged by the EECS department in delivering core computer science courses—despite the overwhelming demand. Third, faculty outside EECS participate in a variety of computationally oriented research activities (some of which happen in CSAIL) and teach courses in computational methods. Typically, however, they do not participate in teaching core elements of computer science.

Potential Reorganization of the Matrix

The first dimension of our proposal is shown in the figure below.



The key idea is to decouple the delivery of teaching from the traditional departmental structure by creating a Common Ground (course Σ) composed of several cross-departmental teaching groups. One of the major challenges of the existing structure is that the current grouping of activities does not match student demand. We envision that instruction in the foundations of computation could be reorganized into a set of groups that represent the major sub-topics in computing (recognizing that no labeling scheme will be perfect).

We discussed three types of potential groups. The first offers courses focused on theory and methods that are foundational to most, if not all, applications of computation. These might include basics like programming and algorithms as well as rapidly expanding topics like machine learning. Our working group did not arrive at a definitive final scheme for the number and labeling of these groups. The second set of groups would focus on courses that provide the foundations for computational science and engineering. The third grouping would contain courses focused on the social and societal implications of computing.

The first collection of groups would subsume, but not be limited to, much of what happens currently in the EECS department and draw heavily on computer science fundamentals. The second set would draw heavily on courses taught by faculty connected to CCE. The third set would draw heavily on faculty in SHASS and Sloan. The key change embodied in this approach relative to MIT's existing model of a school is that each teaching group would be staffed by either a combination of SCF and MCF or entirely by MCF (more on that later in this report).

Several potential benefits could accrue from this structure. First, implementing this conception forces a set of conversations around what constitutes the foundations of computing. To meet the target of producing bilingual students, we need to agree on the elements of the common language in which everyone will be trained. Creating a Common Ground for computation education will go a long way towards meeting such an objective.

Second, we expand the number of faculty who can teach critical computation courses in a coordinated fashion as MCF with partial appointments in other department and schools. Those MCF could help define and teach Common Ground courses. Third, curricular administration and delivery provide a concrete set of activities around which the associated faculty would have to collaborate, thus creating a key mechanism for preventing disciplinary isolation.

Finally, if these activities are at least partially staffed by MCF and we implement an effective incubation structure, the system could be more dynamic and adaptable than our traditional department structure. With incubation in place, faculty who believe they have developed (or at least identified) a new Common Ground offering could secure the resources to develop these ideas. If the new offering is met with sufficient demand, it could (with the approval of the Computing Council) become a Common Ground group. Conversely, if currently offered topics become less popular or otherwise are no longer considered Common Ground, the Computing Council could effectively move that member of the faculty back to his or her home department by pulling the portion of the slot it is funding.

This final scenario does create risks for departments in that they could end up overstaffed with faculty members who are not easily matched to courses in their home departments, thereby precluding future hiring. We believe, however, that these risks would provide the needed

incentive for the departments to use their MCF lines to support genuinely cross-disciplinary research and teaching. If a department uses an MCF line to fulfill another need under the guise of computing, for example, it risks having the slot pulled and having to cover all of that line rather than a portion of it. There would undoubtedly be difficult conversations on occasion—nobody likes being sent “home”—but this may be a necessary cost of creating a more vibrant, adaptable system for delivering cutting-edge education in computation.

Our working group agrees on the potential value of teaching through the creation of a Common Ground that results in a set of foundational courses in computing delivered by faculty from around the Institute. The disposition of the EECS department within this structure was more controversial, but one of our three potential models ultimately proved to garner many more pros than cons.

The SCoC and EECS

We considered three different models for how the EECS department might connect to the SCoC.

- SCF Only ($SCoC=EECS$)
- MCF Only ($SCoC \cap EECS = \emptyset$)
- MCF/SCF Hybrid ($SCoC \supset EECS$)

SCF Only ($SCoC=EECS$)

Although it was rejected by our working group—and it stands in contrast to the Common Ground notion described above—consider first a model in which the SCoC is staffed entirely by EECS faculty members (who would all be SCF). In this conception, following our discussion above of splitting the EECS department, the SCoC would become a school of computer science or a school of computer engineering and science with all the strengths and weaknesses associated with the school model.

That configuration would be excellent for maintaining high scholarly standards and ensuring the depth of computer engineering and science research at MIT. Its major weakness is that it would do little to insure that computing is infused across campus. Moreover, it would likely further insulate computer science faculty from other computational research at MIT, thus impeding cross-pollination.

MCF Only ($SCoC \cap EECS = \emptyset$)

At the other extreme, consider a model in which the SCoC has no permanent members and is entirely staffed by MCF. Every participant would have two homes in this configuration, one in an existing department and one in the SCoC. This option has a roughly opposite set of costs and benefits compared to the SCF-only model. The main benefits lie in the large number of connections to the rest of campus and the dynamism it would encourage. Computing would likely move easily from the SCoC to other departments and back again, and the SCoC dean and Computing Council could respond quickly to changes in student demand.

We see two major costs. The first would be a diffusion of responsibility. Without any faculty completely committed to the SCoC, there is a risk that no one would view a particular course as their unique responsibility. If this were the case, the SCoC would run the risk of spreading course and curricular ownership in ways that may hurt quality. That said, if the SCoC dean and Computing Council are willing to pull slots, it would serve as a strong incentive to make sure classes are delivered effectively.

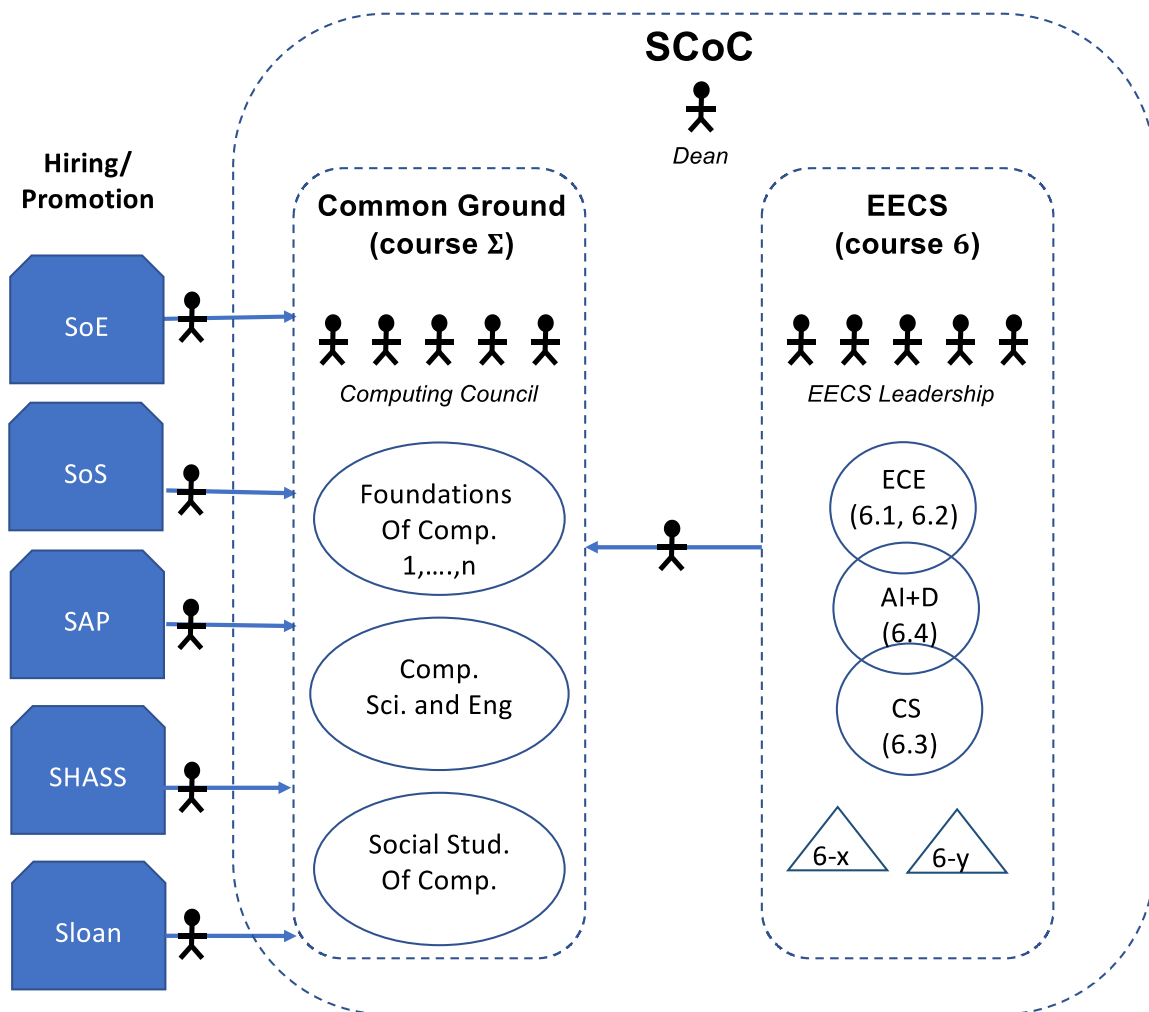
The second potential challenge with this model concerns the disposition of computer science faculty. If the SCoC is not their home, where do they sit, both literally and figuratively? The same question also could apply to EECS faculty in related areas such as machine learning. The seemingly obvious choice is for CS and EE faculty to remain in the EECS department and act as MCF just like everyone else involved in the SCoC. This scheme has its own potential challenges, however.

To start, the CS faculty on our working group saw many drawbacks to this model, suggesting that it could be confusing to have a college of computing that is not the primary home to the Institute's computer science faculty. It is not clear how prospective hires might react to this structure. In addition, it might require a complex network of cross-listed courses, as CS faculty would still be teaching their non-Common Ground offerings, and the CS-related majors would still be managed by EECS. More generally, both undergraduates and graduate students who focus on computer science would be getting their degrees from EECS, not the SCoC.

Despite these costs, several members of our working group initially believed that the costs would be outweighed by the benefits and that MCF-only had more pros than cons. The school model is MIT's default operating mode, and if we wish to change it, we need to intervene decisively to counter our habitual tendencies. An MCF-only configuration would be a strong statement to both the Institute and the world that we treat computing differently and are committed to infusing it throughout our research and teaching. Even if, ultimately, you do not find these arguments compelling, they do speak to the necessity of clearly signaling that we are committed to a new and dynamic approach to research on and teaching about computing at MIT.

MCF/SCF Hybrid (SCoC \supset EECS)

The final option is to pursue the Common Ground set-up discussed above while also moving the EECS faculty into the SCoC. The challenge here lies in ensuring that EECS continues to thrive as an academic unit while making sure that the associated faculty do not come to dominate the Common Ground. We believe that both objectives could be met with careful attention to governance. The essential idea would be to create two groups of roughly equal authority to represent and advocate for each of these outcomes and charge the SCoC dean with ensuring the final balance. A potential structure aligned with this option is shown in the figure below.



In this set up, although EECS faculty would be permanent members of the SCoC (from the perspective of the Common Ground) they would remain MCF—just like every other participant. As such, they would participate and collaborate with other faculty in designing and offering Common Ground courses.

EECS would still retain control of its majors, although those majors would presumably draw heavily on Common Ground courses. Similarly, the EECS faculty would manage its own PhD program and offer advanced courses (likely under the Course 6 label) that are uniquely needed by EECS students.

Within this structure, we also see an opportunity for EECS to reorganize and refocus, especially to go beyond its previous binary structure. The most natural way to achieve this would be to build three semiautonomous groups within EECS:

- Electrical and Computer Engineering (ECE)
- Computer Science (CS)
- Artificial Intelligence and Decision Making (AI+D)

The spectrum of activities in EECS could be divided naturally into three overlapping areas: ECE, CS, and AI+D. ECE would represent the hardware side of computation, focusing on new technologies, devices and systems, quantum computing and engineering, and information science (including representation, storage, and communication of information). CS would represent much of traditional computer science, including systems, theory, and algorithms. AI+D would combine machine learning with our traditional strengths in information analysis and decision sciences to create a new area that purposely spans the EE and CS divide.

The major risk of this approach is not being able to strike the balance between the two constituent parts of the SCoC, Common Ground and EECS. Two potential scenarios represent undesirable outcomes. EECS could come to dominate the SCoC, thus yielding a school rather than the original vision of a college of computing. Alternatively, the legacy of excellence in electrical engineering and computer science could be lost in favor of service functions that provide excellent instruction to the rest of MIT but do not continue to advance the state of the art.

Our working group believes that these risks could be managed and overcome. We also believe that this potential structure represents the most favorable balance of pros and cons for the SCoC.

IMPLEMENTATION

Up to this point, we have discussed our models as though they could be successfully implemented by fiat. In reality, moving from the existing structures to a new configuration would be far from trivial. Here, we briefly discuss how these models might be implemented and highlight some of the

complexities and challenges, particularly as they relate to the disposition of the current EECS department.

The most immediate implementation challenge concerns those in the EECS department who identify as engineers. Not surprising, those who identify as electrical engineers wish to remain associated with the world's leading school of engineering and not be entirely subsumed by the SCoC.

This problem could be solved by allowing Course 6 to remain in the SoE, creating a new course number for those who wish to be in the SCoC, then allowing existing EECS faculty to self-select. Those who wish to be labeled as engineers would stay in SoE, those who wish to be labeled as computer scientists go to the SCoC. As discussed earlier in this report, however, this approach creates the same sorting problem mentioned above: the bulk of the EECS members having a strong incentive to identify as CS and leave the EE department below scale. This also would create the related problem of determining who gets to hang on to the "Course 6" label— which, though entirely symbolic, has the potential to produce a conflict worthy of a *Game of Thrones* sequel.

To avoid instantiating the existing EECS divide and consequent acrimony in the new structure, an interim approach would be to temporarily construe the entire department and all its courses as sitting both in SoE and SCoC. EECS faculty would then be free to continue their teaching and research without the need to declare their allegiance to one side or the other. Temporarily treating EECS as one large group with two homes would allow its members to engage with the new SCoC teaching structure while at least postponing any forced sorting.

Our hope is that the reorganization of EECS into ECE, AI+D, and CS would represent vibrant activities in increasingly active areas and fit well with the vision of the SCoC. We believe it also would lead to productive interactions across the EE/CS divide, particularly in the AI+D group, and begin to remove the current boundaries. Having developed a more productive set of intra-departmental interactions, any subsequent sorting or labeling would likely be more productive and less contentious.

The major cost of this approach is simply that managing a group of 130 faculty who report to two deans would be complicated. However, the current department already is managing most of that

complexity. Incurring the temporary costs of continuing with this administrative overhead appears to have more upsides than locking in a more efficient, but clearly suboptimal, structure.

SUMMARY: STRUCTURING KNOWLEDGE PRODUCTION

The challenges facing computing at MIT are largely the consequence of success. While the Institute has likely always adjusted its structure to match the changing landscape of research and teaching, the rapid expansion of computing has been a bit of an extreme-conditions test demonstrating that the pace of innovation now exceeds our existing adaptive processes.

While launching an entirely new entity like the SCoC feels like a unique moment, this is probably not the last time that MIT will face a mismatch of its own making. If this episode has implications beyond the study and teaching of computing, it is perhaps that we need to develop more dynamic structures for reorganizing ourselves as the state of knowledge and practice evolve.

MIT faculty have proven quite adept at working around structural impediments, often creating significant innovations in the process. It is not clear, however, that such work-arounds represent the optimal configuration for knowledge production and dissemination. Explicit attention to the ways in which MIT reorganizes itself might yield an even more productive, more impactful, and more collegial environment.

APPENDIX

PROPOSAL FOR THE SCHWARZMAN COLLEGE SOCIETY OF FELLOWS (SCSOF)

The report of the working group on the organizational structure of the SCoC includes the concept of “Faculty Fellows” as one of the ways for non-SCoC faculty to engage with the SCoC. This proposal elaborates on how Faculty Fellows would benefit MIT through the SCoC. We believe that a Society of Fellows is a valuable contribution to MIT and the SCoC—whatever its ultimate configuration with respect to permanent, rotating, single, or multi-community members. The Schwarzman College Society of Fellows (SCSoF) would be a permanent organization with rotating membership distinct from the proposed research clusters or departments. We offer a proposed structure for the SCSoF below.

This concept builds on similar units at other universities such as the Simons Institute for the Theory of Computing, Berkeley’s Miller Institute, the Center for Advanced Study in Behavioral Science at Stanford. It also builds on the several Societies of Fellows at Harvard, Princeton, Columbia, and Michigan, which often host scholars in the humanities as well as social sciences. Each of these organizations combines its organizational features in different ways and often with different substantive foci and institutional emphases. MIT would invent its own particular flavor—and that flavor would ensure a cross-Institute computing-focused alliance of disciplines. The central benefit is a permanent organization, with possibly unique endowment and name, but with rotating members from outside as well as inside MIT. The Weatherhead Center for International Affairs at Harvard is another possible model that combines internal and external appointees.

We offer this proposal for a SCSoF to create a center through which:

- All departments, laboratories, and centers will have a stake in the SCoC.
- Barriers for general participation in computational research and curricula will be minimal.
- Cutting-edge research takes place in the SCoC for all disciplines, including computer sciences, engineering, sciences, humanities, and management.
- The SCoC would have a unique contribution to MIT’s educational commons.

While the entire organizational structure of the SCoC may facilitate participation and investment from across the Institute, the SCSoF has particular advantages:

- Rapid adoption of computational fields as they develop.
- High external and internal visibility.
- Accelerated commencement of SCoC activities.
- Accelerated diffusion of innovations across MIT and externally.
- Naming opportunities for fundraising at several scales/amounts.
- Bringing to MIT expertise in fields that we do not offer but are essential to the digital world (e.g., law).

Structure

1. MIT faculty appointments to the SCSoF will be for a limited term (e.g., three to four years). The SCSoF Fellows will be selected competitively by an internal advisory committee with cross-MIT representation. These may be independently named or endowed (e.g., The XYZ Computing Fellow). The MIT-SCSoF Fellows would come from a range of disciplines and departments and would retain their positions in their respective departments. All promotion considerations would remain in those departments. Each Fellow would have resources to support their research and teaching efforts.
2. The center also would host visiting external scholars or experts. Visits would typically be one to two years. The Visiting-SCSoF Fellows would be selected competitively by the internal advisory committee in collaboration with a specifically appointed selection committee. The visiting Fellows would come from other academic institutions, research labs, or industry. They might be relatively senior researchers or promising, newly established early career researchers. Applications would be solicited annually (with a rotation established from the outset). The Visiting-SCSoF Fellows might be separately endowed with a distinct name, which would provide additional funding opportunities.

In addition to their research and collaboration, the Visiting-SCSoF Fellows would be obligated to contribute to the MIT educational commons. They might, for example, create a computation module that would be jointly offered by a department and the SCoC. They might offer a short course, or they might contribute a few MITx lectures via the Office of Digital Learning (ODL).

3. The SCoC could also create a competitive fellowship for postdocs (SCoC Postdoc Fellows). Each postdoc would be assigned to one or more Faculty Fellows (i.e., two Fellows may co-advise two postdocs). Postdoc Fellows would be obligated to contribute to the educational as well as research contributions of the faculty partner. This obligation would not only contribute to the MIT curriculum but would provide materials visiting that Fellows might take back to their home institutions (if they are academically appointed).

Contributions to MIT

All Fellows (MIT, visiting, or postdoc) would be expected to participate in seminars designed for a wide audience (i.e., all members of the SCoC and MIT community). The seminars would be organized by a Postdoc Fellows committee. More specialized seminars would be expected to arise naturally. The SCSoF would have its own seminar with regular sessions scheduled to develop internal community.

In addition to their computing research, SCSoF Fellows would be expected to develop a module supporting computing for the Institute, teach a portion of a class, or develop a short series of MITx lectures. This would be a condition of appointment and continued affiliation. Fellows may create SP.xxx residential subjects for MIT credit or assist in instruction in a regularly offered MIT subject.

A Schwarzman College Society of Fellows has clear benefits:

- Departments would have a stake in the SCoC as appointments should cycle among departments.
- The role of the SCoC would be identifiably different than an MIT school—it would not be perceived as “business as usual.”
- The SCoC would be regularly refreshed because MIT- and Visiting-SCSoF would come from cutting-edge,-research-active backgrounds or be thought leaders in the policy arena.
- Visiting-SCSoF will continually “enlarge the gene pool.”
- Turnover would allow the SCoC to initiate growth rapidly without creating a “plug-flow” of initial faculty or static silos.
- The creation of educational content, whether MITx or residential, would have continuing benefits to the Institute and its prestige. All departments would naturally benefit from computational subjects in their disciplines.

- MIT departments and the SCoC would share in the visibility of new SCoC research (i.e., a department benefits from computational advances in their discipline).
- The research directions of the SCoC would change more rapidly with developing research trends.
- The external visibility may be enhanced by the competitiveness of the external Fellows and postdocs.
- The creation of such Fellows may create opportunities for fundraising. (i.e., named Fellows). Mid-range donors may feel that they can make a recognizable contribution.
- The SCoC could begin to function prior to the completion of the Vassar Street building.
- Collaborations between the departments and the SCoC would be facilitated. The bridges would be structural and populated.
- The relevance to the educational mission of the entire Institute would be very clear to the MIT community.
- Social implications and moral hazards of computing would be included by regularly inviting legal scholars or social scientists as Visiting Fellows. As such, regular seminars would occur and contributions to the educational commons would result.

Of course, a rotating population of visiting faculty and postdocs would generate specific costs. The most important cost would be housing. For the MIT faculty appointed to the SCSof, some compensation to the department may be needed for a temporary loss of faculty while MIT faculty take up fellowships in the SCSof (perhaps appointments of instructional staff could help with this).

ORGANIZATIONAL STRUCTURE WORKING GROUP MEMBERS

Asu Ozdaglar (Co-Chair)

Head and Professor of Electrical Engineering, Department of Electrical Engineering and Computer Science; School of Engineering Distinguished Professor of Engineering

Nelson Repenning (Co-Chair)

Associate Dean of Leadership and Special Projects, and Professor of System Dynamics and Organization Studies, Sloan School of Management

Saman Amarasinghe

Associate Head and Professor of Computer Science and Engineering, Department of Electrical Engineering and Computer Science

Dimitris Bertsimas

Boeing Leaders for Global Operations Professor of Management, and Professor of Operations Research, Sloan School of Management

Cynthia Breazeal

Associate Professor, Program in Media Arts and Sciences

W. Craig Carter

Professor, Department of Materials Science and Engineering

Munther Dahleh

Director, Institute for Data, Systems, and Society; William A. Coolidge Professor, Department of Electrical Engineering and Computer Science

Jim DiCarlo

Head and Peter deFlorez Professor of Neuroscience, Department of Brain and Cognitive Sciences

Ken Goldsmith

Assistant Dean, School of Architecture and Planning

Nicolas Hadjiconstantinou

Professor, Department of Mechanical Engineering; Co-Director, Center for Computational Engineering

Dina Katabi

Andrew (1956) and Erna Viterbi Professor, Department of Electrical Engineering and Computer Science

Maryanne Kirkbride

Executive Administrator, MindHandHeart

Aleksander Madry

NBX Career Development Associate Professor of Computer Science and Engineering, Department of Electrical Engineering and Computer Science

Ankur Moitra

Rockwell International Career Development Associate Professor, Department of Mathematics

Eileen Ng

Assistant Dean, School of Engineering

Daniela Rus

Director, Computer Science and Artificial Intelligence Laboratory; Andrew (1956) and Erna Viterbi Professor, Department of Electrical Engineering and Computer Science

Susan Silbey

Leon and Anne Goldberg Professor of Humanities and Professor of Sociology and Anthropology, Anthropology Program; Professor of Behavioral and Policy Studies, Sloan School of Management; Member, Institute for Data, Systems, and Society

Rohil Verma

Undergraduate Student, Department of Electrical Engineering and Computer Science

Joel Voldman

Associate Head and Professor of Electrical Engineering, Department of Electrical Engineering and
Computer Science

Richard Zhang

Graduate Student, Department of Mathematics